# MÉTHODE D'IRI-IMAI POUR LA PROGRAMMATION LINÉAIRE

Justine Sauce et Aurélien Saou

**Introduction.** Lorsqu'on parle de programmation linéaire la méthode du Simplexe, introduite par George Dantzig en 1947, est souvent la première évoquée. Étant l'un des premiers algorithmes développés pour minimiser une fonction sous un ensemble de contraintes d'inégalités, elle a longtemps été la méthode de référence. Cependant en pratique, cette méthode n'offre pas de bons résultats lorsque la taille du problème est grande puisque le nombre d'itérations nécessaires à sa convergence augmente de façon exponentielle avec la taille du problème.

L'algorithme du Simplexe fut concurrencé au milieu des années 1980 par des méthodes dites méthodes de points intérieurs. Également rencontrées sous le nom de méthodes barrières, les méthodes de points intérieurs regroupent les algorithmes résolvants des problèmes de programmation convexe linéaires ou non. Ces méthodes reposent sur le fait que n'importe quel problème d'optimisation convexe peut-être ramené à la minimisation d'une fonction linéaire sur un ensemble convexe définit par des contraintes d'inégalités. L'idée derrière cela est d'introduire une fonction barrière (ou fonction de pénalité) encodant l'ensemble des contraintes. Cette idée avait été étudiée pour la première fois par Anthony V. Fiacco, Garth P. McCormick et d'autres au début des années 1960.

L'un des plus grands concurrents à la méthode du simplexe, dépassant largement les capacités de celle-ci, est l'algorithme proposé par Narendra Karmarkar en 1984. L'algorithme de Karmakar est une méthode de points intérieurs qui s'exécute en temps polynomial. La percée de cette méthode a alors relancé l'utilisation des méthodes de points intérieurs.

En 1986, M.Iri et H.Imai ont alors proposé un algorithme semblable à celui de Karmarkar, qui repose sur une méthode de Newton et qui présente une convergence linéaire voire quadratique. La fonction barrière introduite par M.Iri et H.Imai est en quelque sorte l'analogue affine de Karamakar. La preuve de la méthode d'Iri et d'Imai est nettement plus claire que celle de Karmakar, de plus elle ne nécessite pas de géometrie projective ni l'introduction de contrainte artificielle.

Dans cette étude, on va reprendre les éléments de preuve sur la convergence de la méthode d'Iri et d'Imai ainsi qu'effectuer des tests numériques en considérant le problème du cube déformé de Klee-Minty.

# Table des matières

# Partie Théorique

# 1 Introduction au problème

## 1.1 Définition du problème

En accord avec [2], le problème considéré est de minimiser la fonction objectif $c$, définie par,

$$c(x) = C^t\, x - c_0 = \sum_{i=1}^{n} c_i x_i - c_0 \tag{1}$$

où, $C = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^n$.

sous les inégalités de contraintes suivantes,

$$a_j(x) = \langle a_j, x\rangle - b_j = \sum_{i=1}^{n}\ a_{i,j}x_i - b_j \geq 0\,, \qquad j = 1, \ldots, m \tag{2}$$

où $c_0, c_i$, $b_j$ et $a_{i,j}$ sont fixés, pour pour $i = 1, \ldots, n$ et $j = 1, \ldots, m$ fixés.

L'ensemble des contraintes (ou ensemble admissible) est alors définit par,

$$X := \{x \in \mathbb{R}^n \,|\, a_j(x) \geq 0\} \tag{3}$$

**Remarque.** Les $a_j(x)$ sont des fonctions affines $\forall j \in \{1, 2, \ldots, m\}$ donc $X$ est un ensemble convexe.

On considère donc le problème d'optimisation sous-contraintes suivant,

$$(P) \qquad \min_{x\in X} c(x). \tag{4}$$

### 1.1.1 Écriture matricielle

En notant,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,j} & \ldots & a_{1,m} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,j} & \ldots & a_{2,m} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i,1} & a_{i,2} & \ldots & a_{i,j} & \ldots & a_{i,m} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,j} & \ldots & a_{n,m} \end{pmatrix} \in \mathcal{M}_{n,m}(\mathbb{R})\,, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

On peut réécrire (2) sous la forme,

$$A^t x - b \geq 0 \tag{5}$$

### 1.2 Hypothèses

Selon [2], on suppose que, $\mathring{X}$ est non-vide, et qu'il existe, un $x^{(0)} \in \mathring{X}$ fixé, tel que,

$$a_j\left(x^{(0)}\right) > 0 \quad j = 1, \ldots, m$$

Et qu'il existe une solution optimale $\bar{x}$ du problème $(P)$ telle que,

$$c(\bar{x}) = \min_{x \in X} c(x) = 0 \tag{6}$$

On émet alors les hyptohèses suivantes,

**$\mathbf{H_0}$** : $\bar{X} \neq X$, avec $\bar{X} = \{x \in X \,|\, c(x) = 0\}$, l'ensemble des solutions optimales.
i.e. [1], $c(x) > 0$, pour $x \in \mathring{X}$.
**$\mathbf{H_1}$** : $\bar{X}$ est borné.
**$\mathbf{H_2}$** : Pour une solution optimale, il y a au moins une contrainte inactive.

## 2 Fonction barrière multiplicative

### 2.1 Introduction

Dans le but de se ramener à un problème d'optimisation sans contraintes, on introduira une fonction barrière. En optimisation linéaire, une fonction barrière est une fonction continue, qui tend vers l'infini lorsque le point sur lequel on évalue la fonction se rapproche de la limite de l'ensemble admissible $X$. L'intérêt de celle-ci est donc de supprimer les contraintes d'inégalités, en reformulant le problème avec un terme pénalisant.

Par exemple, si on veut minimiser une fonction $c$ avec la contrainte $x \geq b$, où $b \in \mathbb{R}$ on peut changer le problème en minimisant $c(x) + h(x)$ avec cette fois l'inégalité supprimée, et $h(x)$ est définie comme étant égale à $\infty$ si $x \geq b$ et 0 sinon.

On définit ici une fonction barrière multiplicative selon [2],

$$F(x) = \frac{c(x)^{m+1}}{\prod_{j=1}^{m} a_j(x)} \tag{7}$$

La fonction $F$ est définie sur $\mathring{X}$, et on a $F(x) > 0$, $\forall x \in \mathring{X}$ sous l'hypothèse $\mathbf{H_0}$.

### 2.2 Équivalence des problèmes

La minimisation de la fonction barrière (7), est équivalente à la minimisation de la fonction objectif (1), sous les contraintes (2).

1. Puisque la solution optimale est atteinte sur le bord de l'ensemble des contraintes $X$, [6] implique que $c(x) > 0$ pour $x \in \mathring{X}$.

En effet, si $F(x^{(k)}) \underset{k\longrightarrow\infty}{\longrightarrow} 0$ pour toute suite de points $x^{(k)} \in \mathring{X}$ alors la suite converge vers la solution optimale du problème $(P)$.

On suppose que l'ensemble des $\{x^{(k)}, \forall k \geq 1\}$ est borné.
C'est-à-dire, $\exists M$ tel que, $\max(\,|x_1^{(k)}|, \ldots, |x_n^{(k)}|\,) \leq M$. Alors on en déduit que,

$$\begin{aligned} a_j(x^{(k)}) &= \sum_{i=1}^{n} a_{i,j} x_i^{(k)} - b_j \\ &\leq \sum_{i=1}^{n} a_{i,j} M - b_j \end{aligned}$$

Comme les $a_{i,j}$ et les $b_j$ sont des constantes fixées, on en déduit que $a_j(x)$ est borné. Par hypothèse $F(x^{(k)})$ converge vers 0, ceci implique alors que $c(x^{(k)})^{m+1}$ converge vers 0 et donc $c(x^{(k)})$ aussi. Et puisque $c(x) > 0$ si $x \in \mathring{X}$ alors $x^{(k)}$ tend vers la solution optimale [2].

De cette façon on étudiera alors un problème d'optimisation sans contraintes. On a supposé que la solution de $(P)$ était connue comme étant nulle (cf. (6)), et on vient de démontrer l'équivalence entre la recherche du minimum de $F(x)$ et de celui de $c(x)$ sous contraintes. Si de plus $F(x)$ est strictement convexe sur $\mathring{X}$, ce minimum sera unique. Sous cette condition de strict convexité on pourra appliquer une méthode de Newton. Le principe de cette méthode sera détaillé en section 3, page 10.

## 2.3 Expression des dérivées de la fonction barrière

On considère le logarithme de la fonction barrière, dont les dérivées seront plus faciles à manipuler que celles de la fonction barrière.

Notons $f$ cette fonction,

$$f(x) = \log F(x) = (m+1)\log c(x) - \sum_{j=1}^{m} \log a_j(x) \tag{8}$$

Remarquons tout d'abord que la dérivée de la log-barrière est de la forme,

$$\begin{aligned} \nabla f(x) = \nabla \log F(x) &= \frac{\nabla F(x)}{F(x)} \\ &= (m+1)\frac{\nabla(c(x))}{c(x)} - \sum_{j=1}^{m} \frac{\nabla a_j(x)}{a_j(x)} \end{aligned}$$

Composantes par composantes cela nous donne,

$$\begin{aligned} \nabla f_i(x) = \frac{\partial \log F(x)}{\partial x_i} &= \frac{1}{F(x)} \times \frac{\partial F(x)}{\partial x_i} \\ &= (m+1)\frac{c_i}{c(x)} - \sum_{j=1}^{m} \frac{a_{i,j}}{a_j(x)} \end{aligned}$$

2. Car la distance entre $x^{(k)}$ et $\bar{X}$ tend vers 0 quand $k$ tend vers $+\infty$.

Dans le but de simplifier la forme de $\nabla f(x)$, on suivra la notation de [2], en posant,

$$\begin{aligned}\tilde{c}_i(x) &= \frac{c_i}{c(x)} \\ \tilde{a}_{i,j}(x) &= \frac{a_{i,j}}{a_j(x)} \\ \bar{a}_i(x) &= \frac{1}{m}\sum_{j=1}^{m} \tilde{a}_{i,j}(x)\end{aligned}$$

Ainsi,

$$\nabla f_i(x) = (m+1)\tilde{c}_i(x) - m\,\bar{a}_i(x) \tag{9}$$

pour $i = 1, \ldots, n$.

D'autre part la matrice hessienne de $f$ se décompose de la façon suivante,

$$\begin{aligned}\nabla^2 f_{i,l}(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_l} &= \frac{\partial}{\partial x_l}\left(\frac{1}{F(x)} \times \frac{\partial F(x)}{\partial x_i}\right) \\ &= \frac{\left(\dfrac{\partial^2 F(x)}{\partial x_i \partial x_l} \times F(x) - \dfrac{\partial F(x)}{\partial x_i} \times \dfrac{\partial F(x)}{\partial x_l}\right)}{F^2(x)} \\ &= \frac{1}{F(x)}\frac{\partial^2 F(x)}{\partial x_i \partial x_l} - \frac{1}{F(x)}\frac{\partial F(x)}{\partial x_i}\frac{1}{F(x)}\frac{\partial F(x)}{\partial x_l} \\ &= \frac{1}{F(x)}\frac{\partial^2 F(x)}{\partial x_i \partial x_l} - \nabla f_i(x)\nabla f_l(x)\end{aligned}$$

pour $i, l = 1, \ldots, n$.

On remarque que grâce au gradient et à la hessienne de $f(x)$, on peut obtenir une expression du gradient et de la hessienne de $F(x)$ divisé par $F(x)$. On notera,

$$\begin{aligned}\nabla f(x) &= \frac{\nabla F(x)}{F(x)} \\ H(x) &= \frac{\nabla^2 F(x)}{F(x)}\end{aligned}$$

Ainsi, en obtenant une expression relativement simple de $\nabla f$ et de $H$ on pourra vérifier des propriétés de convexité pour $F$.

De ce qui précède,

$$H_{i,l}(x) = \nabla^2 f_{i,l}(x) + \nabla f_i(x)\nabla f_l(x)$$

Il reste à déterminer une expression de $\nabla^2 f_{i,l}(x)$, en dérivant $\nabla f_i(x)$ par rapport à $x_l$ on a,

$$\begin{aligned}\nabla^2 f_{i,l}(x) &= \frac{\partial\left((m+1)\frac{c_i}{c(x)} - \sum_{j=1}^m \frac{a_{i,j}}{a_j(x)}\right)}{\partial x_l} \\ &= \frac{-(m+1)c_i c_l}{c(x)^2} - \frac{-\sum_{j=1}^m a_{i,j}a_{l,j}}{a_j(x)^2} \\ &= -(m+1)\tilde{c_i}(x)\tilde{c_l}(x) + \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x)\end{aligned}$$

On peut alors réécrire $H(x)$ composantes par composantes tel que,

$$H_{i,l}(x) = \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - (m+1)\tilde{c_i}(x)\tilde{c_l}(x) + ((m+1)\tilde{c_i}(x) - m\,\bar{a}_i(x))\,((m+1)\tilde{c_l}(x) - m\,\bar{a}_l(x))$$

$$\begin{aligned}&= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - (m+1)\tilde{c_i}(x)\tilde{c_l}(x) + (m+1)^2\tilde{c_i}(x)\tilde{c_l}(x) \\ &\quad - m(m+1)(\tilde{c_i}(x)\bar{a_l}(x) + \tilde{c_l}(x)\bar{a_i}(x)) + m^2\bar{a_i}(x)\bar{a_l}(x)\end{aligned}$$

$$\begin{aligned}&= (m+1)(m+1-1)\tilde{c_i}(x)\tilde{c_l}(x) - m(m+1)(\tilde{c_i}(x)\bar{a_l}(x) + \tilde{c_l}(x)\bar{a_i}(x)) \\ &\quad + m^2\bar{a_i}(x)\bar{a_l}(x) + \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x)\end{aligned}$$

$$= m(m+1)\,(\tilde{c_i}(x) - \bar{a_i}(x))(\tilde{c_l}(x) - \bar{a_l}(x)) - m\bar{a_i}(x)\bar{a_l}(x) + \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x)$$

pour $i = 1, \ldots, n$ et $l = 1, \ldots, n$.

En remarquant que,

$$\begin{aligned}\sum_{j=1}^m (\tilde{a}_{i,j}(x) - \bar{a_i}(x))(\tilde{a}_{l,j}(x) - \bar{a_l}(x)) &= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - \tilde{a}_{i,j}(x)\bar{a_l}(x) - \bar{a_i}(x)\tilde{a}_{l,j}(x) + \bar{a_i}(x)\bar{a_l}(x) \\ &= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - \sum_{j=1}^m \tilde{a}_{i,j}(x)\bar{a_l}(x) - \sum_{j=1}^m \bar{a_i}(x)\tilde{a}_{l,j}(x) + \sum_{j=1}^m \bar{a_i}(x)\bar{a_l}(x) \\ &= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - \bar{a_l}\sum_{j=1}^m \tilde{a}_{i,j}(x) - \bar{a_i}(x)\sum_{j=1}^m \tilde{a}_{l,j}(x) + m\bar{a_i}(x)\bar{a_l}(x) \\ &= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - m\bar{a_l}(x)\bar{a_i}(x) - m\bar{a_i}(x)\bar{a_l}(x) + m\bar{a_i}(x)\bar{a_l}(x) \\ &= \sum_{j=1}^m \tilde{a}_{i,j}(x)\tilde{a}_{l,j}(x) - m\bar{a_l}(x)\bar{a_i}(x)\end{aligned}$$

On obtient,

$$H_{i,l}(x) = m(m+1)\,(\tilde{c_i}(x) - \bar{a_i}(x))(\tilde{c_l}(x) - \bar{a_l}(x)) + \sum_{j=1}^m (\tilde{a}_{i,j}(x) - \bar{a_i}(x))(\tilde{a}_{l,j}(x) - \bar{a_l}(x)) \quad (10)$$

### 2.3.1 Écriture matricielle

En réécrivant (9) sous forme matricielle, on obtient,

$$\nabla f(x) = \frac{(m+1)}{c(x)}C - AD^{-1}e \tag{11}$$

avec, $D(x) = \begin{pmatrix} a_1(x) & & \\ & \ddots & \\ & & a_m(x) \end{pmatrix} \in \mathcal{M}_{m,m}(\mathbb{R})$.

De même en utilisant les expressions de la section 2.3, page 6,

$$\nabla^2 f = \frac{-(m+1)}{c(x)^2}CC^t + AD^{-1}(AD^{-1})^t$$

Et,

$$H = \frac{-(m+1)}{c(x)^2}CC^t + AD^{-1}(AD^{-1})^t + \nabla f \nabla f^t \tag{12}$$

## 2.4 Propriétés de convexité

On rappelle qu'une fonction est convexe si sa matrice hessienne est semi-définie positive et strictement convexe si elle est définie positive.

De la formule (10) on pose

$$\begin{aligned} M_{i,j} &= (\tilde{a}_{i,j}(x) - \bar{a}_j(x)), \quad & M \in \mathcal{M}_{n,m} \\ L_i &= (\tilde{c}_i(x) - \bar{a}_i(x)), & L \in \mathbb{R}^n \end{aligned}$$

D'où,

$$H = m(m+1)LL^t + MM^t$$

Puisque $LL^t$ et $MM^t$ sont semie-définie positive, $H$ est semi-définie positive comme somme de matrices semi-définies positives. On a donc que $F$ est une fonction convexe.

On peut alors montrer que de plus, $H$ est définie positive en montrant que $u^tHu = 0 \Rightarrow u = 0$.

$$u^tHu = \sum_{i=1}^{n}\sum_{l=1}^{n} u_i H_{i,l} u_l$$

$$= \sum_{i=1}^{n}\sum_{l=1}^{n}\left( m(m+1)\left(\tilde{c}_i(x) - \bar{a}_i(x)\right)\left(\tilde{c}_l(x) - \bar{a}_l(x)\right) + \sum_{j=1}^{m}(\tilde{a}_{i,j}(x) - \bar{a}_i(x))(\tilde{a}_{l,j}(x) - \bar{a}_l(x))\right) u_i u_l$$

$$= \sum_{i=1}^{n}\sum_{l=1}^{n} m(m+1)\left(\tilde{c}_i(x) - \bar{a}_i(x)\right)u_i\left(\tilde{c}_l(x) - \bar{a}_l(x)\right)u_l + \sum_{i=1}^{n}\sum_{l=1}^{n}\sum_{j=1}^{m}(\tilde{a}_{i,j}(x) - \bar{a}_i(x))u_i(\tilde{a}_{l,j}(x) - \bar{a}_l(x))u_l$$

$$= m(m+1)\sum_{i=1}^{n}(\tilde{c}_i(x) - \bar{a}_i(x))u_i \sum_{l=1}^{n}(\tilde{c}_l(x) - \bar{a}_l(x))u_l + \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{l=1}^{n}(\tilde{a}_{i,j}(x) - \bar{a}_i(x))u_i(\tilde{a}_{l,j}(x) - \bar{a}_l(x))u_l$$

$$= m(m+1)\left(\sum_{i=1}^{n}(\tilde{c}_i(x)-\bar{a}_i(x))\right)^2+\sum_{j=1}^{m}\sum_{i=1}^{n}(\tilde{a}_{i,j}(x)-\bar{a}_i(x))u_i\sum_{l=1}^{n}(\tilde{a}_{l,j}(x)-\bar{a}_l(x))u_l$$
$$= m(m+1)\left(\sum_{i=1}^{n}(\tilde{c}_i(x)-\bar{a}_i(x))u_i\right)^2+\sum_{j=1}^{m}\left(\sum_{i=1}^{n}(\tilde{a}_{i,j}(x)-\bar{a}_i(x))u_i\right)^2$$

On suppose alors en raisonnant par l'absurde qu'il existe $u$ non-nul tel que $u^tHu=0$.

$$m(m+1)(\sum_{i=1}^{n}(\tilde{c}_i(x)-\bar{a}_i(x))u_i)^2+\sum_{j=1}^{m}(\sum_{i=1}^{n}(\tilde{a}_{i,j}(x)-\bar{a}_i(x))u_i)^2=0$$

$$\Leftrightarrow \sum_{i=1}^{n}(\tilde{c}_i(x)-\bar{a}_i(x))u_i=\sum_{i=1}^{n}(\tilde{a}_{i,j}(x)-\bar{a}_i(x))u_i=0$$

$$\Leftrightarrow \sum_{i=1}^{n}\tilde{c}_i(x)u_i=\sum_{i=1}^{n}\tilde{a}_{i,j}(x)u_i=\sum_{i=1}^{n}\bar{a}_i(x)u_i$$

$\forall j=1,\ldots,m$.

On note $J=\{a_j(\bar{x})=0\,|\,j=1,\ldots,n\}$ l'ensemble des contraintes actives.

On peut choisir $\hat{J}$ tel que l'ensemble $\{a_{i,j}\,|\,j\in\hat{J},\,i=1,\ldots,n\}$ forme une base de l'ensemble $\{a_{i,j}\,|\,j\in J, i=1,\ldots,n\}$.

De plus par définition de $\hat{J}$ on a,

$$\sum_{i=1}^{n}\tilde{a}_{i,j}(x)u_i=0,\quad \forall j\in\hat{J}$$

Cette égalité reste vraie $\forall j\in J$ car les $a_{i,j}, j\in J$ sont combinaisons linéaires des $a_{i,j}, j\in\hat{J}$. On déduit alors,

$$A^tu=0$$

Par conséquent $\forall t\in\mathbb{R}$, on a,

$$\begin{aligned}a_j(\bar{x}+tu)&=A^t(\bar{x}+tu)-b\\&=A^t\bar{x}+tA^tu-b\\&=A^t\bar{x}-b\\&=a_j(\bar{x})\geq 0\end{aligned}$$

Donc $\bar{x}+tu$ est une solution admissible $\forall t\in\mathbb{R}$.

De plus en posant $g=\sum_{i=1}^{n}\tilde{a}_{i,j}(x)u_i$. Alors de ce qui précède $g=0$.

Or $g=\sum_{i=1}^{n}\tilde{a}_{i,j}(x)u_i=\sum_{i=1}^{n}\tilde{c}_i(x)u_i$ ce qui implique que,

$$\sum_{i=1}^{n}\tilde{c}_i(x)u_i=0\Leftrightarrow C^tu=0$$

Et par le même raisonnement que précédemment,

$$\begin{aligned} c(\bar{x}+tu) &= C^t(\bar{x}+tu) - c_0 \\ &= C^t\bar{x} + tC^tu - c_0 \\ &= C^t\bar{x} - c_0 = c(\bar{x}) \end{aligned}$$

Ce qui signifie que $\bar{x} + tu, \forall t \in \mathbb{R}$ est une solution optimale.

On a supposé $u \neq 0$ donc si $g = 0$, on a que $\bar{x} + tu \in \bar{X}$, $\forall t \in \mathbb{R}$, ce qui contredit notre hypothèse que $\bar{X}$ est un ensemble borné. Donc $g$ ne peut pas être nul.

D'autre part, en posant $d(x)$ une fonction affine de la forme,

$$d(x) = \sum_{i=1}^{n} d_i x_i - d_0$$

Et $A_{\hat{J}}$ la matrice formée à partir des vecteurs colonnes $a_j$, $j \in \hat{J}$. On a,

$$\begin{aligned} d_i &= \sum_{j\in\hat{J}} a_{i,j}\beta_j \\ d &= A_{\hat{J}}\beta \end{aligned}$$

La matrice $A_{\hat{J}}$ est de rang plein égal à $\text{Card}(\hat{J})$ qui correspond au nombre de colonnes de $A$. Donc $A^t_{\hat{J}}A_{\hat{J}}$ est une matrice inversible. Ainsi,

$$\begin{aligned} & d = A_{\hat{J}}\beta \\ \Leftrightarrow\ & A^t_{\hat{J}}d = A^t_{\hat{J}}A_{\hat{J}}\beta \\ \Leftrightarrow\ & (A^t_{\hat{J}}A_{\hat{J}})^{-1}A^t_{\hat{J}}d = \beta \end{aligned}$$

De cette façon $\beta$ s'exprime de manière unique.

On peut alors écrire,

$$\begin{aligned} d(x) - d(\bar{x}) &= \sum_{i=1}^{n} d_i x_i - d_0 - \sum_{i=1}^{n} d_i\bar{x}_i + d_0 \\ &= \sum_{i=1}^{n}\sum_{j\in\hat{J}} \beta_j a_{i,j}x_i - \sum_{i=1}^{n}\sum_{j\in\hat{J}} \beta_j a_{i,j}\bar{x}_i \\ &= \sum_{j\in\hat{J}} \beta_j(\sum_{i=1}^{n} a_{i,j}x_i - \sum_{i=1} a_{i,j}\bar{x}_i) \\ &= \sum_{j\in\hat{J}} \beta_j(a_j(x) - a_j(\bar{x})) \end{aligned}$$

Or, $a_j(\bar{x}) = 0$ pour $j \in \hat{J}$.

Ainsi,

$$d(x) - d(\bar{x}) = \sum_{j\in\hat{J}} \beta_j a_j(x) \tag{13}$$

S'il existe $a_{j_0}(\bar{x})$ une contrainte inactive, en prenant $a_{j_0}(x) = d(x)$ tel que,

$$a_{j_0}(x) = \sum_{i=1}^{n} a_{i,j_0} x_i - b_{j_0} \quad \text{et} \quad a_{i,j_0} = \sum_{j \in \hat{J}} \beta_j a_{i,j}, \ \ \forall i = 1, \ldots, n$$

Alors on a,

$$\begin{aligned} & a_{j_0}(x) - a_{j_0}(\bar{x}) \neq a_{j_0}(x) \\ \Leftrightarrow & g\, a_{j_0}(x) \neq g(a_{j_0}(x) - a_{j_0}(\bar{x})) \end{aligned}$$

Cependant avec l'égalité (13) on obtient,

$$\begin{aligned} g(a_{j_0}(x) - a_{j_0}(\bar{x})) &= g \sum_{j \in \hat{J}} \beta_j a_j(x) \\ &= \frac{1}{a_j(x)} \sum_{i=1}^{n} a_{i,j} u_i \sum_{j \in \hat{J}} \beta_j a_j(x) \\ &= \sum_{j \in \hat{J}} \sum_{i=1}^{n} \beta_j a_{i,j} u_i \\ &= \sum_{i=1}^{n} a_{i,j_0} u_i \\ &= g a_{j_0}(x) \end{aligned}$$

Ce qui est absurde donc nécessairement pour que $u^t H u = 0$ il faut que $u = 0$ ce qui montre que $H$ est bien définie positive et donc $F(x)$ est une fonction strictement convexe.

# 3 Principe de l'algorithme

Puisque $F$ est une fonction strictement convexe et qu'elle atteint son minimum en 0, i.e.

$$\min_{x \in \mathring{X}} F(x) = 0$$

Alors pour la recherche du minimum on peut s'appuyer sur la méthode itérative de Newton.

Soit $x^{(k)}$ une suite telle que,

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$$

D'après la formule de Taylor-Young, on a,

$$F(x^{(k)} + \alpha^{(k)} d^{(k)}) \approx F(x^{(k)}) + \langle \nabla F(x^{(k)}), \alpha^{(k)} d^{(k)} \rangle + \frac{1}{2} \nabla F(x^{(k)}) \alpha^{(k)} d^{(k)}, \alpha^{(k)} d^{(k)} \rangle$$

Alors, la direction de Newton est la direction $d^*$ solution de,

$$\min \langle \nabla F(x^{(k)}), d \rangle + \frac{1}{2} \nabla^2 F(x^{(k)}) d, d \rangle$$

Ce qui est équivalent à résoudre,

$$\nabla F(x^{(k)}) + \nabla^2 F(x^{(k)}) d^* = 0$$

$$\Leftrightarrow d^* = -\frac{\nabla F(x^{(k)})}{\nabla^2 F(x^{(k)})}$$

Ainsi l'algorithme de Newton s'écrit :

**Initialisation** | $k = 0$
$x^{(0)} = x_0 \in \mathring{X}$

**Étape** $k$ | Itération :
$x^{(k+1)} = x^{(k)} - \nabla^2 F(x^{(k)})^{-1} \nabla F(x^{(k)})$

Dans notre cas, on a explicité $H(x)$ et $\nabla f(x)$, qui pour rappel sont respectivement la matrice hessienne de $F(x)$ divisée par $F(x)$ et le gradient de $F(x)$ divisé par $F(x)$. C'est donc sur cette base que repose l'algorithme, en commençant par la direction de descente $d$ qui est déterminée par,

$$d = -H^{-1}(x)\nabla f(x) \tag{14}$$

Rappelons que $H(x)$ est définie positive et $\nabla f(x)$ ne s'annule pas, pour tout $x \in \mathring{X}$. En remplaçant $d$ par son expression on a,

$$\langle \nabla f(x), d \rangle = \left\langle \nabla f(x), -H^{-1}(x)\nabla f(x) \right\rangle$$

Puisque $H$ est symétrique elle est diagonalisable dans une base orthonormée, on notera $\lambda_{max}$ la plus grande valeur propre de $H$. Puisque $H$ est définie positive, alors $\lambda_{max}$ est strictement positive ce qui nous donne,

$$\left\langle \nabla f(x), -H^{-1}(x)\nabla f(x) \right\rangle \leq -\frac{1}{\lambda_{max}} ||\nabla f(x)||^2 < 0$$

$$\Leftrightarrow \langle \nabla F(x), d \rangle < 0, \quad \text{puisque } F(x) > 0 \text{ pour } x \in \mathring{X}.$$

Ce qui démontre bien que $d$ est une direction de descente stricte pour $F(x)$.

L'algorithme proposé par Iri et Imai est le suivant. Soit $x_0 \in \mathring{X}$ et $\epsilon > 0$,

**Initialisation** | $k = 0$
$x^{(0)} = x_0$

**Étape** $k$ | Itération :
$\log F^{(k)} = \log F(x^{(k)})$
$\nabla f^{(k)} = \nabla f(x^{(k)})$
$H^{(k)} = H(x^{(k)})$
$d^{(k)} = -{H^{(k)}}^{-1} \nabla f^{(k)}$

Résoudre :
$\left[\frac{d}{dt} F(x^{(k)} + \alpha\, d^{(k)}) = 0\right]_{\alpha=\alpha^*}$
$x^{(k+1)} = x^{(k)} + \alpha^* d^{(k)}$

Incrémentation :
**Si :** $(F(x^{(k+1)}) < \epsilon) \longrightarrow$ **Arrêt.**
**Sinon :** $k = k + 1 \longrightarrow$ **Étape 1.**

Le choix du pas $\alpha$ peut être déterminé par une recherche linéaire (ou par itération de Newton) dans la direction $d$. On cherche,

$$\alpha^* = \arg \min_{\alpha \in \mathbb{R}^{+*}} F(x^{(k)} + \alpha\, d^{(k)})$$

On verra dans la suite un résultat théorique sur le choix du pas $\alpha$ pour avoir une convergence linéaire.

# 4 Synthèse de la convergence de l'algorithme

## 4.1 Préliminaires à la démonstration de la convergence

L'algorithme proposé par Iri-Imai repose sur une méthode de Newton ainsi on pourrait s'attendre à ce que sa convergence soit quadratique, ce qui est le cas lorsque le point initial $x_0$ est choisi très proche de la solution $\bar{x}$. Lorsque ce n'est pas le cas l'algorithme a tout de même une convergence linéaire ce que l'on va démontré en section 4.3. Commençons par remarquer que,

$$c(x) = c(x) - c(\bar{x}) = \sum_{i=1}^{n} c_i(x_i - \bar{x}_i)$$
$$a_j(x) = a_j(x) - a_j(\bar{x}) = \sum_{i=1}^{n} a_{i,j}(x_i - \bar{x}_i),\ j \in J$$

Dans le but de simplifier les calculs pour la démonstration de la convergence linéaire on pose $\epsilon_i^{(k)} = x_i^{(k)} - \bar{x}_i$ et on en déduit,

$$\sum_{i=1}^{n} \tilde{c}_i(x^k)\epsilon_i^{(k)} = \sum_{i=1}^{n} \frac{c_i x_i^{(k)} - c_i \bar{x}_i}{c(x^{(k)})} = \frac{c(x^{(k)}) - c(\bar{x})}{c(x^{(k)})} = 1$$
$$\sum_{i=1}^{n} \tilde{a}_{i,j}(x^{(k)})\epsilon_i^{(k)} = \sum_{i=1}^{n} \frac{a_{i,j}(x^{(k)}) - a_{i,j}(\bar{x})}{a_{i,j}(x^{(k)})} = \frac{a_j(x^{(k)}) - a_j(\bar{x})}{a_j(x^{(k)})} = 1 - \frac{a_j(\bar{x})}{a_j(x^{(k)})} \leq 1$$

On définit alors,

$$\begin{aligned}
\sum_{i=1}^{n} \nabla f_i^{(k)} \epsilon_i^{(k)} &= (m+1) \sum_{i=1}^{n} \tilde{c}_i(x^{(k)})\epsilon_i^k - \sum_{j=1}^{m}\sum_{i=1}^{n} \tilde{a}_{i,j}(x^{(k)})\epsilon_i^{(k)} \\
&= (m+1) \times 1 - \sum_{j=1}^{m} 1 - \frac{a_j(\bar{x})}{a_j(x^{(k)})} \\
&= (m+1) - m + \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \\
&= 1 + \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \geq 1
\end{aligned}$$

$$\begin{aligned}
\sum_{i=1}^{n} H_{l,i}^{(k)} \epsilon_i^{(k)} &= -(m+1)\, \tilde{c}_l\,(x^{(k)}) \sum_{i=1}^{n} \tilde{c}_i(x^{(k)})\epsilon_i^{(k)} + \sum_{j=1}^{m} \tilde{a}_{l,j}(x^{(k)}) \sum_{i=1}^{n} \tilde{a}_{i,j}\epsilon_i^{(k)} + \nabla f_l^{(k)} \sum_{i=1}^{n} \nabla f_i^{(k)} \epsilon_i^{(k)} \\
&= (-m+1)\, \tilde{c}_l(x^{(k)}) \times 1 + \sum_{j=1}^{m} \left( \tilde{a}_{l,j}(x^{(k)}) \times \left( 1 - \frac{a_j(\bar{x})}{a_j(x^{(k)})} \right) \right) + \nabla f_l^{(k)} \left( 1 + \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \right)
\end{aligned}$$

Or comme,

$$-\nabla f_l^{(k)} = -(m+1)\tilde{c}_l(x^{(k)}) + \sum_{j=1}^{m} \tilde{a}_{l,j}$$

En remplaçant dans l'égalité ci-dessus on obtient,

$$\sum_{i=1}^{n} H_{l,i}^{(k)} \epsilon_i^{(k)} = -\nabla f_l^{(k)} - \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \tilde{a}_{l,j}(x^{(k)}) + \nabla f_l^{(k)} + \nabla f_l^{(k)} \left( \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \right) \tag{15}$$

$$= \nabla f_l^{(k)} \left( \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \right) - \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})} \tilde{a}_{l,j}(x^{(k)}) \tag{16}$$

On pose,

$$w_j(x) = \frac{a_j(\bar{x})}{a_j(x)}$$
$$w(x) = \sum_{j \notin J} w_j(x)$$

car si $j \in J$ alors $w_j(x) = 0$.

## 4.2 Convergence quadratique

En supposant qu'il y ait une unique solution optimale $\bar{x}$ et que $x^{(k)}$ converge vers celle-ci. Si $||x^{(k)} - \bar{x}||$ est suffisamment petit alors $||x^{(k+1)} - \bar{x}|| = O(||x^{(k)} - \bar{x}||^2)$ ce qui implique la convergence quadratique de l'algorithme.

La démonstration de la convergence quadratique est délicate et longue, reposant sur des transformations affines et des notions subtiles telle que la magnitude des valeurs propres d'une matrice. Les idées de celle-ci sont relativement bien détaillées dans [2], mais dûe à sa complexité on préfèrera s'attarder sur la démonstration de la convergence linéaire.

## 4.3 Convergence linéaire de l'algorithme

Pour démontrer la convergence linéaire, on commence par supposer qu'il y a une unique solution optimale $\bar{x}$ vers laquelle $x^{(k)}$ converge.

Sous l'hypothèse qu'en partant de $x^{(k)}$ dans une direction arbitraire $d$ on peut approximer d'après Taylor-Young,

$$F(x^{(k)} + \alpha d) \approx F(x^{(k)}) + \langle \nabla F(x^{(k)}), \alpha d \rangle + \frac{1}{2} \langle \nabla^2 F(x^{(k)}) \alpha d, \alpha d \rangle \tag{17}$$

On en déduit alors que,

$$\frac{F(x^{(k)}) + \alpha d)}{F(x^{(k)})} = 1 + \frac{1}{F(x^{(k)})} \langle \nabla F(x^{(k)}), \alpha d \rangle + \frac{1}{F(x^{(k)})} + \frac{1}{2} \langle \nabla^2 F(x^{(k)}) \alpha d, \alpha d \rangle$$

A l'aide des relations déterminées précédemment et en introduisant $g(\alpha) = F(x^{(k)} + \alpha d^{(k)})$, on obtient,

$$\frac{g(\alpha)}{g(0)} = \frac{F(x^{(k)} + \alpha d)}{F(x^{(k)})} = 1 + \alpha\langle \nabla f(x^{(k)}), d\rangle + \frac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle \tag{18}$$

On suppose de plus que,

$$\frac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle \leq K^2 \tag{19}$$

avec $K > 0$.

Par conséquent, on peut approximer $\dfrac{g(\alpha)}{g(0)}$ comme une fonction linéaire en $\alpha$, avec une erreur d'au maximum $K^2$.

$$\frac{g(\alpha)}{g(0)} = 1 + \alpha\langle \nabla f(x^{(k)}), d\rangle$$

On cherche alors à minimiser cette fonction sous la contrainte (19),

$$\frac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle \leq K^2$$

Cette fonction étant de classe $\mathcal{C}^1$ on peut appliquer les conditions de Karush-Kuhn-Tucker.

$$\min_{\alpha\in\mathbb{R}} \; 1 + \alpha\langle \nabla f(x^{(k)}), d\rangle, \quad \frac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle - K^2 \leq 0$$

L'ensemble $Z = \left\{\alpha \in \mathbb{R} \,|\, \dfrac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle - K^2 \leq 0\right\}$ est un ensemble fermé et borné donc compact. $Z$ est non vide car $\alpha = 0$ appartient à $Z$.

La fonction $\alpha \mapsto 1 + \alpha\langle \nabla f(x^{(k)}), d\rangle$ est continue et admet donc au moins un minimum global sur $Z$.

Alors $\alpha$ est solution du problème si et seulement si,

$$\begin{cases} \exists \mu \in \mathbb{R}^+, \; \nabla(1 + \alpha\langle \nabla f(x^{(k)}), d\rangle) + \mu\nabla\left(\dfrac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle\right) = 0 \\ \dfrac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle - K^2 \leq 0 \\ \mu\left(\dfrac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle\right) = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \langle \nabla f(x^{(k)}), d\rangle + \mu\alpha\langle H(x^{(k)})d, d\rangle = 0 \\ \mu\left(\dfrac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle - K^2\right) = 0 \end{cases}$$

De la 2ème égalité on a 2 solutions possibles. Soit $\mu = 0$, soit $\mu \neq 0$.

- Si $\mu = 0$ alors de la première égalité on déduit que $\langle \nabla f(x^{(k)}), d\rangle = 0$ ce qui est absurde [3].
- Si $\mu \neq 0$ alors,

$$\begin{aligned}
&\frac{\alpha^2}{2}\langle H(x^{(k)})d, d\rangle - K^2 = 0 \\
&\Leftrightarrow \alpha^2 = \frac{2K^2}{\langle H(x^{(k)})d, d\rangle} \\
&\Leftrightarrow \alpha = \sqrt{\frac{2K^2}{\langle H(x^{(k)})d, d\rangle}}
\end{aligned}$$

On a donc unicité de $\alpha$. De plus le minimum de $\dfrac{g(\alpha)}{g(0)} = 1 + \alpha\langle \nabla f(x^{(k)}), d\rangle$ dépend de la direction $d$ et est atteint si on choisit $d^{(k)} = -H(x^{(k)})^{-1}\nabla f(x^{(k)})$.

En appliquant ce choix de $d$ et de $\alpha$ on déduit que,

$$\begin{aligned}
\hat{r} &\approx 1 + \alpha\langle \nabla f(x^{(k)}), d^{(k)}\rangle \\
&= 1 + \sqrt{\frac{2K^2}{\langle H(x^{(k)})d^{(k)}, d^{(k)}\rangle}}\langle -H(x^{(k)})d^{(k)}, d^{(k)}\rangle + K^2 \\
&= 1 - K\sqrt{2\langle H(x^{(k)}d^{(k)}, d^{(k)}\rangle} + K^2
\end{aligned}$$

On s'intéresse mainteant au cas où on prend $d = \bar{x} - x^{(k)} = -\epsilon^{(k)}$ et le même choix de $\alpha$. On va réécrire certaines égalités afin de simplifier le problème.

$$\begin{aligned}
\langle \nabla f(x^{(k)}), d\rangle &= \langle \nabla f(x^{(k)}), -\epsilon^{(k)}\rangle \\
&= -\sum_{i=1}^{n} \nabla f_i(x^{(k)})\epsilon_i^{(k)} \\
&= -\left(1 + \sum_{j=1}^{m} \frac{a_j(\bar{x})}{a_j(x^{(k)})}\right) \\
&= -\left(1 + \sum_{j=1}^{m} w_j(x^{(k)})\right) \\
&= -\left(1 + \sum_{j\notin J} w_j(x^{(k)})\right) \\
&= -(1 + w(x^{(k)}))
\end{aligned}$$

3. La solution est atteinte sur le bord donc $\nabla f(x^{(k)}) \neq 0$.

De même en utilisant (15),

$$
\begin{aligned}
\langle H(x^{(k)})d, d\rangle &= \langle H(x^{(k)})\epsilon^{(k)}, \epsilon^{(k)}\rangle \\
&= \sum_{i=1}^{n}\sum_{l=1}^{n} H_{i,l}(x^{(k)})\epsilon_i^{(k)}\epsilon_l^{(k)} \\
&= \sum_{l=1}^{n}\left(\sum_{i=1}^{n} H_{l,i}(x^{(k)})\epsilon_i^{(k)}\right)\epsilon_l^{(k)} \\
&= \sum_{l=1}^{n}\left(\nabla f_l(x^{(k)})\left(\sum_{j=1}^{m}\frac{a_j(\bar{x})}{a_j(x^{(k)})}\right) - \sum_{j=1}^{m}\frac{a_j(\bar{x})}{a_j(x^{(k)})}\tilde{a}_{l,j}(x^{(k)})\right)\epsilon_l^{(k)} \\
&= \sum_{l=1}^{n}\left(w(x^{(k)})\nabla f_l(x^{(k)}) - \sum_{j=1}^{m} w_j(x^{(k)})\tilde{a}_{l,j}(x^{(k)})\right)\epsilon_l^{(k)} \\
&= \sum_{l=1}^{n}\left(w(x^{(k)})\left(\nabla f_l(x^{(k)}) - \frac{1}{w(x^{(k)})}\sum_{j=1}^{m} w_j(x^{(k)})\tilde{a}_{l,j}(x^{(k)})\right)\right)\epsilon_l^{(k)} \\
&= w(x^{(k)})\left(\sum_{l=1}^{n}\nabla f_l(x^{(k)})\epsilon_l^{(k)} - \sum_{l=1}^{n}\frac{1}{w(x^{(k)})}\sum_{j=1}^{m} w_j(x^{(k)})\tilde{a}_{l,j}(x^{(k)})\epsilon_l^{(k)}\right) \\
&= w(x^{(k)})\left(1 + w(x^{(k)}) - \frac{1}{w(x^{(k)})}\sum_{j=1}^{m}\sum_{l=1}^{n} w_j(x^{(k)})\tilde{a}_{l,j}(x^{(k)})\epsilon_l^{(k)}\right) \\
&= w(x^{(k)})\left(1 + w(x^{(k)}) - \frac{1}{w(x^{(k)})}\sum_{j=1}^{m} w_j(x^{(k)})\sum_{l=1}^{n}\tilde{a}_{l,j}(x^{(k)})\epsilon_l^{(k)}\right) \\
&= w(x^{(k)})\left(1 + w(x^{(k)}) - \frac{1}{w(x^{(k)})}\sum_{j\notin J} w_j(x^{(k)})\left(1 - \frac{a_j(\bar{x})}{a_j(x^{(k)})}\right)\right) \\
&= w(x^{(k)})\left(1 + w(x^{(k)}) - \frac{1}{w(x^{(k)})}\sum_{j\notin J} w_j(x^{(k)})\left(1 - w_j(x^{(k)})\right)\right) \\
&= w(x^{(k)}) + w(x^{(k)})^2 - w(x^{(k)}) + \sum_{j\notin J} w_j(x^{(k)})^2 \\
&= w(x^{(k)})^2 + \sum_{j\notin J} w_j(x^{(k)})^2
\end{aligned}
$$

On déduit alors,

$$
\begin{aligned}
r &\approx 1 - \alpha(1 + w(x^{(k)})) + K^2 \\
&= 1 - \frac{\sqrt{2}(1 + w(x^{(k)}))}{\sqrt{w(x^{(k)})^2 + \sum_{j\notin J} w_j(x^{(k)})^2}} + K^2
\end{aligned}
$$

Puisque $w_j(x^{(k)}) > 0$ et $w(x^{(k)}) = \sum_{j\notin J} w_j(x^{(k)})$ on déduit que,

$$
w(x^{(k)})^2 \geq \sum_{j\notin J} w_j(x^{(k)})^2
$$

Alors on obtient,

$$\begin{aligned}\sqrt{w(x^{(k)})^2 + w(x^{(k)})^2} &\geq \sqrt{w(x^{(k)})^2 + \sum_{j\notin J}^{m} w_j(x^{(k)})^2} \\ \Leftrightarrow \sqrt{2}w(x^{(k)}) &\geq \sqrt{w(x^{(k)})^2 + \sum_{j\notin J} w_j(x^{(k)})^2}\end{aligned}$$

Finalement,

$$\begin{aligned}r &\leq 1 - \frac{\sqrt{2}K(1+w(x^{(k)}))}{\sqrt{2}w(x^{(k)})} + K^2 \\ &= 1 - \frac{K(1+w(x^{(k)}))}{w(x^{(k)})} + K^2 \\ &= 1 - K - \frac{K}{w(x^{(k)})} + K^2 \\ &< 1 - K + K^2 = 1 - K(1-K)\end{aligned}$$

Dépendemment du choix de la direction on a que $\hat{r} \leq r$ ce qui implique $\hat{r} < 1 - K(1-K)$.

Ce qui nous permet de conclure, sous les hypothèses (17) et (19) que,

$$\frac{F(x^{(k+1)})}{F(x^{(k)})} < 1 - K(1-K)$$

C'est à dire qu'à chaque itération la valeur de $F(x^{(k)})$ est réduite d'au moins un facteur $1 - K(1-K)$ ceci implique donc par définition la convergence linéaire de l'algorithme.

**Remarque.** Le document [2] mentionne même une convergence super-linéaire.

Expériences numériques

# 5 Implémentation de l'algorithme

## 5.1 Méthode d'Iri et Imai sous Scilab

Lors de la mise en place du problème, on a pris soin de rédéfinir le problème de façon matricielle, ce qui nous sera utile pour l'implémentation de l'algorithme. On implémentera alors les fonctions $F$, $\nabla f$, et $H$ sous Scilab, en commençant par $F(x)$.

```
function [y] = F(C,c0,A,x,b)
     // Entrees :
        // C : vecteur de la fonction objectif (1)
        // c0 : constante de la fonction objectif (1)
        // A : matrice des contraintes (2)
        // x : vecteur (x_1, ..., x_n) (1)
        // b : vecteur constant des contraintes (2)
    //Sorties :
        // y : valeur de la fonction barriere en x, F(x)
        m = size(A,2); //nombre de contraintes (colonnes de A)
    m = size(A,2);
    y = (C'*x-c0)^(m+1)/prod(A'*x-b);
endfunction
```

L'implémentation de $F$ nous permettra d'avoir un critère d'arrêt sur la précision de $F(\bar{x})$.

Le gradient de $F(x)$ divisé par $F(x)$, $\nabla f(x)$, est implémenté comme-suit, à l'aide de (11),

```
function [y] = grad_f(C,c0,A,x,Dinv)
    // Entrees :
        // C : vecteur de la fonction objectif (1)
        // c0 : constante de la fonction objectif (1)
        // A : matrice des contraintes (2)
        // x : vecteur (x_1, ..., x_n) (1)
        // Dinv : matrice inverse de D definie en (11)
    //Sorties :
        // y : valeur du gradient en x, gradf(x)
        m = size(A,2); //nombre de contraintes (colonnes de A)
        y = (m+1)/(C'*x-c0)*C-A*Dinv*ones(m,1);
endfunction
```

Pour $H(x)$, en utilisant $\nabla f(x)$ et (12), on implémente la fonction suivante.

```
function [y] = H(C,c0,A,x,Dinv,gradf)
    // Entrees :
        // C : vecteur de la fonction objectif (1)
        // c0 : constante de la fonction objectif (1)
        // A : matrice des contraintes (2)
        // x : vecteur (x_1, ..., x_n) (1)
        // Dinv : matrice inverse de D definie en (11)
        // gradf : valeur du gradient en x, gradf(x)
    //Sorties :
        // y : valeur de la Hessienne en x, H(x)
        m = size(A,2);
        y = sparse(-(m+1)/(C'*x-c0)^2*C*C'+A*Dinv^2*A'+gradf*gradf');
endfunction
```

A l'aide des trois fonctions précedemment codées, on introduit alors l'algorithme basé sur le principe de M.Iri et H.Imai suivant.

```
function [iter, x, Xk] = Iri_Imai(C,c0,A,x0,b,epsilon,N,K)
    // Principe : Calculer le min de C'*x-c0 sous les contraintes A'*x-b > 0
    // Entrees :
        // C : vecteur de la fonction objectif (1)
        // c0 : constante de la fonction objectif (1)
        // A : matrice des contraintes (2)
        // x0 : vecteur initial dans Int(X) (X : ens. des contraintes)
        // b : vecteur constant des contraintes (2)
        // epsilon : précision
        // N : nombre d'iterations maximal
        // K : constante définissant le pas alpha
    //Sortie :
        // iter : nombre d'iterations necessaires
        // x : solution
        // Xk : liste des itérés

    //Initialisation
    iter = 0;
    Xk = list();
    x = x0;

    //Algorithme
    while (iter < N & F(C,c0,A,x,b) > epsilon )
        Dinv = sparse(diag((A'*x-b))^(-1));
        gradfk = grad_f(C,c0,A,x,Dinv);
        Hk = H(C,c0,A,x,Dinv,gradfk);
        dk = -Hk\gradfk;
        alpha = sqrt(2*K**2/(dk'*Hk'*dk));
        x = x + alpha*dk;

        //Incrementation
        iter = iter + 1;
        Xk(iter) = x;
    end
endfunction
```

On pourrait aussi choisir de fixer le critère d'arrêt à $c(x^{(k)}) < \epsilon$.

## 5.2 Exemple à petite échelle

Commençons par introduire un problème de petite taille, par exemple dans $\mathbb{R}^2$, pour tester son fonctionnement et sa vitesse convergence. Considérons le cas $\langle$C1$\rangle$ de la partie (**8. Small-Size Problems**) de [2], avec quatre contraintes d'inégalités.

$$\langle \text{C1} \rangle : c(x) = x + y, \quad \begin{aligned} a_1(x) &= x \\ a_2(x) &= y \\ a_3(x) &= 2 - 2x - y \\ a_4(x) &= 3 + 2x - 4y \end{aligned}$$

### 5.2.1 Écriture matricielle de $\langle$C1$\rangle$

Ce qui se traduit par,

$$C = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \in \mathbb{R}^2, \quad c_0 = 0$$

$$A = \begin{pmatrix} 1 & 0 & -2 & 2 \\ 0 & 1 & -1 & -4 \end{pmatrix} \in \mathcal{M}_{2,4}(\mathbb{R}), \quad b = \begin{pmatrix} 0 \\ 0 \\ -2 \\ -3 \end{pmatrix} \in \mathbb{R}^4.$$

### 5.2.2 Application de la méthode

On choisit $x_0 = \left(\frac{1}{2}, \frac{1}{2}\right) \in \mathring{X}$, et $K = 1$.

```
C = [1;1];
c0 = 0;
A = [1,0,-2,2;0,1,-1,-4];
b = [0;0;-2;-3];
x0 = [1/2;1/2];
[i, xbar, Xit] = Iri_Imai(C,c0,A,x0,b,10^-10,20,1)
```

L'algorithme atteint la solution avec une précision de $10^{-4}$ en seulement 12 itérations et environ 1 à 2ms.

# 6 Le cube déformé de Klee-Minty

## 6.1 Mise en place du problème

Le problème du cube déformé de Klee-Minty est un exemple réputé pour montrer la complexité de calcul dans le pire des cas de nombreux algorithmes d'optimisation linéaire. Il est basé sur un cube déformé avec $2 \times N$ coins en dimension $N$. Sur ce problème linéaire à $N$ variables, la méthode du simplexe a une vitesse de convergence exponentielle en $N$.

On introduira le problème de Klee-Minty sous la forme proposée par Avis et Chvátal selon [2], où, $0 < \epsilon < \frac{1}{2}$.

$$\langle \text{KM} \rangle \ : \ \max \sum_{i=1}^{N} \epsilon^{N-i} x_i, \quad 2\sum_{i=1}^{j-1} \epsilon^{j-i} x_i + x_j \leq 1 \quad \text{pour, } j = 1, \ldots, N$$
$$x_i \geq 0 \quad \text{pour, } i = 1, \ldots, N$$

Ce qui est équivalent à chercher $x$ tel que,

$$\min -\sum_{i=1}^{N} \epsilon^{N-i} x_i, \quad -2\sum_{i=1}^{j-1} \epsilon^{j-i} x_i - x_j + 1 \geq 0 \quad \text{pour, } j = 1, \ldots, N$$
$$x_i \geq 0 \quad \text{pour, } i = 1, \ldots, N$$

La solution optimale du problème, $\bar{x}$, est à priori $x_j = 0$ pour $j = 1, \ldots, N-1$ et $x_N = 1$.

### 6.1.1 Écriture matricielle de Klee-Minty

En notant le problème (20) sous forme matricielle, $n = N$ et $m = 2N$, comme-suit,

$$\min -C^t x, \quad -a^t x + e \geq 0, \quad x \in \mathbb{R}^N \tag{20}$$
$$x \geq 0 \tag{21}$$

On aura alors $C = \begin{pmatrix} \epsilon^{N-1} \\ \vdots \\ \epsilon \\ 1 \end{pmatrix} \in \mathbb{R}^N$, $e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^N$ et la matrice des contraintes s'écrira,

$$a = \begin{pmatrix} 1 & 2\epsilon & 2\epsilon^2 & \dots & 2\epsilon^{N-1} \\ 0 & 1 & 2\epsilon & \dots & 2\epsilon^{N-2} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 2\epsilon \\ 0 & & \dots & 0 & 1 \end{pmatrix} \in \mathcal{M}_{N,N}(\mathbb{R})$$

On peut maintenant inclure les contraintes sur la positivité des coefficients de $x$ en posant,

$$A = \left( \begin{array}{c|c} I_N & -a \end{array} \right) \in \mathcal{M}_{N,2N}(\mathbb{R}), \quad \text{et} \quad b = \begin{pmatrix} 0_{\mathbb{R}^N} \\ -e \end{pmatrix} \in \mathbb{R}^{2N}$$

Ainsi le problème se réécrit de la façon suivante,

$$\min -C^t x, \quad A^t x - b \geq 0, \quad x \in \mathbb{R}^N \tag{22}$$

Puisqu'on connait $\bar{x}$, on peut se ramener à une valeur optimale nulle, ie. $\min c(x) = C^t x - c_0 = 0$ en posant $c_0 = -C^t \bar{x} = -1$. Ainsi on pourra appliquer l'algorithme précèdemment proposé.

## 6.2 Expériences numériques sur Klee-Minty

La question du point inital $x_0 \in \mathring{X}$ est résolue en choisissant, $x_0 = \left( \frac{1}{N}, \dots, \frac{1}{N} \right) \in \mathbb{R}^N$.

Aussi le choix de $K$ (la constante fixée dans la défintion du pas $\alpha$) restera celui utilisé dans le problème précédent c'est-à-dire $K = 1$, pour cette valeur on obtient de bon résultats sur la vitesse de convergence.

On choisira un nombre maximal de 500 itérations et une précision de $10^{-7}$ sur la minimisation de la fonction barrière, ce qui implique une précision d'environ $10^{-3}$ sur la fonction objectif.

Enfin, on utilisera les relations explicitées ci-dessus (22) pour implémenter le problème, et on illustrera la convergence de l'algorithme en représentant $C(x^{(k)})$ pour chaque itérés en fonction des itérations effectuées.

Les étapes sous Scilab sont décrites ci-contre,

```
//Implémentation du problème <KM>
E = 0.4; // 0 < epsilon < 1/2
N = 10; //N = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
x0 = ones(N,1)/N;

IN = eye(N,N);
a = zeros(N,N);
C = zeros(N,1);

for i = 1:N-1
    a(i,i) = 1;
    for j = i+1:N
        a(i,j) = 2*E^(j-i);
    end
    C(N-i)=E^(i);
end

a(N,N)=1;
A = sparse(cat(2,IN,-a));
C(N)=1;
c0 = -1;
b = cat(1, zeros(N,1), -ones(N,1));

//Application de l'algorithme
tic()
[i, xbar, Xk] = Iri_Imai(-C,c0,A,x0,b,10^-7,500,1);
timeKM = toc() //temps de calcul

//Illusration de la convergence
CXk = zeros(i,1);
I = (1:i)';
for k=1:i
    CXk(k)=C'*Xk(k);
end

scf(0);
plot2d(I,CXk);
```

**Remarque.** L'utilisation du mode `sparse()` de Scilab est utile dans le cadre du problème de Klee-Minty, étant donné que la matrice des contraintes est composée de la matrice identité et d'une matrice triangulaire, presque trois-quarts de ses coefficients sont nuls. On verra en section 6.3 que dans le cas du dual la matrice des contraintes contiendra encore plus de zéros.

Puisque l'algorithme est implémenté avec le pas théorique $\alpha$ explicité en section 4.3, on s'attend donc à une convergence en $\mathcal{O}(N)$.

Les résultats numériques illustrent bien cette convergence, pour $N = 40$ on converge vers la solution à une précision $10^{-2}$ sur la fonction objectif en 0.12s et 113 itérations. Pour $N = 100$ on garde une convergence linéaire qui semble toujours être en $\mathcal{O}(3N)$, on atteint la solution avec les mêmes paramètres en 298 itérations.

En résolvant le problème pour $N = 10, 20, 30, 40, 0, 60, 70, 80, 90, 100$ à l'aide de la méthode d'Iri et d'Imai on obtient les résultats présentés sur le graphe ci-dessous.

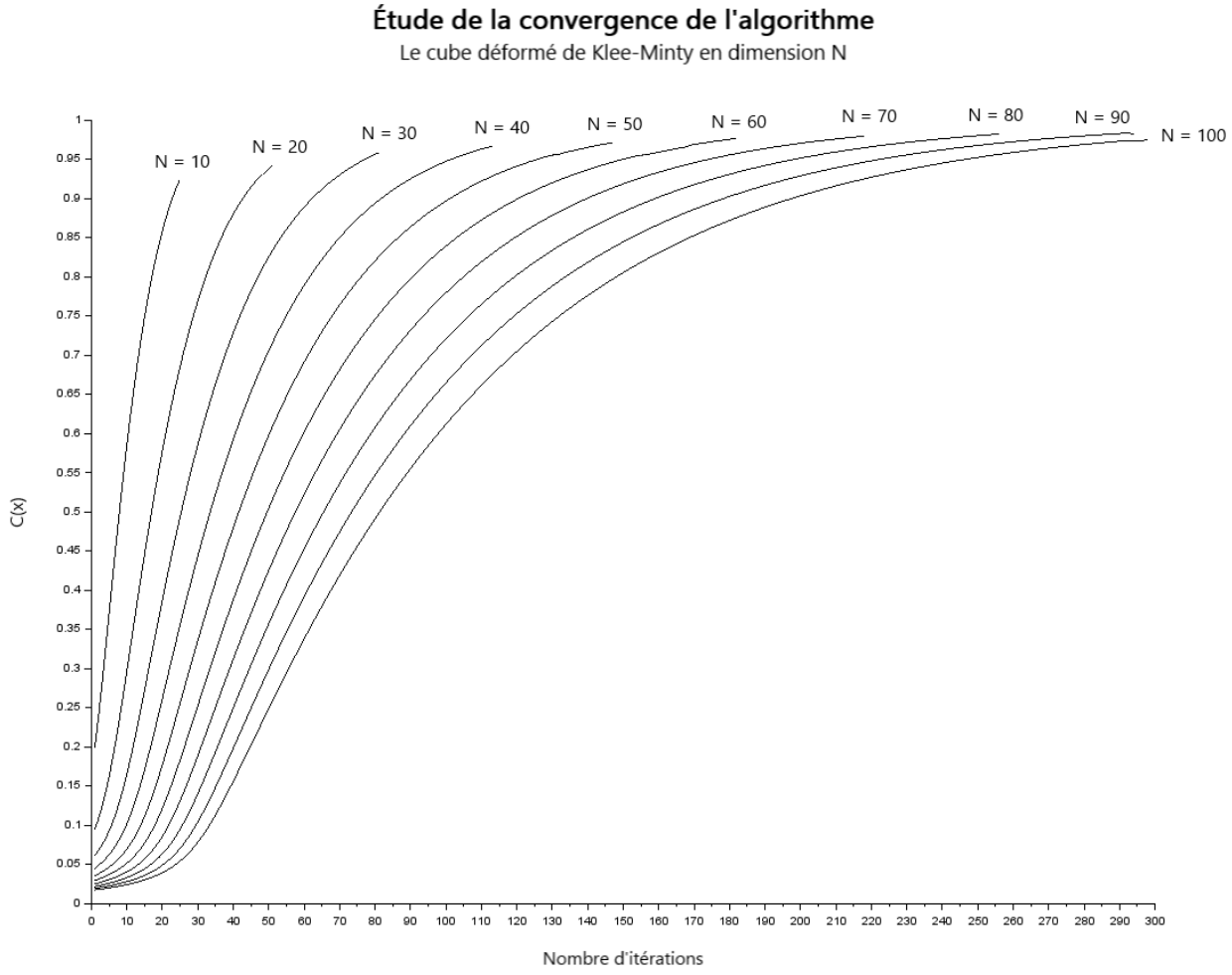


FIGURE 1 – Résultats de convergence $\mathcal{O}(N)$ sur le problème $\langle \mathrm{KM} \rangle$

Notons cependant que le choix de $c_0$ ne peut-être fait qu'en connaissant la valeur de la solution optimale du problème. C'est une condition qui est restrictive et l'on peut s'en passer en considérant le dual du problème.

## 6.3 Problème dual de Klee-Minty

On considère le problème (20) et son dual,

$$\begin{aligned} C^t\bar{x} = \min -C^t x, \quad & -a^t x + e \geq 0, \quad x \in \mathbb{R}^N \\ & x \geq 0 \\ C^t\bar{x} = \min e^t y, \quad & ay - c \geq 0, \quad y \in \mathbb{R}^N \\ & y \geq 0 \end{aligned}$$

On peut combiner les deux parties ce qui nous donne,

$$\begin{aligned} 0 = \min e^t y - C^t x \quad & -a^t x + e \geq 0, \quad x \in \mathbb{R}^N \\ & x \geq 0 \\ & ay - c \geq 0, \quad y \in \mathbb{R}^N \\ & y \geq 0 \end{aligned}$$

De la même façon qu'en section 6.1.1 page 21, on peut inclure dans une matrice les contraintes $x, y \geq 0$ en posant,

$$\tilde{A} = \left( \begin{array}{c|c} I_{2N} & \begin{array}{c|c} -a & 0_{N\times N} \\ \hline 0_{N\times N} & a^t \end{array} \end{array} \right) \in \mathcal{M}_{2N,4N}(\mathbb{R}) \quad \text{et} \quad \tilde{b} = \begin{pmatrix} 0_{\mathbb{R}^{2N}} \\ -e \\ -C \end{pmatrix} \in \mathbb{R}^{4N}$$

De la même façon, on aura $\tilde{x} = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{2N}$ et $\tilde{C} = \begin{pmatrix} -C \\ -e \end{pmatrix} \in \mathbb{R}^{2N}$.

Ainsi, le problème devient,

$$\min \tilde{C}^t \tilde{x}, \quad \tilde{A}^t \tilde{x} - \tilde{b} \geq 0 \tag{23}$$

Le minimum du problème vaut alors 0, on peut directement appppliquer l'algorithme en utilisant l'expression ci-dessus. On implémentera la nouvelle matrice et les nouveaux vecteurs à partir de ceux posés en section 6.2, de la façon suivante,

```
I2N = eye (2*N,2*N);
a1t = cat(2, -a, zeros(N,N));
a2t = cat(2, zeros(N,N), a');
at = cat(1, a1t, a2t);
At = sparse(cat(2,I2N,at));

e = ones(N,1);
Ct = cat(1,-C,e);
bt = cat(1,zeros(2*N,1),-e,-C);

x0 = ones(N,1)/N;
y0 = ones(N,1);
xt0=cat(1,x0,y0);

[itert, xtbar, Xtk] = Iri_Imai(Ct,0,At,xt0,bt,10^-8,200,1);
```

Dans le cas où l'on combine le problème avec son dual, les dimensions de la matrice des contraintes et du problème de manière générale sont doublées. Cela n'est pas réellement dérangeant dans le cas de cet algorithme puisque la convergence est linéaire, mais on privilégiera la méthode précedente lorsque qu'on connait la valeur de la solution optimale.

**Conclusion.** L'objectif était de minimiser une fonction linéaire sous des contraintes d'inégalités affines. Dans ce but, nous avons présenté et redemontré les propriétés de l'algorithme d'Iri et Imai.

La méthode d'Iri et d'Imai repose sur l'introduction d'une fonction barrière multiplicative bien choisie. En effet, la fonction barrière introduite était strictement convexe et son minimum était atteint en $F(x) = 0$, ce qui a permit l'utilisation d'une méthode de Newton. Une fois l'équivalence démontrée entre le problème initial et la minimisation de la fonction barrière, on résolvait alors le problème initial en appliquant un algorithme itératif de Newton à la fonction barrière.

La convergence linéaire de l'algorithme est un réel point fort qui a été démontré pour un certain pas $\alpha$ et qui permet la résolution de problème à grande échelle. En particulier dans le cadre du problème de Klee-Minty, les résultats numériques montrent une convergence en $\mathcal{O}(N)$ où $N$ définit la dimension du cube. Là où l'algorithme du simplexe avait une convergence en $\mathcal{O}(e^N)$.

# Références

[1] N. Karmarkar.*A new polynomial-time algorithm for linear programming.* Combinatorica 4 (1984), no. 4, 373–395.

[2] M. Iri and H. Imai. *A multiplicative barrier function method for linear programming*, Algorithmica 1 (1986), no. 4, 455–482.