RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDELHAMID IBN BADIS MOSTAGANEM

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master en mathématiques

Présenté par

Melle *FEKIR Fatima*
Option : *Modélisation Contrôle et Optimisation*

Thème

Quelques variantes de la méthode Active-Set pour l'optimisation quadratique

Président :	Mr.Ablaoui Hocine. M-A-A. Université de Mostaganem.
Examinateur :	Mr.Bokhari Ahmed. M-A-A. Université de Médea.
Encadreur :	Mr.AMIR Abdessamad. M-C-A. Université de Mostaganem.

Promotion 2015

Table des matières

Remerciements

Au terme de ce travail je tiens à remercier : Monsieur AMIR ABDESSAMAD mon directeur de mémoire pour ses orientations qui m'ont été utiles pour la réalisation de ce travail.

Monsieur BELHAMITI OMAR le Chef de département de Mathématiques. Monsieur Bokhari Ahmed qui m'a fait l'honneur de présider le Jury. Monsieur ABLAOUI HOCINE pour avoir accepté de juger ce travail.

Mes vifs remerciements s'adressent également à : Tous les enseignants de département de mathematique pour leur enseignement.

Enfin, à tous ceux qui ont participé de prés où de loin, à l'élaboration de ce mémoire, trouvent ici mes remerciements les plus sincères.

Merci

Introduction

Un problème de programmation quadratique (En anglais, Quadratique Programming (QP)) est un problème d'optimisation dans laquelle on minimise (ou maximise) une fonction quadratique avec des contraintes égalité et inégalité, ces contraintes peuvent être décrites par des fonctions linéaires. Beaucoup de problèmes pratiques se modélisent via un QP, à titre d'exemple le problème d'optimisation de portefeuille en finance ; les problèmes de classification SVM (machine à vecteurs de supports) en apprentissage automatique...ext. La programmation linéaire peut être vue comme un cas particulier de la programmation quadratique. Pour résoudre un QP dans le cas convexe on à plusieurs méthodes de résolution, comme les méthodes de point intérieur ; la méthode Active-Set ; la méthode de projection de gradient ;... ext.

Nous considérons dans ce mémoire la méthode Active-Set. La méthode Active-Set peut être vue comme une généralisation de la méthode du Simplexe Dantzig, elle a été utilisée à partir des années 70, et elle est efficace pour résoudre les problèmes quadratiques de taille petites et moyennes.

Ce mémoire est organisé comme suit : Dans le premier chapitre, on propose une introduction en Programmation Quadratique, nous donnons ainsi les conditions nécessaires d'optimalité pour un QP avec contraintes égalité et/ou inégalité. Dans le second nous présentons la méthode Active-Set de base, que nous implémentons et testons sur un exemple académique. Dans le dernier chapitre on propose une technique efficace qui permet de résoudre un sous programme dans Active-Set basé sur la factorisation QR.

Chapitre 1

Introduction en Programmation Quadratique

En général un programme de programmation quadratique est de la forme :

$$\begin{aligned} \min_x q(x) &= \frac{1}{2}x^T G x + x^T c, \\ a_i^T x &= b_i, i \in J \\ a_i^T x &\geqslant b_i, i \in I \end{aligned} \tag{1.1}$$

avec $G \in \mathbb{R}^{n\times n}$ une matrice symétrique, J et I des ensembles d'indices finies, et c, x, $\{a_i\} \in \mathbb{R}^n$ pour tout $i \in I \cup J$. L'effort nécessaire pour trouver une solution du problème [1.1] dépend fortement des caractéristiques de la fonction objectif et les contraintes. Si la matrice Hessienne G est semi définie positive, alors [1.1] est un programme quadratique convexe ; il est connu dans ce cas que le problème est souvent similaire en difficulté à un programme linéaire. Si la matrice Hessienne G est indéfinie, alors [1.1] est un programme quadratique non convexe, ce dernier peut être plus difficile, car il peut avoir plusieurs points stationnaires ou minimums locaux. Nous aborderons d'abord le cas où le problème [1.1] admet uniquement des contraintes égalité.

1.1 Programme quadratique avec contraintes égalité

1.1.1 Conditions d'optimalité

Le problème considéré est donc

$$\begin{cases} \min\limits_x q(x) = \frac{1}{2}x^T G x + x^T c \\ a_i^T x = b_i, i \in J \end{cases}$$

Pour plus de simplicité, nous écrivons les contraintes égalités en forme matricielle et le programme quadratique avec contrainte égalité s'écrit :

$$\begin{cases} \min q(x) = \frac{1}{2}x^T G x + x^T c \\ Ax = b \end{cases} \tag{1.2}$$

où A est une matrice de $\mathbb{R}^{m\times n}$ avec $m \leq n$, les ligne de A sont donc les vecteurs a_i, $i \in J = \{1, ..., m\}$ et $b \in \mathbb{R}^m$. En appliquant le théorème de Lagrange (Voir Annexe) donnant une condition nécessaire d'optimalité pour que x^* soit un minimum local, on obtient le système suivant :

$$\begin{bmatrix} G & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}, \tag{1.3}$$

avec $\lambda^* \in \mathbb{R}^m$ est le multiplier de Lagrange. Le système [1.3] peut être récrit dans une forme plus pratique point de vue algorithmique en exprimant x^* comme $x^* = x + p$, où x est une estimation de la solution et p l'étape désirée. En introduisant cette notation, nous obtenons le système

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -p \\ \lambda^* \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix} \tag{1.4}$$

avec

$$h = Ax - b, \qquad g = c + Gx, \qquad p = x^* - x \tag{1.5}$$

La matrice dans [1.4] est appelée la matrice de Karush–Kuhn–Tucker (K-K-T). Le résultat suivant donne une condition suffisante pour que la matrice de K-K-T soit non singulière. Pour se faire, nous utilisons la matrice $Z \in \mathbb{R}^{n\times n-m}$, dont les colonnes forme une base de $\ker A$, d'où Z est de rang plein et $AZ = 0$.

Lemme 1.1 ***Lemme 1.2 (noc)*** *Supposons que A est de rang plein et que la matrice $Z^T GZ$ est définie positive. Alors la matrice de K-K-T*

$$K = \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \tag{1.6}$$

est non singulière, et par conséquent il existe une solution unique (x^*, λ^*) vérifiant [1.3].

Preuve. Supposons qu'il existe deux vecteurs ω et ν tel que

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \omega \\ \nu \end{bmatrix} = 0. \tag{1.7}$$

Vue que $A\omega = 0$, de [1.7] on aura,

$$0 = \begin{bmatrix} \omega \\ \nu \end{bmatrix}^T \begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \omega \\ v \end{bmatrix} = \omega^T G\omega.$$

Puisque $\omega \in \ker\ A$, il peut être écrit pour certain vecteur $u \in \mathbb{R}^{n-m}$, $\omega = Zu$. Par conséquent, nous avons $0 = \omega^T G\omega = u^T Z^T GZu$, comme $Z^T GZ$ est définie positive, ceci implique que $u = 0$, et donc $\omega = 0$, de la relation [1.7], on aura $A^T\nu = 0$. Comme A est de rang plein, alors $\nu = 0$. D'où l'unique solution pour l'équation [1.7] et la solution nulle $\omega = 0$ et $\nu = 0$, donc la matrice de K-K-T est inversible. ■

Exemple 1.1 *Considérons le problème quadratique*

$$\begin{aligned}
\min q(x) &= 3x_1^2 + 2x_1x_2 + x_1x_3 + \frac{5}{2}x_2^2 + 2x_2x_3 + 2x_3^2 - 8x_1 - 3x_2 - 3x_3 \qquad (1.8)\\
x_1 + x_3 &= 3\\
x_2 + x_3 &= 0.
\end{aligned}$$

Nous pouvons écrire ce problème de la forme [1.2] en définissant
$G = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$, $c = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix}$, $A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$.
La solution x^* et le multiplicateur de Lagrange λ^* sont données d'aprés [1.3] par

$$x^* = (2, -1, 1)^T, \qquad \lambda^* = (3, -2)^T.$$

Dans cet exemple, la matrice G est définie positive et la matrice Z dont les vecteurs colonne forment une base de $\ker A$, pour un $x \in \ker A$, on a

$$\begin{cases} x_1 + x_3 = 0 \\ x_2 + x_3 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -x_3 \\ x_2 = -x_3 \end{cases}, \text{ d'où}$$
$$\ker A = \left\{\alpha(-1, -1, 1)^T : \alpha \in \mathbb{R}\right\},$$

alors on peut définir Z comme suit :

$$Z = (-1, -1, 1)^T. \tag{1.9}$$

Alors, $Z^T G Z = 13 > 0$.

Théorème 1.1 ([1]) *Si A est de rang plein et si la matrice Hessienne $Z^T G Z$ est définie positive. Alors le vecteur x^* vérifiant [1.3] est l'unique solution globale de [1.2].*

Preuve. Soit x un point admissible pour [1.2] (vérifiant $Ax = b$). considérons le vecteur p tel que $p = x^* - x$. Puisque $Ax^* = Ax = b$, alors $A(x^* - x) = Ap = 0$. En substituant $x = x^* - p$, dans la fonction objective du problème [1.1], on obtient

$$\begin{aligned}
q(x) &= q(x^* - p) = \frac{1}{2}(x^* - p)^T G(x^* - p) + (x^* - p)^T c \qquad (1.10)\\
&= \frac{1}{2}x^{*T}Gx^* + x^{*T}c + \frac{1}{2}p^T Gp - p^T Gx^* - p^T c\\
&= \frac{1}{2}p^T Gp - p^T Gx^* - p^T c + q(x^*).
\end{aligned}$$

De [1.3] on a $Gx^* = -c + A^T\lambda^*$, et vue que $Ap = 0$, on obtient

$$p^T Gx^* = p^T(-c + A^T\lambda^*) = -p^T c.$$

En remplaçent cette relation dans [1.10], on obtient

$$q(x) = \frac{1}{2}p^T Gp + q(x^*).$$

Vue que $p \in \ker A$, on peut écrire $p = Zu$, pour un certains vecteurs $u \in \mathbb{R}^{n-m}$, ainsi

$$q(x) = \frac{1}{2}u^T Z^T G Z u + q(x^*).$$

Comme $Z^T G Z$ est définie positive, on déduit que $q(x) > q(x^*)$ sauf lorsque $u = 0$, mais dans ce cas on obtient $x^* = x$. Donc x^* est l'unique solution globale pour [1.2]. Si la matrice Hessienne $Z^T G Z$ est semi définie positive, alors le vecteur x^* de [1.3] est un minimum locale mais pas un minimum locale stricte. Si la matrice Hessienne est indéfinie, alors x^* est un point stationnaire. ■

1.1.2 Méthode de résolution

Il existe plusieurs méthodes qui permet d'avoir la solution du système [1.3]. Ici nous considérons une méthode qui ne demande pas l'inversibilité de G, cette méthode s'intitule Espace nul (Null-Space Method). Supposons que les conditions du lemme [1.1] sont vérifiées c.-à-d (A est de rang plein et $Z^T G Z$ est définie positive). Elle nécessite la connaissance de la matrice Z. Cette méthode exploite le découplage du système de K-K-T [1.4] par bloque en deux petits systèmes.

Soit $A \in \mathbb{R}^{m\times n}$, de rang plein avec $m < n$. En faisant cette hypothèse on peut trouver m colonnes linéairement indépendantes. On peut toujours moyennant une matrice de permutation P vérifiant $P^T P = I$, avoir ces m colonnes dans les premières colonnes de la matrice AP, on peut donc écrire

$$AP = [B \ N], \tag{1.11}$$

où $B \in \mathbb{R}^{m\times m}$ une matrice inversible, $N \in \mathbb{R}^{m\times n-m}$ formée des autres $n - m$ colonnes de A.

On définit les sous vecteurs $x_B \in \mathbb{R}^m$ et $x_N \in \mathbb{R}^{n-m}$, comme suit :

$$\begin{bmatrix} x_B \\ x_N \end{bmatrix} = P^T x, \tag{1.12}$$

en écrivant la contrainte $Ax = b$, comme suit :

$$\begin{aligned} b &= Ax = AP(P^T x) = Bx_B + Nx_N \\ &\Rightarrow x_B = B^{-1}b - B^{-1}Nx_N. \end{aligned} \tag{1.13}$$

De [1.12] et [1.13], on obtient

$$\begin{bmatrix} x_B \\ x_N \end{bmatrix} = x = Yb + Zx_N,$$

où

$$Y = \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} B^{-1}N \\ I \end{bmatrix}. \tag{1.14}$$

Notons que Z admet $n - m$ colonnes linéairement indépendants (Vue la présence de la matrice identité) et satisfait $AZ = 0$, par conséquent, Z est une base de $\ker A$, appelée matrice du noyau de A. En plus les colonnes de Y et de Z sont tous linéairement indépendantes, on remarque aussi que Y est une solution particulière de l'équation $Ax = b$.

Considérons à présent le vecteur p dans [1.4] qu'on peut écrire sous la forme :

$$p = Yp_Y + Zp_Z, \tag{1.15}$$

Les matrices Y et Z sont définies comme ci-dessus avec $Z \in \mathbb{R}^{n \times n-m}$ une matrice du noyau de A, Y est toute matrice de $\mathbb{R}^{n \times m}$ telle que $[Y/Z] \in \mathbb{R}^{n \times n}$ est inversible, p_Y est un vecteur à m composantes, et p_Z est un vecteur à $n-m$ composantes.

En remplaçant P dans la deuxième équation de [1.4] et comme $AZ = 0$, on obtient :

$$(AY)p_Y = -h. \tag{1.16}$$

Parce que A est de rang m et $[Y/Z] \in \mathbb{R}^{n \times n}$ est inversible, le produit $A[Y/Z] = [AY/0]$ est de rang m. Donc, AY est une matrice de $\mathbb{R}^{m \times m}$ inversible, et p_Y est déterminé par l'equation de [1.16]. En remplaçant [1.15] dans la première équation de [1.4], on obtient :

$$-GYp_Y - GZp_z + A^T\lambda^* = g,$$

et multiplier par Z^T les deux membres de l'équation on obtient

$$(Z^TGZ)p_Z = -Z^TGYp_Y - Z^Tg. \tag{1.17}$$

Ce système peut être résolue en effectuant une factorisation de Cholesky (voir Annexe) de la matrice Hessienne Z^TGZ pour déterminer p_Z. Donc, on peut calculer $p = Yp_Y + Zp_Z$.

Pour calculer le multiplicateur de Lagrange λ^*, on multiplions la première rangée de blocs de [1.4] par Y^T pour obtenir

$$(AY)^T\lambda^* = Y^T(g + Gp). \tag{1.18}$$

Exemple 1.2 *Considérons le problème [1.8] donnée dans l'exemple [1.2].On peut choisir* $Y = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} \\ \frac{-1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix}$ *et* $Z = (-1, -1, 1)^T$ *d'apres l'exemple [1.2]. Notée que* $AY = I$.

Supposons que $x = (0, 0, 0)^T$ *en [1.5] on a*

$$h = Ax - b = -b, \qquad g = c + Gx = c = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix},$$

on obtient après calcul

$p_Y = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \qquad p_Z = [0],$

alors,

$$p = x^* - x = Yp_Y + Zp_Z = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}.$$

De la relation [1.18], on aura

$$x^* = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \qquad \lambda^* = \begin{bmatrix} 3 \\ -2 \end{bmatrix}.$$

1.2 Programme quadratique avec contraintes inégalité

Considérons à présent la formulation générale d'un problème quadratique avec contraintes égalité et inégalité donné par la relation [1.1]. Le Lagrangien associé à ce le problème est :

$$L(x,\lambda) = \frac{1}{2}x^T G x + x^T c - \sum_{i \in I \cup J} \lambda_i (a_i^T x - b_i). \tag{1.19}$$

L'ensemble des contraintes actives au point x^*, noté $\mathcal{A}(x^*)$ est un ensemble d'indice définie comme suit :

$$\mathcal{A}(x^*) = \left\{ i \in I \setminus a_i^T x^* = b_i \right\} \cup J. \tag{1.20}$$

En appliquant les conditions de K-K-T à ce problème (voir Annexe), on trouve que tout solution x^* de [1.1] est la suite de conditions de premier ordre, pour certain multiplicateurs de Lagrange $\lambda_i^*, i \in \mathcal{A}(x^*)$:

$$Gx^* + c - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* a_i = 0. \tag{1.21}$$

$$a_i^T x^* = b_i, \text{ pour tous } i \in \mathcal{A}(x^*) \tag{1.22}$$

$$a_i^T x^* \geq b_i, \text{ pour tous } i \in I \backslash \mathcal{A}(x^*) \tag{1.23}$$

$$\lambda_i^* \geq 0, \text{ pour tous } i \in I \cap \mathcal{A}(x^*) \tag{1.24}$$

Supposons que la contrainte de l'indépendance linéaire qualification(LICQ) est vérifier dans le problème [1.1].

Remarque 1.1 $\forall i \in \mathcal{A}(x^*)$; *les vecteurs a_i sont linéairement indépendant et les contraintes associé sont linéaires (c'est l'hypothèse de qualification des contraintes).*

Théorème 1.2 ([1]) *Si x^* satisfait les conditions de K-K-T pour un certain λ^*, $i \in \mathcal{A}(x^*)$, et si G est semi définie positive, alors x^* est une solution global pour [1.1].*

Preuve. Soit x une solution admissible pour [1.1] et G est semi définie positive, on à

$$\begin{aligned} a_i^T x &= b_i, \forall i \in J, \text{ et } a_i^T x \geq b_i, \forall i \in \mathcal{A}(x^*) \cap I \\ \text{d'où } a_i^T (x - x^*) &= 0, \forall i \in J, \ \ a_i^T (x - x^*) \geq 0, \forall i \in \mathcal{A}(x^*) \cap I. \end{aligned}$$

de [1.21] et [1.22] on à :

$$\begin{aligned} Gx^* + c - \sum_{i \in I \cup J} \lambda_i^* a_i &= 0 \\ Gx^* + c &= \sum_{i \in I \cup J} \lambda_i^* a_i, \end{aligned}$$

alors

$$(x - x^*)^T (Gx^* + c) = \sum_{i \in J} \lambda_i^* a_i^T (x - x^*) + \sum_{i \in \mathcal{A}(x^*) \cap I} \lambda_i^* a_i^T (x - x^*) \geq 0. \tag{1.25}$$

Par manipulation élémentaire, nous trouvons cela

$$\begin{aligned} q(x) &= q(x^*) + \frac{1}{2}(x - x^*)^T(Gx^* + c) + \frac{1}{2}(x - x^*)^T G(x - x^*) \\ &\geq q(x^*) + \frac{1}{2}(x - x^*)^T G(x - x^*) \\ &\geq q(x^*). \end{aligned}$$

Donc : x^* est un minimum globale. ■

Chapitre 2

La méthode Active-Set

Nous utilisons la méthode Active-Set pour résoudre des programmes quadratiques de la forme [1.1]. Nous considérons seulement le cas convexe, avec la matrice G dans [1.1] est semi définie positive. Si on connait à l'avance l'ensemble des contraintes actives à la solution optimale $\mathcal{A}(x^*)$, la résolution du problème [1.1] est équivalente à la résolution du problème

$$\left\{ \begin{array}{c} \min\limits_{x} q(x) = \frac{1}{2}x^T G x + x^T c \\ a_i^T x = b_i, \; i \in \mathcal{A}(x^*) \end{array} \right. . \tag{2.1}$$

Cependant, l'ensemble $\mathcal{A}(x^*)$ n'est pas toujours connu à l'avance, l'objectif de la méthode Active-Set est de démarrer d'un ensemble $\mathcal{A}(x^{(0)})$ pour un certain point initial $x^{(0)}$ est construire itérativement des ensembles qui convergerai vers $\mathcal{A}(x^*)$. Plus précisément étant à l'itération k et disposant d'une solution admissible x_k, nous définissons l'ensemble

$$W_k = \left\{ i \in \mathcal{A}(x^*) \text{ et } \{a_i\}_{i \in W_k} \text{ sont linéairement indépendants} \right\},$$

W_k est appelé l'ensemble de travail (working set). Nous vérifiant d'abord si le point x_k minimise la fonction objectif q sur W_k. Sinon nous cherchons la direction p solution du problème restreint qu'on va définir ci-dessous, pour se faire écrivons la direction désirée comme suit :

$$p = x - x_k \Rightarrow x = x_k + p.$$

En posant $g_k = Gx_k + c$ et en remplaçant x dans la fonction objectif $q(x)$ du problème donné en [1.1], on trouve

$$\begin{aligned} q(x) &= q(x_k + p) = \frac{1}{2}(x_k + p)^T G (x_k + p) + (x_k + p)^T c \\ &= \frac{1}{2}p^T G p + p^T G x_k + x_k^T c + p^T c + \frac{1}{2}x_k^T G x_k \\ &= \frac{1}{2}p^T G p + p^T g_k + \rho_k, \end{aligned}$$

Avec $\rho_k = \frac{1}{2}x_k^T G x_k + x_k^T c$, puisque ρ_k est indépendant de p, la résolution du problème

$$\left\{ \begin{array}{c} \min\limits_{p} \frac{1}{2}p^T G p + g_k^T p + \rho_k \\ a_i^T p = 0, i \in W_k \end{array} \right.$$

est équivalente au problème appelé problème restreint

$$(\mathcal{P}_k)\begin{cases} \min\limits_{p} \frac{1}{2}p^TGp + g_k^Tp \\ a_i^Tp = 0, i \in W_k \end{cases}, \tag{2.2}$$

appelé problème restreint. Notons par p_k la solution optimale du problème $(\mathcal{P}_k)$. Pour $i \in W_k$, $x = x^k + \alpha p_k$, reste admissible $\forall \alpha \geq 0$. En effet :

$a_i^Tx = a_i^T(x^k + \alpha p_k) = a_i^Tx^k + \alpha a_i^Tp_k = b_i$. On à deux possibilités pour p_k, soit $p_k = 0$ ou $p_k \neq 0$. Supposons que la solution de [2.2] est $p_k \neq 0$. Si $x^k + p_k$ vérifie toutes les contraintes, alors

$$x^{k+1} = x^k + p_k. \tag{2.3}$$

Sinon on cherche un paramètre $\alpha_k \in [0,1]$, telle que $x^{k+1} = x^k + \alpha^k p_k$ soit admissible pour le problème [2.1]. On peut donner une définition explicite pour α_k, en étudiant le comportement des contraintes quand $i \notin W_k$, puisque pour $i \in W_k$ les contraintes seront satisfaites vu le choix de α_k. Pour $i \notin W_k$, on à deux possibilités soit $a_i^Tp_k \geq 0$ ou $a_i^Tp_k < 0$.

– Si $a_i^Tp_k \geq 0$, on à

$$\begin{aligned} a_i^T(x^{k+1}) &= a_i^T(x^k + \alpha^k p_k), \\ &= a_i^Tx^k + \alpha^k a_i^Tp_k \\ &\geq a_i^Tx^k \\ &\geq b_i \end{aligned}$$

– Si $a_i^Tp_k < 0$, on à :

$$\begin{aligned} a_i^T(x^k + \alpha^k p_k) &\geq b_i \quad \text{si et seulement si} \\ \alpha^k &\leq \frac{b_i - a_i^Tx^k}{a_i^Tp_k};\ i \notin W_k \text{ et } a_i^Tp_k < 0. \end{aligned}$$

On définie α^k par :

$$\alpha^k = \min(1, \min_{\substack{i \notin W_k \\ \text{et } a_i^Tp_k<0}} \frac{b_i - a_i^Tx^k}{a_i^Tp_k}). \tag{2.4}$$

S'il existerai $i_0 \notin W_k$ tel que

$$\alpha^k = \frac{b_{i_0} - a_{i_0}^Tx^k}{a_{i_0}^Tp_k},$$

alors la contrainte i_0 est appelé contrainte bloquante. Le paramètre α^k peut prendre tout les valeurs de l'intervalle $[0,1]$. S'il prendra la valeur 0, ceci signifie qu'il y à une contrainte active qui n'a pas encore été introduite dans W_k. Si aucune contrainte bloquante n'a été trouvé α^k prendra la valeur 1. Dans cette étape de l'algorithme, x^k sera mise à jour

$$x^{k+1} = x^k + \alpha^k p_k.$$

Si une contrainte bloquante a été trouvé W_k, se mettra à jour tel que

$$W_{k+1} = W_k \cup \{i_0 \text{ l'indice de la contrainte bloquante}\}.$$

On continuera à itérer en rajoutant des indices chaque fois qu'une contrainte bloquante sera rencontrée. Jusqu'à avoir une solution $\widehat{x}$ qui minimisera la fonction objectif q sur son ensemble de travail $\widehat{W}$, il est simple de reconnaitre un tel point car le problème restreint aura forcément la solution nulle $p = 0$. Puisque, la solution nulle $p = 0$, satisfait la condition d'optimalité (1.4) pour le problème (2.2) on aura donc

$$g_k - \sum_{i \in \hat{W}} \hat{\lambda}_i a_i = 0, \tag{2.5}$$

avec les $\{a_i\}_{i \in W_k}$ sont linéairement indépendants. Quand on résoudre ce système on aura deux situation

- Soit $\hat{\lambda}_i \geq 0, \forall\, i \in \hat{W} \cap I$, les condition de K-K-T seront donc tous satisfait et la solution trouvée est optimale.
- Si $\exists\ j$ avec $j = \arg\min_{i \in \hat{W} \cap I} \left\{ \hat{\lambda}_i \ / \ \hat{\lambda}_i < 0 \right\}$, alors W_{k+1} sera mise à jour tel que $W_{k+1} = W_k / \{j\}$.

Algorithm 2.1 (Active-Set pour QP Convexe) *Initialisation : Calculer un point initial x_0. Poser W_0 l'ensemble de travail initial.*

Pour $k = 1, 2, ...$Résoudre le problème restreint [2.2] :
Si $p_k = 0$
calculer $\hat{\lambda}_i$ vérifiant [2.5], avec $\hat{W} = W_k$
Si $\hat{\lambda}_i \geq 0$, $\forall i \in \hat{W} \cap I$ stop la solution est $x^* = x$;
Sinon
$j = \arg\min_{i \in \hat{W} \cap I} \left\{ \hat{\lambda}_i \ / \ \hat{\lambda}_i < 0 \right\}$
$W_{k+1} = W_k / \{j\}$
$x^{k+1} = x^k$
Fin de Si
Sinon ($p_k \neq 0$)
Calculer α_k par [2.4]
Calculer $x^{k+1} = x^k + \alpha_k p_k$
Si il y à une contrainte bloquante, on obtient
$W_{k+1} = W_k \cup \{\textit{l'indice de la contrainte bloquante}\}$
Sinon
$W_{k+1} = W_k$
Fin de Si
Fin de Si
$k = k + 1$
Fin de Pour

Exemple 2.1

$$\begin{array}{rcll} \min\limits_{x} q(x) & = & (x_1 - 1)^2 + (x_2 - 2.5)^2 & \\ x_1 - 2x_2 + 2 & \geq & 0 & \textit{(1)} \\ -x_1 - 2x_2 + 6 & \geq & 0 & \textit{(2)} \\ -x_1 + 2x_2 + 2 & \geq & 0 & \textit{(3)} \\ x_1 & \geq & 0 & \textit{(4)} \\ x_2 & \geq & 0 & \textit{(5)} \end{array} \tag{2.6}$$

$J = \{1,2,3,4,5\}, \quad x^0 = (2,0)^T, \quad W^0 = (3,5)^T$

$$A = \begin{bmatrix} 1 & -2 \\ -1 & -2 \\ -1 & 2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, b = \begin{bmatrix} -2 \\ -6 \\ -2 \\ 0 \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} -2 \\ -5 \end{bmatrix}.$$

Résoudre le problème [2.6] pour $k = 0$:

$$\nabla q(x) = \begin{bmatrix} 2(x_1 - 1) \\ 2(x_2 - 2.5) \end{bmatrix}; G = \nabla^2 q(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix};$$

– 1. (a) **Exemple 2.2** $g_0 = Gx_0 + c = \begin{bmatrix} 2 \\ -5 \end{bmatrix}$.

$$\begin{bmatrix} G & -A_{W_k}^T \\ A_{W_k} & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_3 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} -g_0 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & -2 & -1 \\ -1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \lambda_3 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} -2 \\ 5 \\ 0 \\ 0 \end{bmatrix},$$

solution donne : $p^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Résoudre le système

$$\begin{aligned} \sum_{i \in W_k} \hat{\lambda}_i a_i &= g_k \Leftrightarrow \hat{\lambda}_3 a_3 + \hat{\lambda}_5 a_5 = g_0 \\ &\Leftrightarrow \begin{bmatrix} -1 \\ 2 \end{bmatrix} \hat{\lambda}_3 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \hat{\lambda}_5 = \begin{bmatrix} 2 \\ -5 \end{bmatrix} \\ &\Leftrightarrow (\hat{\lambda}_3, \hat{\lambda}_5) = (-2, -1), \end{aligned}$$

$$\begin{aligned} j &= \arg \min_{i \in W_k \cap I} \{\lambda_i \;\; / \;\; \lambda_i < 0\} \\ &= 3, \end{aligned}$$

donc
$x^1 = x^0 = (2,0)^T$ *et* $W^1 = W^0 / \{3\} = \{5\}$.
Résoudre le problème [2.6] pour $k = 1$:

$g_1 = g_0 = \begin{bmatrix} 2 \\ -5 \end{bmatrix}$.

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix},$$

on aura $p^1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \neq 0.$

D'aprés la formule [2.4], on à $\alpha^1 = 1$ *et la nouvelle contrainte est* $x^2 = x^1 + p^1 = (1, 0)^T$.

Alors, Il n'y a pas de contraintes bloquante, et $W^2 = W^1 = \{5\}$.

Pour l'itération 2*, on trouve la solution de [2.6] est :* $p^2 = 0$*. De [2.5] nous déduisons que le multiplicateur de Lagrange est* $\lambda_5 = -5 < 0$*, donc* 5 *sort de l'ensemble active, on obtient* $W^3 = \emptyset$ *et* $x_3 = x_2 = (1{,}0)^T$*. On continue les calcules de même manière, on obtient :*

Pour $k = 3$*, on à :* $p^3 = (0{,}2.5)^T$, $\alpha^3 = 0.6$, $W^4 = \{1\}$, $x^4 = (1, 1.5)^T$.

Pour $k = 4$*, on à :* $p^4 = (0.4{,}0.2)^T$, $\alpha^4 = 1$ *(pas de contrainte bloqué),* $W^5 = W^4 = \{1\}$, $x^5 = (1.4{,}1.7)^T$.

Pour $k = 5$*, on à :* $p^5 = 0$, $\hat{\lambda}_1 = 0.8 > 0$*, donc* $x^* = (1.4{,}1.7)^T$.

Chapitre 3

La factorisation QR appliqué à la méthode Active-set

On a vu que l'étape de programmation dans la méthode Active-Set donnée par l'algorithme [2.1] qui exige une solution avec contrainte égalité du problème [2.2]. On l'a déjà mentionné dans le chapitre 1, ces calcules montrent qu'on va résoudre un système de K-K-T [1.4]. Le sous ensemble W_k peut être changé par un indice à chaque itération, et la matrice de K-K-T se diffère par une ligne et une colonne par rapport à la précédente itération. Autrement, G reste fixé, mais la matrice A de contraintes gradients correspondant à W_k peut être changé avec une addition et/ou suppression d'une ligne. Cette observation nous montrent qu'on peut calculer la matrice des facteurs nécessitants pour résoudre le problème [2.2] dans l'itération actuel par une mise à jour des facteurs dans la dernière itération tous les techniques de mise à jour sont cruciales pour prouver l'efficacité de la méthode Active-Set.

On a limité notre discussion que pour le cas où l'étape est calculé avec la méthode d'espace nul [1.15] et [1.18]. Supposons que A est de m lignes linéairement indépendantes et assumons que les bases Y et Z (pour plus de détails voir méthode de résolution) sont définies avec les moyens de factorisation QR de A. Alors

$$A^T\pi = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}\begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{3.1}$$

où π est une matrice de permutation, $R \in \mathbb{R}^{n\times n}$ triangulaire supérieure et inversible, $Q = [Q_1\ Q_2]$ est $n \times n$ orthogonal, et Q_1 et R ont m colonnes où moment on Q_2 est de $n-m$ colonnes (pour plus de détails voir Annexe), on peut choisir Z peut être tous simplement la matrice orthonormal Q_2.

On suppose qu'une contrainte est additionnée avec W_k dans la prochaine itération, alors la nouvelle matrice contrainte est

$$\bar{A}^T = \begin{bmatrix} A^T & a \end{bmatrix},$$

où a est un vecteur colonne de longueur n avec $\bar{A}^T$ est de rang plein. So, on a une façon économique pour faire la mise à jour des facteurs Q et R de [3.1] pour obtenir un nouveau facteur (parce qu'on a une nouvelle matrice de base d'espace nul $\bar{Z}$ est de $n-m-1$ colonnes) pour la matrice $\bar{A}$. on remarque premièrement que

$$Q_1Q_1^T + Q_2Q_2^T = I,$$

on a

$$\bar{A}^T \begin{bmatrix} \pi & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A^T\pi & a \end{bmatrix} = Q \begin{bmatrix} R & Q_1^T a \\ 0 & Q_2^T a \end{bmatrix}. \tag{3.2}$$

On peut maintenant définir la matrice orthogonal $\hat{Q}$ qui transforme le vecteur $Q_2^T a$ à un vecteur dans laquelle tous les éléments sauf le premier sont zéro. Alors, on a

$$\hat{Q}(Q_2^T a) = \begin{bmatrix} \gamma \\ 0 \end{bmatrix},$$

où γ est un scalaire (comme $\hat{Q}$ est orthogonal, alors $\left\|Q_2^T a\right\| = |\gamma|$). De la forme [3.2], on à

$$\bar{A}^T \begin{bmatrix} \pi & 0 \\ 0 & 1 \end{bmatrix} = Q \begin{bmatrix} R & Q_1^T a \\ 0 & \hat{Q}^T \begin{bmatrix} \gamma \\ 0 \end{bmatrix} \end{bmatrix} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \\ 0 & 0 \end{bmatrix}.$$

C'est la factorisation de la forme

$$\bar{A}^T \pi = \bar{Q} \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix},$$

où

$$\bar{\pi} = \begin{bmatrix} \pi & 0 \\ 0 & 1 \end{bmatrix}, \bar{Q} = Q \begin{bmatrix} I & 0 \\ 0 & \hat{Q}^T \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2\hat{Q}^T \end{bmatrix}, \bar{R} = \begin{bmatrix} R & Q_1^T a \\ 0 & \gamma \end{bmatrix}.$$

On peut donc choisir $\bar{Z}$ contient les dernières $n-m-1$ colonnes de $Q_2\hat{Q}^T$. Si on connaît Z explicitement et on veut une représentation explicite de $\bar{Z}$. On a besoin de calcul pour obtenir $\hat{Q}$ et le coût pour former le produit $Q_2\hat{Q}^T = Z\hat{Q}^T$, puisque la structure est spéciale à $\hat{Q}$, si le coût est d'ordre $n \times n - m$, on a déjà comparé avec [3.1] à partir de zéro qui est d'ordre $n^2 m$.

La stratégie de mise à jour est moins chère, surtout quand l'espace nul est petit (c'est, quant $n - m < n$).

La technique de mise à jour peut être aussi appliquée pour le cas où une ligne est enlève de A. Cette opération à l'effet de supprimer une colonne de R dans [3.1], donc on dérange la propriété de la triangularité supérieur de cette matrice on introduit donc un nombre différent de zéro sur la diagonale spécialement sur la diagonale principale de la matrice. La propriété de la triangularité supérieure peut être restaurée avec l'application d'une séquence de rotations planes. Ces rotations introduisent un nombre de transformations moins chères dans les m premières colonnes de Q, et la mise à jour d'une matrice d'espace nul est obtenue par choisir la dernière $n-m+1$ colonnes de cette matrice après que les transformations sont complètes.

La nouvelle base d'espace nul dans ce cas est de la forme

$$\bar{Z} = \begin{bmatrix} \bar{z} & Z \end{bmatrix}, \tag{3.3}$$

donc, c'est la matrice actuelle Z augmentée par une seule colonne. Le coût total de cette opération varie en fonction de l'emplacement de la colonne enlevée dans A mais est en général on recalcule une factorisation QR à partir de zéro. Maintenant, considérons la matrice Hessienne. Puisque la forme spéciale de [2.2], on a $h = 0$ dans [1.4], et l'étape p_Y

donnée dans [1.16] est zéro. Alors de [1.17], l'espace nul qui se compose de p_Z est la solution de

$$(Z^T G Z) p_Z = -Z^T g. \tag{3.4}$$

On peut parfois trouver des manières pour faire la mise à jour de la factorisation de la matrice Hessienne $Z^T G Z$ après que Z est change.

Supposons qu'on a la factorisation de cholesky (voir Annexe) de la matrice Hessienne actuelle, qui est écrite sous la forme suivante

$$Z^T G Z = L L^T,$$

et que dans le prochain changement de Z comme dans [3.3], on va gagner une colonne après la suppression d'un contrainte. Une série d'opérations élémentaires peut être utilisée pour le facteur de transformée de cholesky L dans le nouveau facteur $\bar{L}$ pour la nouvelle matrice Hessienne $\bar{Z}^T G \bar{Z}$. Une d'autre variété de simplification est possible.

Conclusion

Il est à signaler, que faute de temps nous avons implémenté uniquement La méthode Active-Set de base. L'implémentation de la méthode avec la technique considérée en dernier chapitre peut améliorer considérablement le comportement de la méthode point de vue numérique. Le problème d'initialisation n'a pas été abordé, dans ce cadre les techniques d'initialisation de la méthode du Simplexe peuvent être utiles. Cependant, même avec la méthode de base les résultats obtenus étaient très satisfaisantes. Indiquant aussi qu'uniquement, le cas convexe a été considéré, le cas non convexe semble être une bonne perspective.

Bibliographie

[1] J. Nocedal and S.J. Wright. Numerical Optimization. Springer Series in Operations Research. Springer, 1999.

[2] Edwin K.P.Chong and Stanislaw H.Żak. An Introduction to Optimization. Wiley Interxience Series in Discrete Mathematics and Optimization. Wiley, 2001.

[3] A. Quarteroni and R. Sacco and F. Sleri. Méthode Numériques (Algorithmes, analyse et application). Springer, 2004.

[4] http ://www.finance-investissement.com/l-optimisation-a-la-rescousse-des-fonds.

[5] https ://www.lri.fr/perso/~antoine/Papers/SVM-synthese.pdf.

Annexes :

3.1 Factorisations des matrices : Cholesky, LU, QR

Les méthodes de factorisation des matrices sont importantes dans le dessin d'algorithmes et dans leurs analyses.

Tous les algorithmes de la factorisation décris ci-dessous utilisent une matrice de permutation. Supposons que, l'on change la première et la quatrième ligne d'une matrice A. Pour faire cette opération en multipliant à gauche A par une matrice de permutation p, qui est construite par permutation avec la première et la quatrième ligne d'une matrice d'identité qui contient le même nombre de lignes que A. Par exemple, supposons, que A est une matrice de 5×5. Le choix approprié de p sera

$$p = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Une technique semblable est utilisée pour trouver une matrice de permutation p en échangeant les colonnes d'une matrice.

La factorisation LU d'une matrice $A \in \mathbb{R}^{n \times n}$ est définie par

$$pA = LU,$$

où $p \in \mathbb{R}^{n \times n}$ est une matrice de permutation (qui permuté les lignes d'une matrice identité de $\mathbb{R}^{n \times n}$).

L est une matrice triangulaire inférieure avec des éléments diagonaux égaux à 1, et U est triangulaire supérieur. On utilise cette factorisation pour résoudre des systèmes linéaires de la forme $Ax = b$.

Définition 3.1 *Soit $A \in \mathbb{R}^{n \times n}$ une matrice symétrique, définie positive, il est possible de calculer une factorisation, noté la factorisation de cholesky, produit d'une matrice L telle que*

$$A = LL^T, \tag{3.5}$$

avec L un élément diagonaux positive.

La factorisation de cholesky d'une matrice symétrique définie positive sans changer les lignes aux colonnes. Mais, pour la permutation symétrique (permuté le même nombre de lignes et le même nombre de colonnes) on à

$$p^T AP = LL^T.$$

Pour une matrice de permutation p. La factorisation de cholesky peut être utilisée pour calculer la solution du système $Ax = b$.

Définition 3.2 *La factorisation QR est une factorisation pour les matrices rectangulaires $A \in \mathbb{R}^{m\times n}$ est de la forme*

$$Ap = QR, \tag{3.6}$$

où $p \in \mathbb{R}^{n\times n}$ est une matrice de permutation.
$A \in \mathbb{R}^{m\times m}$ est une matrice orthogonale.
$R \in \mathbb{R}^{m\times n}$ est une matrice triangulaire supérieure.

Dans le cas d'une matrice carrée, on utilise la factorisation LU par élimination gaussienne, et dans le cas d'une matrice rectangulaire A avec $m < n$, on peut utiliser la factorisation QR de A^T, on trouve une matrice avec les colonnes de cette matrice contient $\ker A$, et on écrit

$$A^T p = QR = [Q_1\ Q_2]\, R,$$

où Q_1 consiste les m premier colonnes de Q, et Q_2 contient les $n - m$ dernier colonnes de Q. C'est facile de montrer que les colonnes de la matrice Q_2 contiennent $\ker A$ (les colonnes de Q_2 sont orthogonales).

En particulier dans le cas où A est mini, comme A est de rang plein, on fait une identification entre le facteur R dans [3.6] et la factorisation de cholesky. En multipliant la formule [3.6] par sa transposée, on obtient

$$p^T A^T Ap = R^T Q^T QR = R^T R,$$

et par comparaison avec [3.5], on remarque que R^T est le facteur de cholesky de la matrice symétrique définie positive $p^T A^T Ap$.

3.2 Conditions d'optimalité

Théorème 3.1 (Théorème de Lagrange [2]) *soit x^* un minimum local de $f : \mathbb{R}^n \to \mathbb{R}$ avec $h(x) = 0$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $m \leq n$. Supposons que x^* est un point régulier (i.e. $\{\nabla g(x)\}$ et $\{\nabla h(x)\}$ sont linéairement indépendant). Alors il existe $\lambda^* \in \mathbb{R}^m$ telle que*

$$\begin{aligned} \nabla L(\lambda^*, x^*) &= 0^T \\ &\Leftrightarrow \nabla f(x^*) + \lambda^{*T} \nabla h(x^*) = 0^T, \end{aligned}$$

avec, la fonction de Lagrange est de la forme

$$L(\lambda, x) = f(x) + \lambda^T h(x).$$

Théorème 3.2 (Théorème de K-K-T [2]) *Soient f, h, $g \in \mathbb{C}^1$, x^* est un point régulier et est un minimum local pour le problème de minimisation f avec $h(x) = 0$, $g(x) \leq 0$. $\exists \lambda^* \in \mathbb{R}^m$ et $\mu^* \in \mathbb{R}^p$ tel que*

$* \ \mu^* \geq 0.$

$*$ $\nabla f(x^*) + \lambda^{*T}\nabla h(x^*) + \mu^{*T}\nabla g(x^*) = 0^T$.
$*$ $\mu^{*T}g(x^*) = 0$.

Avec λ^* *est le vecteur multiplier de Lagrange et* μ^* *est le vecteur multiplier de K-K-T. Pour* $\mu_j^* \geq 0$ *et* $g_j(x^*) \leq 0$ *(condition 1), on à les conditions*

$$\mu^{*T}g(x^*) = \mu_1^{*T}g_1(x^*) + ... + \mu_P^{*T}g_P(x^*) = 0,$$

implique que si $g_j(x^*) < 0$, *alors* $\mu_j^* = 0$, *pour toutes* $j \notin J(x^*)$.