

Remerciements

*Je remercie **Allah** le tout puissant de m'ouvrir les portes du savoir, de m'avoir aidé dans les moments difficiles et de m'avoir permis d'être ce que je suis devenu.*

*Mes remerciements éternels s'adressent à mon père et ma chère mère pour leur présence permanente, leur soutien, leur affection et leurs prières.
Sans oublier mon frère Mouloud, mes soeurs et ma grand-mère.*

Au terme de ce travail, je tiens à exprimer ma profonde gratitude à Monsieur Ouanes, mon directeur de thèse, pour l'intérêt qu'il a porté à mon travail. Par ses précieux conseils. Grâce à lui, j'ai pu réaliser ce travail. Je lui en suis très reconnaissant.

Je remercie vivement l'ensemble des membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Encore et avec une grande fierté et honneur que je tiens à présenter mes remerciements à toute la famille administrative du département des mathématiques, à tous ces enseignants qui m'ont permis d'acquérir des connaissances.

Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste mémoire.

*L'imagination est plus importante que la connaissance(Le savoir).
Car la connaissance est limitée, tandis que l'imagination englobe
le monde entier, stimule le progrès, suscite l'évolution.
Albert Einstein.*

Table des matières

Préambule	7
Préliminaires	10
Introduction générale	11
1 Éléments d'analyse convexe et conditions d'optimalité	13
1.1 Convexité	14
1.1.1 Ensemble convexe	14
1.1.2 Fonction convexe	15
1.2 Définitions	16
1.2.1 Un algorithme	16
1.2.2 Formelle d'un algorithme	17
1.2.3 La notion d'algorithme itératif	17
1.2.4 La complexité	17
1.2.5 La classe N-P difficile	17
1.2.6 Point limite	18
1.2.7 Direction de descente	18
1.2.8 Point stationnaire	18
1.2.9 Minima locaux et globaux	19
Minimum local	19
Minimum global	19
1.3 Conditions d'optimalité	20
1.3.1 Pourquoi avons-nous besoin de conditions d'optimalité? . .	20
1.3.2 Conditions nécessaires	21
Condition nécessaire du premier ordre	21
Condition nécessaire du second ordre	21
1.3.3 Conditions suffisantes	22
Condition suffisante du second ordre	22
Condition suffisante d'optimalité globale	23
1.3.4 Condition d'optimalité pour les problèmes quadratiques . .	24
Fonction quadratique	24
Condition d'optimalité pour les problèmes quadratiques . .	24

1.4	Le problème	25
2	Méthodes de pénalité intérieure	27
2.1	Vue d'ensemble	28
2.1.1	Définition	29
2.1.2	Proposition	30
2.2	Les méthodes de points intérieurs	30
2.2.1	Méthodes affines (Dikin 1967)	31
2.2.2	Méthodes de réduction du potentiel	31
2.2.3	Méthodes de trajectoire centrale (T.C)	32
2.3	Méthode de Karmarkar	32
2.3.1	Description de l'algorithme de base	32
2.3.2	Algorithme de base	34
2.3.3	Convergence de l'algorithme	35
2.4	Généralisation de l'algorithme de base	36
3	Méthode de la fonction barrière	39
3.1	Le principe	40
3.2	La fonction barrière	40
4	Algorithme et exemples d'application	52
4.1	La méthode	53
4.2	L'idée	56
4.3	L'algorithme	56
4.4	Application de la méthode sur certains problèmes	60
4.4.1	Problème 1	60
4.4.2	Problème 2	61
4.4.3	Problème 3: Recherche de clique maximum dans un graphe	61
4.5	Langage de programmation	64
4.6	Exemples d'application	65
	Conclusion générale	82
	Bibliographie	83

Table des figures

1.1	<i>Ensemble convexe et non convexe</i>	12
1.2	<i>Fonction convexe</i>	13
1.3	<i>Fonction concave</i>	14
1.4	<i>Direction de descente (Gradient à pas optimal)</i>	16
1.5	<i>Minima locaux et globaux</i>	18
3.1	<i>La surface de $f(x)$</i>	39
3.2	<i>La surface de $e(x,50)$</i>	40
3.3	<i>La surface de $e(x,25)$</i>	41
3.4	<i>La surface de $e(x,15)$</i>	41
3.5	<i>La surface de $e(x,5)$</i>	42
4.1	<i>L'interface principale de l'algorithme</i>	65
4.2	<i>Résultat de l'exemple 1</i>	66
4.3	<i>Résultat de l'exemple 2</i>	68
4.4	<i>Calcul de c</i>	69
4.5	<i>Résultat de l'exemple 3</i>	71
4.6	<i>Le graphe $G_1 = (V_1, E_1)$</i>	72
4.7	<i>Le graphe $\overline{G_1}$</i>	73
4.8	<i>Résultat de l'exemple 4</i>	75
4.9	<i>La clique maximum du graphe $G_1 = (V_1, E_1)$</i>	76
4.10	<i>Le graphe $G_2 = (V_2, E_2)$</i>	77
4.11	<i>Le graphe $\overline{G_2}$</i>	78
4.12	<i>Résultat de l'exemple 5</i>	80
4.13	<i>La clique maximum du graphe $G_2 = (V_2, E_2)$</i>	81

Liste des tableaux

3.1	<i>Exécution de la méthode de barrière pour l'exemple 1</i>	66
3.2	<i>Exécution de la méthode de barrière pour l'exemple 2</i>	67-68
3.3	<i>Exécution de la méthode de barrière pour l'exemple 3</i>	70
3.4	<i>Exécution de la méthode de barrière pour l'exemple 4</i>	74-75
3.5	<i>Exécution de la méthode de barrière pour l'exemple 5</i>	79-80

Préambule

“La recherche opérationnelle est une attitude scientifique et une vision
Mathématique devant les phénomènes d’organisation”
(A.Kaufman).

1 . Historique :

Dès le *XVII^{ème}* siècle, des mathématiciens comme *Blaise Pascal*^[22] tentent de résoudre des problèmes de décisions dans l’incertain avec l’espérance mathématique. D’autres, au *XVIII^{ème}* et *XIX^{ème}* siècle, s’intéressaient à résoudre les problèmes combinatoires. Au début du *XX^{ème}* siècle, l’étude de la gestion de stock était à l’origine de la recherche opérationnelle moderne avec la formule du lot économique proposée par *Harris*^[22] en 1913.

Mais ce n’est qu’avec la seconde guerre mondiale que le domaine de la R.O était en exploitation. Cependant, en 1940 *Patrick Blackett*^[22] est appelé par l’état-major anglais à diriger la première équipe de recherche opérationnelle, pour résoudre certains problèmes, tels que l’implantation optimale de radars de surveillance.

Le mot *Opérationnelle* vient donc du fait que la première application d’un groupe de travail organisé dans cette discipline avait trait aux opérations militaires. Le nom est resté par la suite, même si le domaine militaire n’est plus le principal champ d’application de cette pratique.

Après la guerre, les techniques se sont considérablement développées, grâce notamment à l’explosion des capacités de calcul des ordinateurs, mais aussi à la multiplicité des domaines d’application.

2 . Types de problèmes traités par la recherche opérationnelle :

La recherche opérationnelle peut aider le décideur lorsque celui-ci est confronté à un problème combinatoire, aléatoire ou concurrentiel.

- Un problème est dit **combinatoire**, lorsqu’il comprend un grand nombre de solutions admissibles, parmi lesquelles on cherche une solution optimale ou proche de l’optimum.
- Un problème est dit **aléatoire**, s’il consiste à trouver une solution optimale face à un problème qui se pose en termes incertains.
- Un problème est dit **concurrentiel**, s’il consiste à trouver une solution optimale face à un problème dont les termes dépendent de l’interrelation entre ses propres agissements et ceux d’autres décideurs.

3 . Applications pratiques :

Les différentes applications utilisées dans le domaine de la *R.O* sont comme suit:

La gestion de projets :

travaux d'ordonnancement, logistique, planification et problèmes d'emploi du temps.

Les problèmes d'investissements :

ils consistent en général à maximiser le profit obtenu en investissant son argent.

Le domaine de l'énergie :

elle est couramment utilisée dans l'industrie pétrolière.

Le domaine de l'informatique :

choix de la localisation et du nombre de serveurs à mettre en place, de la capacité de stockage, de la puissance de calcul et du débit du réseau, en d'autre terme, c'est le choix d'une architecture matérielle informatique. Malgré son importance intrinsèque, la *R.O* est encore peu utilisée dans le monde industriel soit à cause du manque d'information ou de formation des décideurs, soit par le manque de pertinence de l'outil ou sa difficulté de mise en œuvre.

4 . Principales méthodes :

Il existe plusieurs méthodes permettant de résoudre les problèmes traités par la *R.O*, à savoir :

Processus stochastique:

concernent tous les problèmes aléatoires, en particulier les problèmes de fiabilité et les phénomènes d'attente.

Simulation informatique:

est souvent employée pour résoudre des problèmes de *R.O*, notamment dans le milieu non académique.

Programmation linéaire ou non linéaire:

sont très souvent utilisées pour résoudre des problèmes combinatoires.

Méthodes arborescentes:

les méthodes de type séparation et évaluation sont couramment utilisées pour trouver la solution exacte d'un problème de recherche opérationnelle. Pour une résolution exacte, un soin particulier est apporté au calcul de bornes supérieures ou inférieures pour la valeur de la solution.

Heuristiques et méta-heuristiques :

lorsque la solution optimale ne peut être obtenue en temps raisonnable, on a souvent recours à des méthodes approchées de type heuristique ou méta-heuristique.

Préliminaires

Définition 1 : Boules ouvertes et ensembles ouverts

Soit x^0 un point de \mathbb{R}^n et $r > 0$ un nombre réel donné, strictement positif. L'ensemble des points x de \mathbb{R}^n tels que : $\|x - x^0\| < r$, est appelé une n -boule ouverte de rayon r et de centre x^0 . On la note $B(x_0; r)$.

Un exemple est donné en dimension une par un intervalle ouvert de centre x^0 . Dans \mathbb{R}^2 , nous retrouvons le disque circulaire ouvert de centre x^0 et de rayon r . Dans \mathbb{R}^3 , c'est la boule usuelle ouverte de centre x^0 et de rayon r .

Définition 1.1 :

Soit S un sous-ensemble de \mathbb{R}^n et soit $x^0 \in S$. Alors, x^0 est appelé *un point intérieur* de S s'il existe une n -boule ouverte de centre x^0 , tous ses points appartenant à S . L'ensemble de tous les points intérieurs de S est appelé *l'intérieur de S* et il est noté $intS$.

Un ouvert contenant un point x^0 est appelé *un voisinage* de x^0 .

Définition 1.2 :

Un ensemble S de \mathbb{R}^n est appelé *ouvert* si tous ses points sont des points intérieurs. En d'autres termes, si et seulement si $S = intS$.

Définition 2 : Un ellipsoïde

Considérons le problème suivant:

$$\begin{cases} \min f(x) \\ x \in E, \end{cases}$$

où $x \in \mathbb{R}^n$ et E un *ellipsoïde* défini par son centre a^0 , par r et la matrice symétrique définie positive A .

Algébriquement : $E = \{x \in \mathbb{R}^n : (x - a^0)A(x - a^0) \leq r^2\}$.

Définition 3 : Un box

Un domaine D est dit *box* s'il est défini de la manière suivante :

$$\mathbb{D} = \{x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n / l_i \leq x_i \leq u_i, \quad i = \overline{1, n}\}.$$

Introduction générale

La résolution d'un problème non linéaire de taille importante reste toujours un challenge malgré tout le succès observé dans ce domaine.

Le relancement des méthodes de points intérieurs depuis 1984 a donné espoir aux spécialistes du domaine pour réaliser des progrès considérables dans ce sens. Ces méthodes ont fait leurs preuves dans le domaine de la programmation linéaire notamment par leurs bonnes propriétés théoriques (complexité polynômiale et convergence superlinéaire) et leur bon comportement numérique. Aussitôt, des variantes sont introduites pour la programmation quadratique et la complémentarité linéaire.

Ceci étant, il faut signaler tout de même que (sur la plan technique), ces méthodes présentent des inconvénients d'ordre algorithmique et numérique entre autre : Le problème d'initialisation et le coût excessif de l'itération lié aux choix de la direction et du pas de déplacement.

Le travail consiste à approcher la solution des problèmes d'optimisation quadratiques non convexes définis sur un box avec une méthode de la fonction barrière, qui essaye de produire une solution de bonne qualité en suivant un chemin dans une direction de descente pendant que le paramètre de barrière diminue.

Notre étude est répartie en quatre chapitres :

Chapitre 1 : Eléments d'analyse convexe et conditions d'optimalité : Ce chapitre se veut être, un aperçu panoramique sur les différentes notions et techniques à lesquelles nous avons eu recours dans l'implémentation des méthodes d'optimisation. Nous procédons donc, par une représentation progressive allant de la notion de convexité jusqu'à l'exposition du problème d'optimisation à résoudre.

Chapitre 2 : Méthodes de pénalité intérieure : Dans cette partie nous allons présenter les différentes méthodes de points intérieurs ainsi un aperçu général sur l'algorithme de *Karmarkar* qui est à l'origine de ces méthodes.

Chapitre 3 : Méthode de la fonction barrière : Dans ce chapitre nous allons présenter une fonction barrière pour la résolution des problèmes quadratiques non convexes définis sur un box ainsi les différents résultats tirés de cette fonction.

Chapitre 4 : Algorithme et exemples d'application : Dans cette partie nous al-

lons présenté l'algorithme de résolution et son application sur certains problèmes. Et nous le finalisons par des exemples numériques.

Chapitre 1

Eléments d'analyse convexe et conditions d'optimalité

◁ Se donner du mal pour les petites choses, c'est
parvenir aux grandes, avec le temps ▷
Samuel Beckett.

Introduction

Pour étudier les problèmes d'optimisation, il est nécessaire de recourir à des outils spécifiques dont l'étude est basée sur l'analyse convexe. En effet, l'hypothèse de convexité va jouer un rôle très important pour l'algorithme que nous décrirons, la convergence vers un optimum ne pourra être démontrée qu'avec ces hypothèses.

Dans la première section de ce chapitre, nous introduirons rapidement quelques éléments de l'analyse convexe requis pour l'étude de l'optimisation. Ensuite, nous aborderons les conditions nécessaires et suffisantes d'optimalité pour la résolution de ce type de problème. Et enfin, nous exposerons le problème d'optimisation à résoudre.

1.1 Convexité :

Nous allons ici rappeler brièvement les définitions de quelques notions importantes auxquelles nous ferons appel par la suite.

1.1.1 Ensemble convexe

Soit l'ensemble $C \subseteq \mathbb{R}^n$. C est convexe si :

$$\lambda x + (1 - \lambda)y \in C \quad \forall x, y \in C, \quad \forall \lambda \in [0, 1].$$

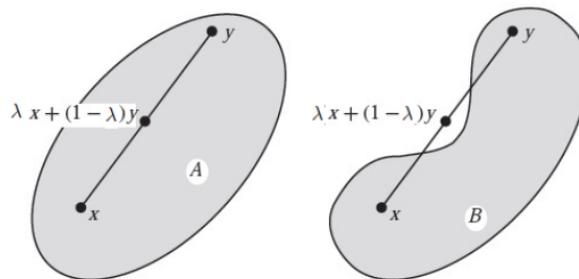


Figure 1.1- Ensemble convexe et non convexe

1.1.2 Fonction convexe

Soit l'ensemble convexe : $C \rightarrow \mathbb{R}^n$.

Une fonction $f : C \rightarrow \mathbb{R}$ est convexe si :

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad \forall x, y \in C, \quad \forall \lambda \in [0, 1].$$

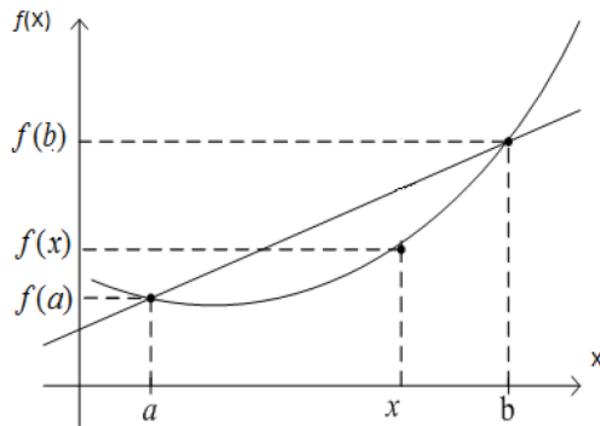


Figure 1.2- Fonction convexe

Remarque :

Une fonction f est dite *concave* si $-f$ est convexe.

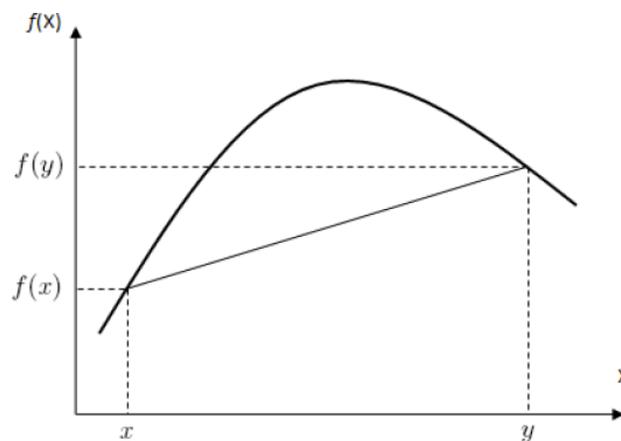


Figure 1.3- Fonction concave

Une fonction f est *strictement convexe* si l'inégalité ci-dessus est stricte pour tout $x, y \in C$ tel que $x \neq y$ et tout $\lambda \in [0, 1]$.

1.2 Définitions :

1.2.1 Un algorithme :

Un algorithme répond à un problème. Il est composé d'un ensemble d'étapes simples nécessaires à la résolution dont le nombre varie en fonction du nombre d'éléments à traiter. D'autre part, plusieurs algorithmes peuvent répondre à un même problème.

Pour savoir quelle méthode est plus efficace il faut les comparer. Pour cela, on utilise une mesure que l'on appelle la "*complexité*" qui représente le nombre d'étapes qui seront nécessaires pour résoudre le problème pour une entrée de taille donnée.

1.2.2 Formelle d'un algorithme :

La *formelle d'un algorithme* est une application $A : X \subseteq \mathbb{R}^n \rightarrow Y \subseteq P(X)$. Le vecteur x^{k+1} est donc généré par l'algorithme de la manière suivante en fonction de x^k , A est appliquée à x^k et l'on choisit $x^{k+1} \in A(x^k) \subseteq P(X)$.

1.2.3 La notion d'algorithme itératif :

Considérons le problème consistant à minimiser $f(x)$ sous la contrainte $x \in X$.

Un *algorithme itératif* permettant de résoudre ce problème est un *processus itératif* générant une suite de vecteurs x^0, x^1, \dots, x^n de X en fonction d'une séquence d'instruction et d'une condition d'arrêt.

La production d'un vecteur x^{k+1} à partir d'un vecteur x^k (qui tous les deux appartiennent à X) constitue une *itération* de l'algorithme.

Un tel algorithme est dit *de descente*, si le coût du vecteur généré à l'itération $k+1$ est strictement inférieur au coût du vecteur généré à l'itération k , c'est-à-dire :

$$f(x^{k+1}) < f(x^k), \forall k \geq 0.$$

1.2.4 La complexité :

La théorie de la complexité vise à savoir si la réponse à un problème peut être donnée très efficacement, efficacement ou au contraire être inatteignable en pratique (et en théorie), avec des niveaux intermédiaires de difficulté entre les deux extrêmes; pour cela, elle se fonde sur une estimation -théorique- des temps de calcul et des besoins en mémoire informatique.

Dans le but de mieux comprendre comment les problèmes se placent les uns par rapport aux autres, la théorie de la complexité établit des hiérarchies de difficultés entre les problèmes algorithmiques, dont les niveaux sont appelés des "*classes de complexité*". Ces hiérarchies comportent des ramifications, suivant que l'on considère des calculs déterministes -l'état suivant du calcul est déterminé par l'état courant- ou non déterministes.

1.2.5 La classe NP-difficile :

Un problème est *NP-difficile* si tout problème s'y réduit en temps polynomial.

1.2.6 Point limite :

On appelle *point limite* d'une séquence $\{x^k\}$ de points de \mathbb{R}^n tout point $x \in \mathbb{R}^n$ tel qu'il existe une sous-séquence de $\{x^k\}$ qui converge vers x .

1.2.7 Direction de descente :

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuellement différentiable et $x^* \in \mathbb{R}^n$. S'il existe un vecteur d tel que la dérivée directionnelle de f dans la direction d au point x^* (notée $f'(x^*, d)$) est strictement inférieure à zéro, alors d est *une direction de descente* de f en x^* .

En outre, selon, une propriété connue de l'analyse :

$$f'(x, d) = \nabla f(x)d.$$

La définition peut être également énoncée sous la forme équivalente suivante : s'il existe un vecteur d tel que : $\nabla f(x^*)d < 0$, alors d est *une direction de descente* de f en x^* .

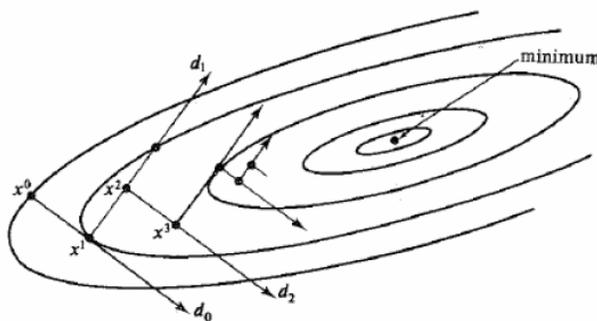


Figure 1.4- Direction de descente (Gradient à pas optimal)

1.2.8 Point stationnaire :

Soit $f : \mathbb{R}^n \mapsto \mathbb{R}$ une fonction différentiable. Tout vecteur $x \in \mathbb{R}^n$ qui vérifie : $\nabla f(x) = 0$ est appelé *un point stationnaire* ou *point critique* de f .

1.2.9 Minima locaux et globaux :

Soient l'ensemble $S \subseteq \mathbb{R}^n$ et une fonction $f : S \mapsto \mathbb{R}$. Les minima locaux et globaux de f sur S sont définis de la manière suivante :

a. Minimum local :

Intuitivement, un vecteur $x^* \in S$ est un *minimum local* de f sur S s'il a un coût plus faible que celui de ses voisins. Formellement, x^* est un minimum local de f sur S si $\exists \epsilon > 0$ tel que :

$$f(x^*) \leq f(x) \quad \forall x \in S \text{ avec } |x - x^*| < \epsilon$$

où $|v|$ désigne la norme du vecteur v .

Le minimum local est *strict* si :

$$f(x^*) < f(x) \quad \forall x \in S \text{ avec } |x - x^*| < \epsilon.$$

b. Minimum global :

Un vecteur $x^* \in S$ est un minimum *global* de f sur S s'il a un coût plus faible que celui de tous les autres vecteurs dans S . Formellement, x^* est un minimum global de f sur S si :

$$f(x^*) \leq f(x) \quad \forall x \in S.$$

Le minimum global est *strict* si

$$f(x^*) < f(x) \quad \forall x \in S.$$

Les maxima locaux et globaux sont définis de manière similaire. Notons que x^* est un maximum local (respectivement global) de la fonction f sur l'ensemble S si x^* est un minimum local (respectivement global) de $-f$ sur S . Il découle de cette observation que tout problème de maximisation peut être réduit immédiatement à un problème de minimisation (et inversement) en multipliant la fonction objectif par -1 .

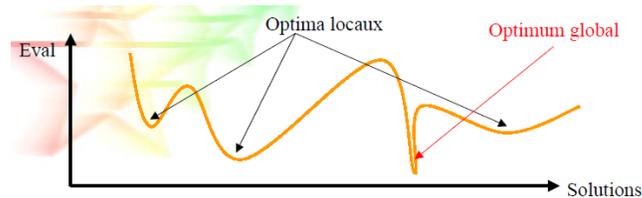


Figure 1.5- Minima locaux et globaux

Remarque :

Dans le cas d'une fonction objectif convexe, il n'y a pas de distinction entre minimum local et global: tout minimum local est également global, comme l'établit le théorème suivant :

Théorème 1.2.1^[12] :

Soit $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ une fonction convexe définie sur l'ensemble convexe X . Alors, tout minimum local de f sur X est également un minimum global. Si f est strictement convexe, alors il existe au plus un minimum global de f .

1.3 Conditions d'optimalité :

1.3.1 Pourquoi avons-nous besoin de conditions d'optimalité?

Afin d'analyser ou de résoudre de manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer de conditions d'optimalité. En effet, celles-ci nous servent non seulement à vérifier la validité des solutions obtenues, mais souvent l'étude de ces

conditions aboutit au développement des algorithmes de résolution eux-mêmes.

Des conditions équivalentes peuvent être obtenues de diverses manières, en procédant à des analyses suivantes différentes "lignes directrices". L'approche considérée ici pour l'obtention de conditions est basée sur les notions de descente et de direction admissible.

1.3.2 Conditions nécessaires :

Etant donné un vecteur x^* , nous souhaiterions être capables de déterminer si ce vecteur est un minimum local ou global de la fonction f . La propriété de différentiabilité continue de f fournit une première manière de caractériser une solution optimale.

Théorème 1.3.1^[1] : (condition nécessaire du premier ordre)

Soient $f : \mathbb{R}^n \mapsto \mathbb{R}$ continuellement différentiable et $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de f , alors $\nabla f(x^*) = 0$.

Ce théorème établit une première condition nécessaire pour que x^* soit un minimum (local ou global) de la fonction f , qui est $\nabla f(x^*) = 0$. Elle fait intervenir le vecteur gradient de f , dont les composantes sont ses premières dérivées partielles; c'est la raison pour laquelle cette condition est appelée *condition nécessaire du premier ordre*.

Remarque :

Si f est convexe, la condition nécessaire du premier ordre est également suffisante pour que x^* soit un minimum global.

Dans le cas où f est deux fois continuellement différentiable, une autre condition nécessaire est donnée par le théorème 1.3.2. Elle est appelée *condition nécessaire du second ordre* car elle fait intervenir la matrice hessienne de f (que nous noterons par $\nabla^2 f$), dont les éléments sont ses secondes dérivées partielles.

Théorème 1.3.2^[1] : (conditions nécessaires du second ordre)

Soient $f : \mathbb{R}^n \mapsto \mathbb{R}$ deux fois continuellement différentiable et $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de f , alors $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est semi-définie positive.

Preuve :

Soit x^* un minimum local de f .

Comme f est deux fois différentiable, le développement de Taylor au voisinage de x^* donne :

$$f(x) = f(x^*) + \nabla f^T(x^*)(x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*) + \|x - x^*\|^2 \theta(x - x^*)$$

avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$.

1. Pour La condition nécessaire du premier ordre :

Si $\nabla f(x^*) \neq 0$ alors en choisissant $x = x^* - \theta \nabla f(x^*)$ on aurait, pour $\theta > 0$ suffisamment petit : $f(x) < f(x^*)$ ce qui contredirait le fait que x^* est un minimum local.

2. Pour la condition nécessaire du second ordre :

Si la matrice $\nabla^2 f(x^*)$ n'est pas semi-définie positive, c'est qu'il existe un vecteur $d \in \mathbb{R}^n (d \neq 0)$ tel que : $d^T \nabla^2 f(x^*) d < 0$.

En choisissant alors $x = x^* + \theta d$, pour $\theta > 0$ suffisamment petit, on aurait $f(x) < f(x^*)$ ce qui contredit encore l'optimalité locale de x^* .

1.3.3 Conditions suffisantes :

Les conditions données précédemment sont nécessaires (si f n'est pas convexe), c'est-à-dire qu'elle doivent être satisfaites pour tout minimum local; cependant, tout vecteur vérifiant ces conditions n'est pas nécessairement un minimum local. Le théorème 1.3.3 établit une condition suffisante pour qu'un vecteur soit un minimum local, si f est deux fois continuellement différentiable.

Théorème 1.3.3^[1] : (conditions suffisantes du second ordre)

Soient $f : \mathbb{R}^n \mapsto \mathbb{R}$ deux fois continuellement différentiable et $x^* \in \mathbb{R}^n$. Si $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est définie positive, alors x^* est un minimum local de f .

Preuve :

Considérons un point x^* satisfaisant les conditions du théorème.

Le développement de Taylor de f au voisinage de x^* s'écrit alors :

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*) (x - x^*) + \|x - x^*\|^2 \theta(x - x^*)$$

avec $\theta(x - x^*) \rightarrow 0$ quand $x \rightarrow x^*$.

Pour toute direction de déplacement $d \in \mathbb{R}^n (\|d\| = 1)$ on a :

$$f(x^* + td) = f(x^*) + \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

d'où

$$f(x^* + td) - f(x^*) = \frac{t^2}{2} d^T \nabla^2 f(x^*) d + t^2 \theta(t)$$

où $\theta(t) \rightarrow 0$ quand $t \rightarrow 0$.

Comme $\nabla^2 f(x^*)$ est définie positive alors on a :

$$\frac{t^2}{2} d^T \nabla^2 f(x^*) d > 0$$

et par suite, pour θ suffisamment petit, on a :

$$f(x^* + td) > f(x^*)$$

Ce qui montre que x^* est bien un minimum local de f .

Théorème 1.3.4^[16] : (conditions suffisantes d'optimalité globale)

Soit une fonction continue : $f : \mathbb{R}^n \mapsto \mathbb{R}$ et $x^* \in \mathbb{R}^n$ un minimum local de f .

- Si f est une fonction convexe, alors x^* est un minimum global de f .
- Si de plus f est strictement convexe, x^* est l'unique minimum global de f .

Autrement dit, dans le cas convexe, la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.

Preuve :

Supposons par l'absurde qu'il existe un autre minimum local $x^* \neq x^*$, tel que $f(x^*) < f(x^*)$.

Par la convexité de f , nous avons

$$f(\lambda x^* + (1 - \lambda)x^*) \leq \lambda f(x^*) + (1 - \lambda)f(x^*)$$

où $0 \leq \lambda \leq 1$.

Comme $f(x^*) < f(x^*)$, nous avons pour chaque $\lambda \in]0,1[$

$$f(\lambda x^* + (1 - \lambda)x^*) < \lambda f(x^*) + (1 - \lambda)f(x^*) = f(x^*) \quad (1.1)$$

Considérons $\epsilon > 0$ arbitraire et montrons que la définition d'un minimum local est contredite.

Si $\epsilon \geq \|x^* - x^*\|$, la définition d'un minimum local n'est pas vérifiée pour $x = x^*$, en prenant $\lambda = 1$ dans (1.1).

Si $\epsilon < \|x^* - x^*\|$, considérons $0 < \eta < 1$ tel que $\epsilon < \|\eta x^* - (1 - \eta)x^*\| = \epsilon$. Dans ce cas, la définition d'un minimum local n'est pas vérifiée pour $x = (\lambda x^* + (1 - \lambda)x^*)$ avec $\eta \leq \lambda < 1$ par (1.1). Comme $\eta < 1$, de tels λ existent toujours.

Considérons maintenant une fonction strictement convexe, et supposons que x^* et y^* soient deux minima globaux distincts, donc $x^* \neq y^*$ et $f(x^*) = f(y^*)$.

En fonction de la convexité stricte, nous avons :

$$f(\lambda x^* + (1 - \lambda)y^*) < \lambda f(x^*) + (1 - \lambda)f(y^*) = f(x^*) = f(y^*)$$

ce qui contredit que x^* et y^* sont des minima globaux.

1.3.4 Conditions d'optimalité pour les problèmes quadratiques

Parmi les fonctions non linéaires, les fonctions quadratiques joueront un rôle important dans les algorithmes d'optimisation.

Fonction quadratique :

Une fonction $f : \mathbb{R}^n \mapsto \mathbb{R}$ sera dite *quadratique* si elle peut s'écrire

$$f(x) = \frac{1}{2}x^T Qx + b^T x + c$$

où Q est une matrice symétrique d'ordre $n * n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$. Nous avons alors :

$$\nabla f(x) = Qx + c$$

et

$$\nabla^2 f(x) = Q$$

Théorème 1.3.5^[16] : Condition d'optimalité pour les problèmes quadratiques

Considérons le problème :

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2}x^T Qx + b^T x + c \quad (1.2)$$

où Q est une matrice symétrique d'ordre $n * n$, $b \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

Si Q est définie positive, alors

$$x^* = -Q^{-1}b \quad (1.3)$$

est l'unique minimum global de (1.2).

Preuve :

Nous avons $\nabla f(x) = Qx + b$ et $\nabla^2 f(x) = Q$

Comme Q est définie positive, le point x^* dans (1.3) est bien définie car :

$$\nabla f(x^*) = Qx^* + b$$

On remplace x^* par sa valeur, on aura

$$\nabla f(x^*) = -QQ^{-1}b + b = 0$$

La condition suffisante d'optimalité est vérifiée et x^* est un minimum local de f . De plus, si f est convexe alors d'après la condition suffisante d'optimalité globale, x^* est l'unique minimum global.

1.4 Le problème d'optimisation :

Il existe de nombreuses sortes de problèmes d'optimisation. Certaines caractéristiques permettent de les distinguer : Comportent-ils des contraintes? Les fonctions en jeu sont-elles linéaires? Sont-elles quadratiques? Sont-elles convexes? Les domaines de définitions des fonctions sont-ils continus ou discrets?

Tous ces problèmes possèdent des structures différentes et ne peuvent être traités de la même façon.

Le présent sujet a pour vocation de se focaliser sur les problèmes *d'optimisation quadratiques non convexes définis sur un box*.

Ceux-ci sont généralement présentés sous la forme suivante :

$$\min f(x) = \begin{cases} \frac{1}{2}x^T Qx + c^T x \\ \text{avec } : l_i \leq x_i \leq u_i; \quad i = 1, \dots, n. \end{cases} \quad (1.4)$$

$$\text{où : } \begin{cases} Q = \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{pmatrix} : \text{matrice d'ordre } n, \text{ symétrique indéfinie ou semi-définie négative.} \\ l = (l_1, \dots, l_n)^T \quad \text{et} \quad u = (u_1, \dots, u_n)^T : \text{des composantes finies.} \\ B = \{x/l \leq x \leq u\} \text{ (C'est un box).} \end{cases}$$

C'est un problème NP-difficile. (Murty and Kabadi, 1987)^[19].

Conclusion

Nous avons vu que les conditions nécessaires d'optimalité permettent d'identifier les points critiques, qui sont les candidats pour la résolution d'un problème d'optimisation. Une manière d'aborder le problème serait de résoudre le système d'équations défini par ces conditions.

Nous allons voir maintenant les différentes méthodes de pénalité intérieure.

Chapitre 2

Méthodes de pénalité intérieure

↳ Cette théorie et ces phénomènes nous montrent comment on peut amener autrui à modifier ses comportements sans recourir à l'autorité, ni même à quelque stratégie persuasive, mais par des moyens détournés ↳
(J.L.BEAUVOIS et R.V.JOULE (1987).Petit traité de manipulation à l'usage des honnêtes gens).

Introduction

L'impact considérable qu'ont eu les méthodes de points intérieurs sur la communauté scientifique de la programmation mathématique, dû à leurs bonnes propriétés théoriques et algorithmiques, se devait être compris. D'où l'intérêt de jeter un aperçu sur ces différentes méthodes et leurs rôles dans la théorie de la complexité.

2.1 Vue d'ensemble :

La pénalisation est un concept simple qui permet de transformer un problème d'optimisation avec des contraintes en un problème ou en une suite de problèmes d'optimisation sans contraintes. C'est un concept qui a une utilité à la fois théorique et numérique.

En analyse, l'approche par pénalisation est parfois utilisée pour étudier un problème d'optimisation dont les contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence. Si on a de la chance ou si la pénalisation est bien choisie, des passages à la limite parfois délicats permettent d'obtenir des propriétés du problème original (l'existence de solution par exemple). D'autre part, la pénalisation est un outil permettant d'étudier les problèmes d'optimisation avec et sans contraintes dans un même formalisme.

D'un point de vue numérique, cette transformation en problèmes sans contraintes permet d'utiliser des algorithmes d'optimisation sans contraintes pour obtenir la solution de problèmes dont l'ensemble admissible peut avoir une structure complexe.

Cela semble merveilleux, inespéré, de voir que l'on puisse ainsi utiliser des algorithmes qui ne cherchent qu'à minimiser une fonction pour trouver des points qui, en plus d'être optimaux, sont admissibles. Cette approche est de ce fait très souvent utilisée. Elle permet d'obtenir une solution de qualité suffisante rapidement sans avoir à entrer dans l'algorithme sophistiqué de l'optimisation avec des contraintes.

Ce n'est cependant pas une technique universelle, car elle a ses propres inconvénients : non-différentiabilité ou nécessité de minimiser une suite de fonctions de plus en plus mal conditionnées.

Les différentes techniques de pénalisation relèvent souvent du principe suivant. On remplace le problème

$$(P_X) \begin{cases} \min f(x) \\ x \in X, \end{cases}$$

où X est une partie d'un espace vectoriel E , par un ou des problème(s)

$$(P_\beta) \quad \min_{x \in E} e(x, \beta)$$

où $e(x, \beta)$ est obtenu en ajoutant à $f(x)$ le terme $\beta p(x)$:

$$e(x, \beta) = f(x) + \beta p(x)$$

Le but de ce terme additionnel est de pénaliser la violation des contraintes (on parle alors de *pénalisation extérieure*) ou l'abord de la frontière du domaine admissible (on parle dans ce cas de *pénalisation intérieure*). Le scalaire $\beta > 0$ est appelé le *facteur (paramètre) de pénalisation*. Les propriétés de $e(x, \beta)$ vont dépendre de sa grandeur. On peut alors résoudre (P_β) par une méthode d'optimisation sans contraintes.

La question qui se pose immédiatement est de savoir si en résolvant (P_β) on résout (P_X) . Autrement dit, on cherche à savoir quand les ensembles de solutions de (P_X) et (P_β) coïncident. Cela va dépendre du choix de la fonction p et de β .

Par exemple, on pourrait choisir p comme suit :

$$p(x) = \begin{cases} 0 & \text{si } x \in X \\ +\infty & \text{si } x \notin X. \end{cases}$$

Il est clair que dans ce cas, les problèmes (P_X) et (P_β) sont identiques : ils ont les mêmes ensembles de solutions. Cette fonction de pénalisation est parfois utilisée dans la théorie, car elle permet de traiter en même temps les problèmes avec ou sans contraintes.

Cependant, ce choix de p n'est pas très utile en pratique car les méthodes classiques d'optimisation ne peuvent pas être utilisées sur des fonctions qui prennent la valeur $+\infty$ dans des régions visitées par les itérés (dans quelle direction se déplacer pour faire décroître $e(x, \beta)$ si l'itéré courant n'est pas dans X ?).

Cette question conduit à la notion de pénalisation exacte.

Définition :

On dit qu'une fonction de pénalisation $e(x, \beta)$ associée au problème (P_X) est *exacte* si toute solution de (P_X) minimise $e(x, \beta)$ et qu'elle est *inexacte* dans le cas contraire (il y a des solutions de (P_X) qui ne minimisent pas $e(x, \beta)$).

Le terme *solution* est pris dans son sens générique et il faudra chaque fois préciser si l'on veut parler de point stationnaire, de minimum local ou de minimum global.

La structure du problème (P_β) , dont le critère est la somme pondérée de deux fonctions, permet d'énoncer d'emblée une propriété très générale sur le comportement de chaque terme *en une solution* lorsque le poids β varie. Intuitivement, si β augmente, on attache

plus d'importance à p et il semble normal que, si x_β^* est une solution, $p(x_\beta^*)$ décroisse.

La proposition suivante énonce cela de façon rigoureuse. Le résultat est très général puisqu'il ne fait aucune hypothèse de convexité ou de différentiabilité; seule la construction de (P_β) intervient.

Proposition^[29] :

On note x_β^* une solution de (P_β) . Alors, lorsque $\beta > 0$ croît, $p(x_\beta^*)$ décroît, $f(x_\beta^*)$ croît et, si de plus $p(\cdot) \geq 0$, $e(x_\beta^*, \beta)$ croît.

Démonstration :

Soient $0 < \beta_1 < \beta_2$. En exprimant que $x_{\beta_1}^*$ est solution de (P_{β_1}) , puis en intervertissant les indices β_1 et β_2 , on obtient :

$$\begin{aligned} f(x_{\beta_1}^*) + \beta_1 p(x_{\beta_1}^*) &\leq f(x_{\beta_2}^*) + \beta_1 p(x_{\beta_2}^*) \\ f(x_{\beta_2}^*) + \beta_2 p(x_{\beta_2}^*) &\leq f(x_{\beta_1}^*) + \beta_2 p(x_{\beta_1}^*) \end{aligned}$$

En sommant, on obtient $0 \leq (\beta_2 - \beta_1)(p(x_{\beta_1}^*) - p(x_{\beta_2}^*))$. Comme $\beta_2 > \beta_1$, on en déduit que $p(x_{\beta_1}^*) \geq p(x_{\beta_2}^*)$ et donc que $p(x_\beta^*)$ décroît.

En tenant compte de ce résultat, la première inégalité ci-dessus montre que $f(x_{\beta_1}^*) \leq f(x_{\beta_2}^*)$: $f(x_\beta^*)$ décroît. Enfin, si $p(\cdot) \geq 0$, on a $e(x, \beta_1) \leq e(x, \beta_2)$ pour tout x ; en particulier $\inf_x e(x, \beta_1) \leq \inf_x e(x, \beta_2)$.

Le même raisonnement montre que l'on a une croissance ou décroissance *stricte* des suites si x_β^* est *l'unique minimum* de (P_β) et si x_β^* change avec β .

2.2 Les méthodes de points intérieurs :

Les méthodes de points intérieurs ont été développées dans les années 60 dans le but de résoudre des programmes mathématiques non linéaires. Leur utilisation pour la programmation linéaire n'a pas reçu d'enthousiasme à cause de la dominance du simplexe à cette époque. Après l'apparition de l'algorithme de *Karmarkar*^[11] en 1984 pour la programmation linéaire, les méthodes de points intérieurs ont connu une véritable révolution, on enregistre plus de 3000 publications en quelques années.

On distingue trois classes fondamentales des méthodes de points intérieurs à savoir :

-Les méthodes affines.

-Les méthodes de réduction du potentiel.

-Et les méthodes de trajectoire centrale (Chemin central).

2.2.1 Méthodes affines (*Dikin 1967*):

Il s'agit pratiquement de l'algorithme de *Karmarkar*^[11] sans fonction potentiel et sans transformation projective, on utilise une transformation affine et on remplace la contrainte de non négativité par un ellipsoïde qui contient le nouveau itéré.

L'algorithme est d'une structure simple, malheureusement, il n'est pas facile de démontrer la polynomialité.

2.2.2 Méthodes de réduction du potentiel :

La fonction potentiel joue un grand rôle dans le développement des méthodes de points intérieurs. L'algorithme de *Karmarkar*^[11] appliqué au programme linéaire sous forme standard utilise une fonction potentiel de la forme:

$$\rho \log(cx - Z) - \sum_{i=1}^n \log(x_i)$$

où $\rho = n + 1$ et Z est une borne inférieure de la valeur optimale de l'objectif. *Karmarkar* prouve la convergence et la polynomialité de son algorithme par monter que cette fonction est réduite à chaque itération par au moins une constante. Depuis 1987, les chercheurs introduisent des fonctions du potentiel de type primales-duales, parmi lesquelles, celle de *Tanabe*, *Todd* et *Ye* définie par:

$$\phi\rho(x,s) = \rho \log(x^T s) - \sum_{i=1}^n \log(x_i s_i) \quad \text{pour } \rho > n.$$

Cette fonction a joué un rôle très important dans le développement des algorithmes de réduction du potentiel après 1988. Les algorithmes correspondants à ces méthodes possèdent une complexité polynomiale, ils nécessitent $o(\sqrt{n} |\log \epsilon|)$ itérations pour réduire le saut de dualité ($x^T s \leq \epsilon$, ϵ est une précision donnée).

2.2.3 Méthodes de trajectoire centrale (TC) :

Elles ont été à la même époque que les méthodes de réduction du potentiel et pleinement développées au début des années 90. Elles possèdent de bonnes propriétés théoriques: une complexité polynomiale et une convergence superlinéaire.

Les algorithmes de (TC) restreints les itérés à un voisinage du chemin central, ce dernier est un arc de points strictement réalisables.

Plusieurs chercheurs essaient de généraliser le principe de ces méthodes pour la programmation non linéaire. En 1987, Megiddo a proposé un algorithme primal-dual de trajectoire centrale pour la programmation linéaire avec une généralisation pour le problème de complémentarité linéaire (PCL). Kojima et al ont développé un algorithme primal-dual pour la programmation linéaire, une extension pour le (PCL) est proposée par les mêmes chercheurs en 1989 avec la complexité $o(\sqrt{n} \log(\frac{1}{\epsilon}))$ itérations.

2.3 Méthode de Karmarkar^[11] :

En 1984, Karmarkar a proposé un algorithme prometteur pour la programmation linéaire, il possède des propriétés théoriques attractives et un bon comportement numérique.

L'algorithme est conçu pour résoudre un programme linéaire de la forme :

$$(PLS) \begin{cases} \min c^t x = z^* \\ Ax = 0 \\ x \in S_n \end{cases} \quad (2.1)$$

pour lequel on connaît à priori la valeur optimale $z^* = 0$ et une solution réalisable par exemple le point $a = \frac{1}{n}e_n$ (centre du simplexe S_n), e_n désigne le vecteur dont toutes les composantes sont égales à 1.

A est une matrice de type (m,n) et de plein rang $rgA = m \leq n$ (c'est une hypothèse classique dans la programmation linéaire). $S_n = \{x \in \mathbb{R}^n : x \geq 0, e_n^t x = 1\}$ est le simplexe de dimension $(n - 1)$.

2.3.1 Description de l'algorithme de base :

Partant de la solution initiale $x^0 = a$, l'algorithme construit une suite de points intérieurs qui converge vers une solution optimale du problème en un temps polynômial.

Dans le but de ramener facilement l'objectif à zéro, on le minimise localement sur une sphère inscrite dans la région réalisable.

Donc, à chaque itération k , l'itéré $x^k > 0$ est ramené au centre de S_n par la transformation projective T_k définie par :

$$T_k : x \in S_n \rightarrow T_k(x) = y \in S_n$$

avec :

$$T_k(x) = \frac{D_k^{-1}x}{e_n^t D_k^{-1}x} = y, \quad T_k^{-1}(y) = \frac{D_k y}{e_n^t D_k y}$$

où $D_k = \text{diag}\{x^k\}$.

La transformation T_k applique le simplexe S_n dans lui même, en même temps l'itéré $x^k > 0$ est envoyé au centre de S_n , le transformé du programme linéaire (PLS) est le programme fractionnaire suivant :

$$\left\{ \begin{array}{l} \min \frac{c^t D_k y}{e_n^t D_k y} \\ \frac{A D_k y}{e_n^t D_k y} = 0 \\ e_n^t y = 1 \\ y \geq 0 \end{array} \right. \quad (2.2)$$

Les hypothèses de départ permettent d'obtenir le programme linéaire suivant :

$$\left\{ \begin{array}{l} \min c^t D_k y \\ A D_k y = 0 \\ e_n^t y = 1 \\ y \geq 0 \end{array} \right. \quad (2.3)$$

qui est bien de la forme (PLS).

On a besoin du lemme suivant pour simplifier la résolution du problème (2.3) ci dessus.

Lemme 2.3.1^[11] :

Si pour un programme linéaire donné on connaît une solution réalisable y^0 tel que ($y_i^0 > 0$, $i = 1 : n + 1$), alors l'ellipsoïde :

$$E = \left\{ y \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} \frac{(y_i - y_i^0)^2}{(y_i^0)^2} \leq \beta^2, \quad 0 < \beta < 1 \right\}$$

est dans l'intérieur de l'orthant positif de \mathbb{R}^{n+1} .

Preuve :

Supposons le contraire, c'est-à-dire : il existe $j \in \{1, \dots, n+1\}$ tel que : $y_j \leq 0$ alors :

$$\sum_{i=1}^{n+1} \frac{(y_i - y_i^0)^2}{(y_i^0)^2} \geq \frac{(y_j - y_j^0)^2}{(y_j^0)^2} \geq 1 > \beta^2.$$

D'après le lemme 1, si on ajoute au problème (2.3) la contrainte :

$y \in \mathbb{R}^n / \|y - a\| \leq \alpha r$ où $0 < \alpha < 1$ et $r = \frac{1}{\sqrt{n(n-1)}}$, alors la contrainte de positivité $y \geq 0$ devient redondante. On obtient alors le problème :

$$\left\{ \begin{array}{l} \min c^t D_k y \\ AD_k y = 0 \\ e_n^t y = 1 \\ \|y - a\|^2 \leq (\alpha r)^2 \end{array} \right. \quad (2.4)$$

C'est la minimisation d'une fonction linéaire sur une sphère, où la solution optimale est triviale conformément au théorème suivant :

Théorème 2.3.1^[11] :

La solution optimale du problème (2.4) est donnée explicitement par : $y^k = a - \alpha r d^k$ où $d^k = \frac{p^k}{\|p^k\|}$ et $p^k = p_{B_k}(D_k c)$, $B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$

Preuve :

Il suffit d'écrire les conditions nécessaires et suffisantes d'optimalité de KKT.

A chaque itération, on revient à la variable initiale x en appliquant la transformation inverse T_k^{-1} et ainsi de suite jusqu'à ce que le test d'optimalité ($c^t x \leq \epsilon$) se réalise où $\epsilon > 0$ est une précision donnée.

2.3.2 Algorithme de base :**Début algorithme****Initialisation :**

$\epsilon > 0$ est une précision donnée, $x^0 = a = \frac{1}{n} e_n$, $k = 0$

Pas 1**Tant que :** $c^t x^k > \epsilon$ faire :

- construire $D_k = \text{diag}\{x^k\}$, $A_k = AD_k$, $B_k = \begin{bmatrix} A_k \\ e_n^t \end{bmatrix}$
- calculer $p^k = (I - B_k^t(B_k B_k^t)^{-1} B_k) D_k c$, $d^k = \frac{p^k}{\|p^k\|}$
- calculer $y^k = a - \alpha r d^k$, $r = \frac{1}{\sqrt{n(n-1)}}$, $0 < \alpha < 1$

Pas 2 :

- prendre $x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k}{e_n^t D_k y^k}$, $k = k + 1$ et retourner au pas 1.

Fin tant que**Fin algorithme.****2.3.3 Convergence de l'algorithme :**

La fonction objectif du problème transformé se réduit d'un montant de $(1 - \frac{\alpha}{n-1})$ conformément au théorème suivant :

Théorème 2.3.2^[11] :Le point y^k vérifie :

$$\frac{c^t D_k y^k}{c^t D_k a} \leq 1 - \frac{\alpha}{n-1}.$$

Pour établir la convergence de cet algorithme, *Karmarkar* introduit à l'objectif la fonction dite *de potentiel* :

$$f(x) = \sum_{i=1}^n \ln\left(\frac{c^t x}{x_i}\right) \text{ définie sur } \{x \in \mathbb{R}^n : x > 0, Ax = 0, e_n^t x = 1\}.$$

Le lemme suivant montre que la réduction de $f(x)$ conduit directement à celle de $c^t x$.

Lemme 2.3.2^[11] :Soit x^k le $k^{\text{ième}}$ itéré de l'algorithme, alors :

$$\frac{c^t x^k}{c^t x^0} \leq (\exp[f(x^k) - f(x^0)])^{\frac{1}{n}}.$$

Donc, si la suite $(f(x^k))$ tend vers $-\infty$ alors la suite $(c^t x^k)$ tend vers zéro.

Dans le théorème suivant, *Karmarkar* montre que la convergence de son algorithme est réalisée en $O(nq + n \ln n)$ itérations pour $0 < \alpha < \frac{1}{4}$.

Théorème 2.3.3^[11] :

Si $0 < \alpha < \frac{1}{4}$, alors en partant de $x^0 = \frac{1}{n}e_n$, l'algorithme trouve après $O(nq + n \ln n)$ itérations un point réalisable x tel que :

$$\begin{aligned} &1/c^t x = 0 \text{ ou} \\ &2/\frac{c^t x}{c^t x^0} \leq \epsilon = 2^{-q} \text{ où } q \text{ est une précision fixée.} \end{aligned}$$

2.4 Généralisation de l'algorithme de base :

Soit le programme linéaire sous la forme standard :

$$(P) \begin{cases} \min c^t x = z^* \\ Ax = b \\ x \geq 0 \end{cases} \quad (2.5)$$

A est une matrice de type (m, n) , $c \in \mathbb{R}^n$ est le vecteur coût et $b \in \mathbb{R}^m$.

On définit le simplexe S_{n+1} par :

$$S_{n+1} = \{x \in \mathbb{R}^{n+1} : x \geq 0, e_{n+1}^t = 1\}$$

On suppose que :

(H1) La matrice A est de plein rang ($rgA = m$).

(H2) On dispose d'un point x^0 strictement réalisable ($Ax^0 = b, x^0 > 0$).

(H3) La valeur optimale z^* de l'objectif est connue au départ.

L'hypothèse 1 est classique.

L'hypothèse 2 n'est pas du tout restrictive, car on peut toujours obtenir une solution strictement réalisable ou prouver que le problème n'est pas réalisable moyennant une méthode simple de variables artificielles.

Pour l'hypothèse 3, si la valeur optimale est non nulle, l'égalité $e_{n+1}^t x = 1$ permet de se ramener à un objectif nul.

En effet, soit x^* une solution du problème et z^* la valeur optimale de l'objectif. Alors :

$$c^t x^* = z^* = z^* e_{n+1}^t x^* \Rightarrow (c - z^* e_{n+1})^t x^* = \hat{c}^t x^* = 0.$$

Si z^* n'est pas connue, il existe des techniques d'approximation fiables permettant de remplacer z^* par des bornes inférieures ou supérieures convenables.

Pour le système de contraintes: $Ax = b$, $b \neq 0$, on se ramène facilement à un système homogène, il suffit d'écrire:

$$Ax = be_{n+1}^t x \Rightarrow (A - be_{n+1}^t)x = 0.$$

La transformation projective notée T_k est une fonction:

$$T_k : \mathbb{R}_+^n \rightarrow S_{n+1} \text{ définie par: } T_k(x) = y$$

avec:

$$\begin{cases} y_i = \frac{x_i/x_i^k}{1 + \sum_{i=1}^n x_i/x_i^k}, & i = 1 : n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{cases} \quad (2.6)$$

On a:

$$y_i = \frac{x_i}{x_i^k} y_{n+1}, \quad i = 1 : n$$

ou encore $y[n] = (D_k^{-1}x)y_{n+1}$ où $y[n]$ désigne les n premières composantes de y et on a:

$$x = T_k^{-1}(y) = \frac{D_k y[n]}{y_{n+1}}, \quad D_k = \text{diag}(x^k).$$

On applique la transformation T_k au programme (2.5), on obtient:

$$\begin{cases} \min \frac{c^t D_k y[n]}{y_{n+1}} \\ A \frac{D_k y[n]}{y_{n+1}} = b \\ \sum_{i=1}^{n+1} y_i = 1 \\ y[n] \geq 0, \quad y_{n+1} > 0 \end{cases} \quad (2.7)$$

ou encore:

$$\begin{cases} \min \hat{c}^t y \\ \hat{A} y = 0 \\ e_{n+1}^t y = 1 \\ y \geq 0 \end{cases} \quad (2.8)$$

$$\text{où: } \begin{cases} \hat{c}_i = c_i x_i^k, & i = 1, \dots, n \\ \hat{c}_{n+1} = -z^* \end{cases}, \quad y = \begin{bmatrix} y[n] \\ y_{n+1} \end{bmatrix} \text{ et } \hat{A} = [AD_k \quad -b]$$

Notons que toute solution réalisable de (2.5) est transformée par T_k en une solution réalisable de (2.8) et réciproquement, toute solution réalisable y de (2.8) avec $y_{n+1} > 0$ est transformée par T_k^{-1} en une solution réalisable de x de (2.5).

Conclusion

On vient de voir l'intérêt considérable des méthodes de pénalité intérieure qu'elles portent sur les problèmes d'optimisation. D'où notre ambition de présenter dans ce qui va suivre une fonction barrière pour les problèmes quadratiques non convexes définis sur un box avec ses propriétés théoriques.

Chapitre 3

Méthode de la fonction barrière

Introduction

Cette méthode essaye de produire une solution de bonne qualité en suivant le paramètre de barrière comme chemin qui diminue.

3.1 Le principe :

Etant donné β la valeur du paramètre de barrière, la méthode cherche un point minimum de la fonction barrière dans une direction de descente.

3.2 Fonction barrière:

Pour approcher une solution du problème (1.4), nous présentons une limite barrière comme suit:

$$(x_i - l_i) \ln(x_i - l_i) + (u_i - x_i) \ln(u_i - x_i)$$

Ensuite on introduit le système de contraintes ($l_i \leq x_i \leq u_i$) à la fonction objectif et on obtient la fonction barrière suivante :

$$e(x, \beta) = f(x) + \beta \sum_{i=1}^n (x_i - l_i) \ln(x_i - l_i) + (u_i - x_i) \ln(u_i - x_i)$$

où β : est le paramètre de barrière qui change d'un nombre positif vers zéro.

La valeur initiale de β est suffisamment grande de telle sorte que $e(x, \beta)$ est strictement convexe.

On observe bien que la limite de la barrière est bien définie quand : $x_i = l_i$ et $x_i = u_i$, c'est-à-dire :

$$\lim_{x_i \rightarrow l_i^+} (x_i - l_i) \ln(x_i - l_i) = \lim_{x_i \rightarrow u_i^-} (u_i - x_i) \ln(u_i - x_i) = 0.$$

Quand " $l_i = 0$ " et " $u_i = 1$ ", la limite de la barrière apparait implicitement comme limite de fonction d'énergie définie par *Hopfield (1984)*^[10].

Exemple :

Soit le problème suivant :

$$\begin{cases} \min f(x) = \frac{1}{2}(x_1, x_2) \begin{pmatrix} -83.75 & 28.34 \\ 28.34 & -48.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (17.72, 15.22) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \text{S.C: } \begin{cases} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{cases} \end{cases}$$

La figure (3.1) suivante montre sa surface :

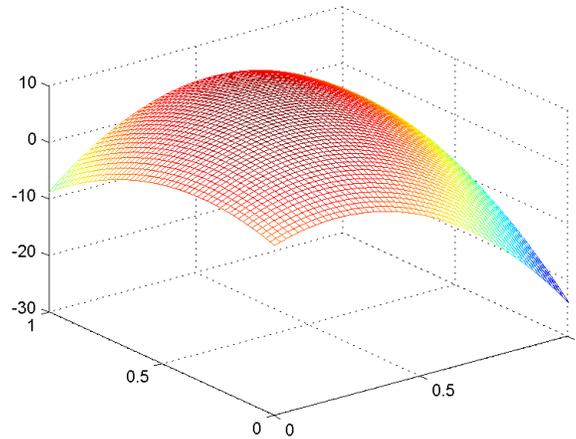


Figure 3.1- La surface de $f(x)$

En utilisant la fonction barrière définie au-dessus, on obtient :

$$e(x, \beta) = \frac{1}{2}(x_1, x_2) \begin{pmatrix} -83.75 & 28.34 \\ 28.34 & -48.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (17.72, 15.22) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ + \beta(x_1 \ln x_1 + (1 - x_1) \ln(1 - x_1) + x_2 \ln x_2 + (1 - x_2) \ln(1 - x_2))$$

$$\text{avec : } \begin{cases} 0 < x_1 < 1 \\ 0 < x_2 < 1 \end{cases} .$$

La figure (3.2) montre la nouvelle surface du problème en intégrant la fonction barrière avec $\beta = 50$:

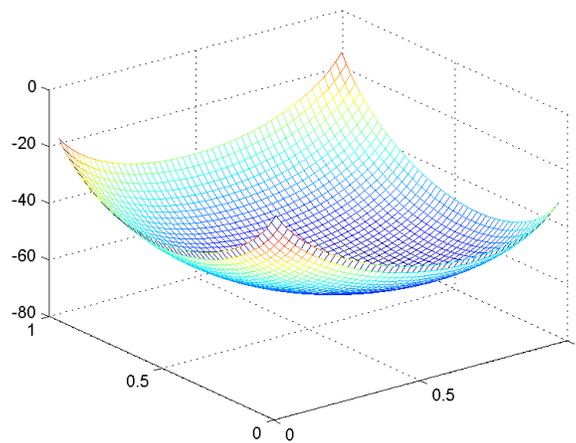


Figure 3.2- La surface de $e(x,50)$

De cette figure, on remarque qu'on a quatre points d'extrémum possibles qui sont : $(1,1)$, $(1,0)$, $(0,1)$ et $(0,0)$.

En faisant diminuer le paramètre de barrière, on aura :

$\beta = 25$: La figure (3.3)

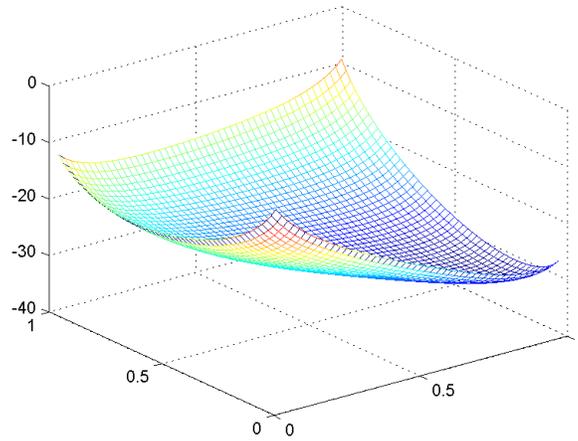


Figure 3.3- La surface de $e(x,25)$

$\beta = 15$: La figure (3.4)

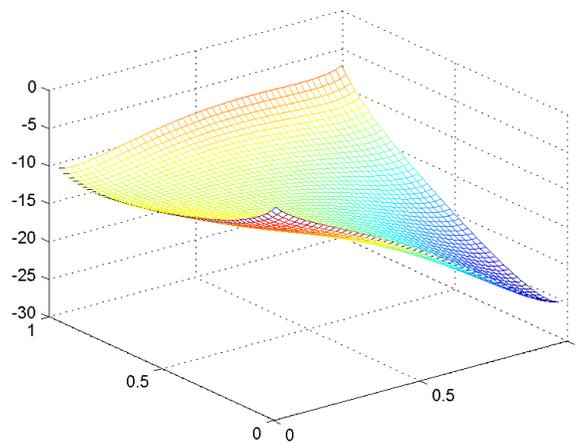


Figure 3.4- La surface de $e(x,15)$

$\beta = 5$: La figure (3.5)

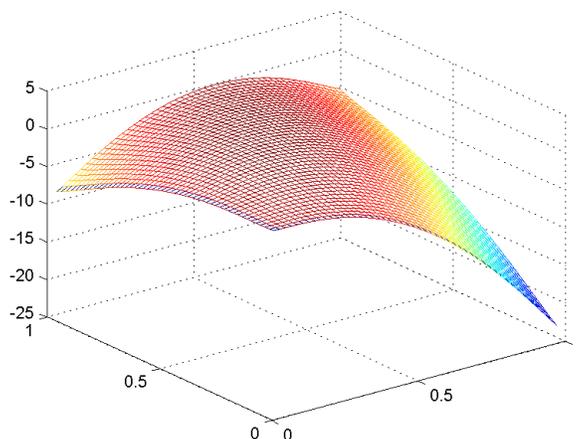


Figure 3.5- La surface de $e(x,5)$

De cette figure, on remarque bien que le point $(1,0)$ est un minimum global.

Au lieu de travailler sur le problème de départ, on considère l'arrangement suivant qui donne la solution du problème posé :

$$\min_{x \in B} e(x, \beta).$$

A la limite, c'est-à-dire quand " $\beta \rightarrow 0$ ", on obtient:

$$\frac{\partial e(x, \beta)}{\partial x_i} = \frac{\partial f(x)}{\partial x_i} + \beta \ln \frac{x_i - l_i}{u_i - x_i}$$

On observe que :

$$\lim_{x_i \rightarrow l_i^+} \frac{\partial e(x, \beta)}{\partial x_i} = -\infty$$

et

$$\lim_{x_i \rightarrow l_i^-} \frac{\partial e(x, \beta)}{\partial x_i} = +\infty$$

La matrice hessienne de $e(x, \beta)$ au point x , avec $l < x < u$, est donnée par :

$$\nabla^2 e(x, \beta) = Q + \beta((X - L)^{-1} + (U - X)^{-1})$$

où:

X: La matrice diagonale constituée par les composantes de x .

L: La matrice diagonale constituée par les composantes de l .

U: La matrice diagonale constituée par les composantes de u .

Quand β est suffisamment grand, c'est clair que $\nabla^2 e(x, \beta)$ est définie positive pour tout x , tel que: $l < x < u$.

Alors :

$e(x, \beta)$ est strictement convexe pour $l \leq x \leq u$ quand β est suffisamment grand.

Puisque: $\nabla f(x) = Qx + c$ est continu sur B , alors on peut dériver les résultats suivants :

Lemme 3.2.1^[3] :

Pour tout $\beta > 0$ donné, si x^* est un point minimum de $\min_{l \leq x \leq u} e(x, \beta)$ alors :

$$l < x^* < u.$$

Preuve :

Supposons que certaines composantes de x^* (Par exemple: x_i^*) sont égales à l_i .

Soit $\epsilon > 0$ proche de zéro "0".

On définit $y^* = (y_1^*, \dots, y_n^*)$ par :

$$y_j^* = \begin{cases} x_j^* & \text{si } j \neq i, \\ x_i^* + \epsilon & \text{si } j = i, \end{cases} \quad j = 1, \dots, n.$$

Donc, lorsque ϵ est suffisamment petit alors :

$$\frac{\partial e(y^*, \beta)}{\partial x_i} = \frac{\partial f(y^*)}{\partial x_i} + \beta \ln \frac{x_i^* + \epsilon - l_i}{u_i - x_i^* + \epsilon} = \frac{\partial f(y^*)}{\partial x_i} + \beta \ln \frac{\epsilon}{u_i - l_i - \epsilon} < 0.$$

Ce qui implique que $\frac{\partial f(y^*)}{\partial x_i}$ est borné.

Puis, en ajoutant à la $i^{\text{ème}}$ composante de y^* un petit nombre positif, on obtien un point de B près de y^* tel que: $e(x, \beta)$ est plus petit que $e(y^*, \beta)$.

Puisque $e(x, \beta)$ est continue sur B , $e(y^*, \beta)$ est près de $e(x^*, \beta)$ si ϵ est près de zéro "0".

Par conséquent, il existe un point de B arbitrairement près de x^* auquel $e(x, \beta)$ est plus petit que $e(x^*, \beta)$.

Ceci contredit le fait que x^* est un point minimum, ce qui implique qu'aucune composante de x^* n'égale à la limite inférieure.

De même, on peut prouver qu'aucune composante de x^* n'égale à la limite supérieure.

D'où la preuve du lemme.

Remarque :

Ce lemme indique que si x^* est un point minimum de:

$$\min_{l \leq x \leq u} e(x, \beta)$$

alors:

$$\nabla_x e(x^*, \beta) = 0$$

$$\text{avec : } \nabla_x e(x, \beta) = \left(\frac{\partial e(x, \beta)}{\partial x_1}, \dots, \frac{\partial e(x, \beta)}{\partial x_n} \right)^T.$$

Soient β_k , $k = 1, 2, \dots$, des nombres positifs tels que: $\beta_1 > \beta_2 > \dots$ et $\lim_{k \rightarrow \infty} \beta_k = 0$.

Soit x^* un minimum global du problème (1.4) et $x(\beta_k) = \arg \min \{ e(x, \beta_k) / x \in B \}$, $k = 1, 2, \dots$ alors on a le théorème suivant :

Théorème 3.2.1^[3] :

Pour $k = 1, 2, \dots$, on a :

$$f(x(\beta_k)) \geq f(x(\beta_{k+1})) \text{ et } \lim_{k \rightarrow \infty} f(x(\beta_k)) = f(x^*).$$

Preuve :

Soit :

$$p(x) = \sum_{i=1}^n ((x_i - l_i) \ln(x_i - l_i) + (u_i - x_i) \ln(u_i - x_i)).$$

Pour tout $x \in B$, on a :

$$\sum_{i=1}^n (u_i - l_i) \ln \frac{u_i - l_i}{2} \leq p(x) \leq \sum_{i=1}^n (u_i - l_i) \ln(u_i - l_i).$$

Poser :

$$b(x) = p(x) - \sum_{i=1}^n (u_i - l_i) \ln \frac{u_i - l_i}{2}.$$

tel que : $b(x) \geq 0$ pour tout $x \in B$.

Poser :

$$\psi(x, \beta) = f(x) + \beta b(x).$$

Alors :

$$e(x, \beta) = \psi(x, \beta) + \beta \sum_{i=1}^n (u_i - l_i) \ln \frac{u_i - l_i}{2}.$$

D'où :

$$x(\beta_k) = \arg \min \{ \psi(x, \beta_k) / x \in B \}.$$

Par les conditions de $x(\beta_k)$ et $x(\beta_{k+1})$, nous avons :

$$f(x(\beta_k) + \beta_k b(x(\beta_k))) \leq f(x(\beta_{k+1}) + \beta_k b(x(\beta_{k+1})))$$

et

$$f(x(\beta_{k+1}) + \beta_{k+1} b(x(\beta_{k+1}))) \leq f(x(\beta_k) + \beta_{k+1} b(x(\beta_k))).$$

Alors :

$$(\beta_k - \beta_{k+1}) b(x(\beta_k)) \leq (\beta_k - \beta_{k+1}) b(x(\beta_{k+1})).$$

Donc :

$$b(x(\beta_k)) \leq b(x(\beta_{k+1})) \text{ tant que } \beta_k > \beta_{k+1}.$$

D'où : $f(x(\beta_k)) \geq f(x(\beta_{k+1}))$.

Pour tout k , on peut écrire :

$$f(x^*) \leq f(x(\beta_k)) \leq f(x(\beta_k) + \beta_k b(x(\beta_k))) = \psi(x(\beta_k), \beta_k). \quad (3.1)$$

Pour tout $\epsilon > 0$, il existe un $\bar{x} \in B$ tel que :

$$f(\bar{x}) \leq f(x^*) + \epsilon.$$

D'où pour tout k , on a :

$$f(x^*) + \epsilon + \beta_k b(\bar{x}) \geq f(\bar{x}) + \beta_k b(\bar{x}) \geq \psi(x(\beta_k), \beta_k).$$

En passant à la limite, on aura :

$$\lim_{k \rightarrow \infty} \psi(x(\beta_k), \beta_k) \leq f(x^*) + \epsilon.$$

Suivant (3.1), on obtient :

$$\lim_{k \rightarrow \infty} \psi(x(\beta_k), \beta_k) \geq f(x^*).$$

Donc :

$$\lim_{k \rightarrow \infty} \psi(x(\beta_k), \beta_k) = f(x^*).$$

Et comme :

$$\lim_{k \rightarrow \infty} \beta_k b(x(\beta_k)) = 0.$$

Alors :

$$\lim_{k \rightarrow \infty} f(x(\beta_k)) = f(x^*).$$

D'où la preuve du théorème.

Le théorème suivant indique que chaque point limite de $x(\beta_k)$, $k = 1, 2, \dots$, est un point minimum global de (1.4).

Théorème 3.2.2^[3] :

Si tous les éléments de la diagonale de la matrice Q sont négatifs alors le problème (1.4) admet un minimum global.

Preuve :

Soit $x(\beta_{kj})$, $j = 1, 2, \dots$, un ordre convergent vers $x(\beta_k)$, $k = 1, 2, \dots$ avec :

$$\lim_{j \rightarrow \infty} x(\beta_{kj}) = v^*.$$

Du théorème 3.3.1, on obtient :

$$f(v^*) = f(x^*).$$

Dans ce qui suit nous prouvons que v^* est un point dans B .

Comme $x(\beta_{kj})$ est un point minimum de $\min_{x \in B} e(x, \beta_{kj})$, alors la matrice hessienne de $e(x, \beta_{kj})$ au point $x(\beta_{kj})$:

$$Q + \beta_{kj}((X(\beta_{kj}) - L)^{-1} + (U - X(\beta_{kj}))^{-1}) \text{ est semi-définie positive.}$$

D'où $X(\beta_{kj})$ est une matrice diagonale constituée par les composantes de $x(\beta_{kj})$.

Alors pour tout i avec $1 \leq i \leq n$, on a :

$$0 \leq (u^i)^T Q u^i + \beta_{kj} (u^i)^T ((X(\beta_{kj}) - L)^{-1} + (U - X(\beta_{kj}))^{-1}) u^i = q_{ii} + \beta_{kj} \left(\frac{1}{x_i(\beta_{kj}) - l_i} + \frac{1}{u_i - x_i(\beta_{kj})} \right) \quad (3.2)$$

où u^i est le $i^{\text{ème}}$ vecteur unité dans R^n .

Du fait que $q_{ii} < 0$ et (3.2), lorsque: $j \rightarrow \infty$, $x_i(\beta_{kj})$ s'approche de (tend vers) (l_i) ou (u_i) car β_{kj} tend vers zéro "0".

D'où, v^* est un point dans B .

D'où la preuve du théorème .

Théorème 3.2.3^[3] :

Une condition suffisante pour que x^k , $k = 1, 2, \dots$ soit un minimum local du problème (1.4) est :

1. A tout point limite de v , on a :

$$Qv + c \neq 0.$$

2. Tous les éléments de la diagonale de Q sont négatifs.

Preuve :

Pour tout x^k , $k = 1, 2, \dots$ continu dans B qui est fermé alors on peut extraire un ordre convergent.

Poser x^{kq} , $q = 1, 2, \dots$ cet ordre convergent vers x^k , $k = 1, 2, \dots$ avec : $\lim_{q \rightarrow \infty} x^{kq} = v$.

Soit X_{kq} la matrice diagonale constituée par les composantes de x^{kq} .

Puisque x^{kq} est un point minimum local de $\min_{l \leq x \leq u} e(x, \beta_{kq})$, alors : la matrice Hessienne de $e(x, \beta_{kq})$ au point x^{kq} est donnée par :

$$Q + \beta_{kq}((X_{kq} - L)^{-1}) + (U - X_{kq})^{-1}) \text{ qui est semi-définie positive .}$$

Alors pour tout i avec $1 \leq i \leq n$, on a :

$$0 \leq (u^i)^T Q u^i + \beta_{kq} (u^i)^T ((X_{kq} - L)^{-1} + (U - X_{kq})^{-1}) u^i = q_{ii} + \beta_{kq} \left(\frac{1}{x_i^{kq} - l_i} + \frac{1}{u_i - x_i^{kq}} \right) \quad (3.3)$$

où u^i est le $i^{\text{ème}}$ vecteur unité dans R^n .

Comme $q_{ii} < 0$ et (3.3) alors lorsque $q \rightarrow \infty$, x_i^{kq} tend vers (l_i ou u_i), car β_{kq} tend vers zéro.

D'où v est un point extrémum de B , car x^{kq} est un minimum local de $\min_{l \leq x \leq u} e(x, \beta_{kq})$.

A partir de l'état d'optimalité nécessaire de premier ordre, on obtient:

$$Qx^{kq} + c = -\beta_{kq} \left(\ln \frac{x_1^{kq} - l_1}{u_1 - x_1^{kq}}, \ln \frac{x_2^{kq} - l_2}{u_2 - x_2^{kq}}, \dots, \ln \frac{x_n^{kq} - l_n}{u_n - x_n^{kq}} \right)^T.$$

Alors:

$$\lim_{q \rightarrow \infty} -\beta_{kq} \left(\ln \frac{x_1^{kq} - l_1}{u_1 - x_1^{kq}}, \ln \frac{x_2^{kq} - l_2}{u_2 - x_2^{kq}}, \dots, \ln \frac{x_n^{kq} - l_n}{u_n - x_n^{kq}} \right)^T = Qv + c \neq 0. \quad (3.4)$$

Soit x un point intérieur dans B , alors:

$$(x - x^{kq})^T (Qx^{kq} + c) = - \sum_{i=1}^n \beta_{kq} (x_i - x_i^{kq}) \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}}.$$

Et comme v est un point extrémum dans B alors :

Si $v_i = l_i$, alors on a: $x_i - v_i > 0$ et $\lim_{q \rightarrow \infty} x_i^{kq} = l_i$.

Et quand q est suffisamment grand alors:

$$\beta_{kq} (x_i - v_i) \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} < 0$$

Si $v_i = u_i$, alors on a: $x_i - v_i < 0$ et $\lim_{q \rightarrow \infty} x_i^{kq} = u_i$.

Et quand q est suffisamment grand donc :

$$\beta_{kq} (x_i - v_i) \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} < 0.$$

De (3.4) on obtient:

$$\lim_{q \rightarrow \infty} \beta_{kq} \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} \neq 0, \quad i = 1, \dots, n.$$

Alors au moins un des $(x_i - v_i) \lim_{q \rightarrow \infty} \beta_{kq} \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} < 0$, $i = 1, \dots, n$, (C'est-à-dire qu'il existe au moins un terme négatif).

D'où:

$$\begin{aligned}
& (x - v)^T(Qv + c) = \lim_{q \rightarrow \infty} (x - x^{kq})^T(Qx^{kq} + c) \\
& = -\sum_{i=1}^n \lim_{q \rightarrow \infty} \beta_{kq} (x_i - x_i^{kq}) \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} \\
& = -\sum_{i=1}^n (x_i - v_i) \lim_{q \rightarrow \infty} \beta_{kq} \ln \frac{x_i^{kq} - l_i}{u_i - x_i^{kq}} \\
& > 0.
\end{aligned}$$

Donc, pour n'importe quel point intérieur x de B on a :

$$0 < (x - v)^T(Qv + c). \quad (3.5)$$

Alors, on aura:

$$f(x) - f(v) = \frac{1}{2}x^T Qx + c^T x - \frac{1}{2}v^T Qv - c^T v = (x - v)^T(Qv + c) + \frac{1}{2}(x - v)^T Q(x - v).$$

Et quand x est très proche de v , de (3.5) on obtient:

$f(x) - f(v) > 0$ car $\frac{1}{2}(x - v)^T Q(x - v)$ tend vers zéro "0" deux fois plus rapidement que $(x - v)^T(Qv + c)$.

Ce qui implique que v est un point minimum local de (1.4).

D'où la preuve du théorème.

Ce théorème signifie qu'au moins un point minimum local de (1.4) peut être obtenu si nous pouvons produire d'un point minimum local de la fonction de barrière pour une séquence de valeurs décroissantes du paramètre de barrière avec la limite zéro "0".

Conclusion

Nous venons de voir le comportement du problème perturbé par la fonction barrière qu'on a défini. Dans ce qui suivra nous présenterons la méthode de résolution qui donnera naissance à l'algorithme.

Chapitre 4

Algorithme et exemples d'application

*<< Le vrai danger, ce n'est pas quand les ordinateurs penseront
comme des hommes, c'est quand les hommes penseront
comme des ordinateurs >>*
Sydney Harris.

Introduction

Nous avons traité dans le chapitre précédent l'influence que porte la fonction barrière définie sur le problème d'optimisation posé.

Dans ce chapitre, nous nous intéressons à la méthode et à l'algorithme de résolution. Ensuite son application sur certains problèmes bien définis. Et enfin, son exécution sur des exemples numériques.

4.1 La méthode :

- Pour n'importe quelle donnée $\beta > 0$, considérant l'état d'optimalité nécessaire de premier ordre:

$$\frac{\partial e(x,\beta)}{\partial x_i} = 0 \quad , \quad i = 1, 2, \dots, n \quad (4.1)$$

A partir de (4.1) , on obtient :

$$x_i = \frac{u_i + l_i \exp\left(\frac{1}{\beta} \frac{\partial f(x)}{\partial x_i}\right)}{1 + \exp\left(\frac{1}{\beta} \frac{\partial f(x)}{\partial x_i}\right)} \quad , \quad i = 1, 2, \dots, n.$$

Poser:

$$d_i(x) = \frac{u_i + l_i \gamma_i(x)}{1 + \gamma_i(x)} \quad , \quad i = 1, \dots, n$$

et

$$d(x) = (d_1(x), \dots, d_n(x))^T,$$

où :

$$\gamma_i(x) = \exp\left(\frac{1}{\beta} \frac{\partial f(x)}{\partial x_i}\right).$$

Le lemme suivant prouve que pour n'importe quel $\beta > 0$ donné, si $l < x < u$ alors $(d(x) - x)$ est une direction de descente pour $e(x,\beta)$.

Lemme 4.1.1^[3] :

Supposons que $l < x < u$. Pour $i = 1, \dots, n$ on a :

- Quand $d_i(x) - x_i > 0$: $\frac{\partial e(x,\beta)}{\partial x_i} < 0$.
- Quand $d_i(x) - x_i < 0$: $\frac{\partial e(x,\beta)}{\partial x_i} > 0$.
- Et quand $d_i(x) - x_i = 0$: $\frac{\partial e(x,\beta)}{\partial x_i} = 0$.
- Quand $d(x) - x \neq 0$: $\nabla_x e(x,\beta)^T (d(x) - x) < 0$.

Preuve :

1. Soit $d_i(x) - x_i < 0$, alors :

$$\frac{u_i + l_i \gamma_i(x)}{1 + \gamma_i(x)} < x_i.$$

D'où:

$$1 < \gamma_i(x) \frac{x_i - l_i}{u_i - x_i}. \quad (4.2)$$

En passant au logarithme de (4.2), on aura:

$$0 < \frac{1}{\beta} \frac{\partial f(x)}{\partial x_i} + \ln \frac{x_i - l_i}{u_i - x_i}. \quad (4.3)$$

En multipliant (4.3) par $\beta > 0$ on aura:

$$0 < \frac{\partial f(x)}{\partial x_i} + \beta \ln \frac{x_i - l_i}{u_i - x_i} = \frac{\partial e(x, \beta)}{\partial x_i}.$$

D'où, quand $d_i(x) - x_i < 0$ on a bien:

$$\frac{\partial e(x, \beta)}{\partial x_i} > 0.$$

2. Soit $d_i(x) - x_i > 0$, alors:

$$\frac{u_i + l_i \gamma_i(x)}{1 + \gamma_i(x)} > x_i.$$

D'où :

$$1 > \gamma_i(x) \frac{x_i - l_i}{u_i - x_i}. \quad (4.4)$$

En passant au logarithme de (4.4) on aura:

$$0 > \frac{1}{\beta} \frac{\partial f(x)}{\partial x_i} + \ln \frac{x_i - l_i}{u_i - x_i}. \quad (4.5)$$

En multipliant (4.5) par $\beta > 0$ on aura:

$$0 > \frac{\partial f(x)}{\partial x_i} + \beta \ln \frac{x_i - l_i}{u_i - x_i} = \frac{\partial e(x, \beta)}{\partial x_i}.$$

D'où, quand $d_i(x) - x_i > 0$ on a bien:

$$\frac{\partial e(x, \beta)}{\partial x_i} < 0.$$

3. Soit $d_i(x) - x_i = 0$, alors:

$$\frac{u_i + l_i \gamma_i(x)}{1 + \gamma_i(x)} = x_i.$$

D'où:

$$1 = \gamma_i(x) \frac{x_i - l_i}{u_i - x_i}. \quad (4.6)$$

En passant au logarithme (4.6), on aura:

$$0 = \frac{1}{\beta} \frac{\partial f(x)}{\partial x_i} + \ln \frac{x_i - l_i}{u_i - x_i}.$$

En multipliant (4.7) par $\beta > 0$ on aura:

$$0 = \frac{\partial f(x)}{\partial x_i} + \beta \ln \frac{x_i - l_i}{u_i - x_i} = \frac{\partial e(x, \beta)}{\partial x_i}.$$

D'où, quand $d_i(x) - x_i = 0$ on a bien:

$$\frac{\partial e(x, \beta)}{\partial x_i} = 0.$$

Ce qui nous conduit à dire que:

$$\nabla_x e(x, \beta)^T (d(x) - x) = \sum_{i=1}^n \frac{\partial e(x, \beta)}{\partial x_i} (d_i(x) - x_i).$$

D'où la preuve du théorème.

Remarques :

1. Pour n'importe quelle valeur x tel que $l < x < u$ on a :

$$(d(x) - x = 0) \text{ si et seulement si } (\nabla_x e(x, \beta) = 0).$$

2. On remarque que $(d(x) - x)$ satisfait toujours et automatiquement les contraintes en cherchant un point dans $(d(x) - x)$ si la longueur du pas est un nombre entre zéro et un .

La méthode développée pour approcher la solution du problème (1.4) est basée sur la direction de descente $(d(x) - x)$.

4.2 L'idée :

L'idée de la méthode est la suivante:

- Poser $\beta_q, q = 1, 2, \dots$ un ordre donné des nombres positifs tels que :

$$\beta_1 > \beta_2 > \dots \text{ et } \lim_{q \rightarrow \infty} \beta_q = 0.$$

- Avec β_1 est suffisamment grand de telle sorte que $e(x, \beta_1)$ est strictement convexe au dessus de $l \leq x \leq u$.
- Soit x^0 un point intérieur dans B .
- Démarrant de $x^{q-1}, q = 1, 2, \dots$, cherchant $x^q \in B$ satisfaisant $d(x^q) - x^q = 0$ en utilisant la direction de descente $(d(x) - x)$.

4.3 Algorithme :

Etape 0 :

- Choisir ϵ une tolérance donnée.
- Choisir θ un nombre entre $(0, 1)$ proche de 1.
- Choisir x^0 un point satisfaisant $l < x^0 < u$.
- Choisir β un nombre positif satisfaisant $e(x, \beta)$ qui est strictement convexe au dessus de $l \leq x \leq u$.
- Prendre $k = 0$ et aller à l'étape 1.

Etape 1 :

- Calculer :

$$d_i(x^k) = \frac{u_i + l_i \gamma_i(x^k)}{1 + \gamma_i(x^k)}, \quad i = 1, \dots, n.$$

- Aller à l'étape 2.

Etape 2 :

- Si $\|d(x^k) - x^k\|$ est inférieure à la tolérance donnée ou β est assez petit alors fin avec $d(x^k)$ comme solution .
- Sinon, faire comme suit:

$$x^{k+1} = x^k + \mu_k(d(x^k) - x^k) \tag{4.8}$$

où

μ_k est un nombre dans $[0,1]$ qui satisfait:

$$e(x^{k+1}, \beta) = \min_{\mu \in [0,1]} e(x^k + \mu(d(x^k) - x^k), \beta).$$

- Poser $\beta = \theta\beta$.
- Poser $k = k + 1$ et aller à l'étape 1.

Remarque :

Une solution exacte de $\min_{\mu \in [0,1]} e(x^k + \mu(d(x^k) - x^k), \beta)$ n'est pas exigée dans l'exécution de la méthode, une solution approximative suffira.

On peut trouver plusieurs moyens de déterminer μ_k (*Minoux* 1986^[18], *Armijo*^[12]).

On remarque que la méthode est peu sensible au point de départ x^0 puisque la fonction barrière est strictement convexe au début de la méthode.

Le théorème suivant montre que quand β est une valeur positive donnée, la méthode converge vers un point stationnaire de $e(x, \beta)$.

Théorème 4.3.1^[3] :

Pour un $\beta > 0$ donné, chaque point limite de x^k , $k = 1, 2, \dots$, produit par le procédé itératif (4.8) est un point stationnaire de $e(x, \beta)$.

Preuve :

On a:

$$\gamma_i(x) = \exp\left(\frac{1}{\beta} \frac{\partial f(x)}{\partial x_i}\right).$$

Poser :

$$\gamma_i^{\min} = \min_{x \in B} \gamma_i(x) \text{ et } \gamma_i^{\max} = \max_{x \in B} \gamma_i(x).$$

Comme $\frac{\partial f(x)}{\partial x_i}$ est continu sur B , alors on aura: $0 < \gamma_i^{\min} < \infty$ et $0 < \gamma_i^{\max} < \infty$.

Poser:

$$h(\omega) = \frac{s + t\omega}{1 + \omega} \text{ avec } s > t.$$

Sa dérivé est donnée par:

$$h'(\omega) = \frac{t - s}{(1 + \omega)^2} < 0.$$

D'où h est une fonction décroissante de ω .

Posant $\omega = \gamma_i(x)$ on obtient:

$$d_i(x) = \frac{u_i + l_i \gamma_i(x)}{1 + \gamma_i(x)} = \frac{u_i + l_i \omega}{1 + \omega}, \text{ qui est de la même forme que } h(\omega).$$

Ce qui implique que pour tout $x \in B$ on a:

$$l_i < \frac{u_i + l_i \gamma_i^{\max}}{1 + \gamma_i^{\max}} \leq d_i(x) \leq \frac{u_i + l_i \gamma_i^{\min}}{1 + \gamma_i^{\min}} < u_i.$$

Posant :

$$x_i^{\min} = \min\left\{x_i^0, \frac{u_i + l_i \gamma_i^{\max}}{1 + \gamma_i^{\max}}\right\} \text{ et } x_i^{\max} = \max\left\{x_i^0, \frac{u_i + l_i \gamma_i^{\min}}{1 + \gamma_i^{\min}}\right\}.$$

Comme $l_i < x_i^0 < u_i$, on aura: $l_i < x_i^{\min}$ et $x_i^{\max} < u_i$.

Poser : $x^{\min} = (x_1^{\min}, \dots, x_n^{\min})^T$ et $x^{\max} = (x_1^{\max}, \dots, x_n^{\max})^T$.

Posant $x^{k+1} = x^k + \mu_k(d(x^k) - x^k)$ et $0 \leq \mu_k \leq 1$.

D'où on peut facilement obtenir x^k , $k = 1, 2, \dots$, satisfaisant :

$$l < x^{\min} \leq x^k \leq x^{\max} < u.$$

Donc, selon le lemme 4.1.1, $d(x^k) - x^k$ est une direction de descente de $e(x, \beta)$ quand $d(x^k) - x^k \neq 0$.

Soit $X = \{x/x^{\min} \leq x \leq x^{\max}\}$ et $\Omega = \{x \in X/\nabla_x e(x, \beta) = 0\}$.

Pour tout $x \in X$, poser :

$$A(x) = \{x + \mu^*(d(x) - x)/u^* \in [0, 1], e(x + \mu^*(d(x) - x), \beta) = \min_{\mu \in [0, 1]} e(x + \mu(d(x) - x), \beta)\}.$$

Dans ce qui suit nous montrons que $A(x)$ est borné au tout point $x \in X \setminus \Omega$.
Soit \bar{x} un point de $X \setminus \Omega$.

Soit $x^q \in X \setminus \Omega$, $q = 1, 2, \dots$, un ordre convergent vers \bar{x} et $y^q \in A(x^q)$, $q = 1, 2, \dots$, un ordre convergent vers \bar{y} .

Pour montrer que $A(\bar{x})$ est borné, il suffit juste de montrer que $\bar{y} \in A(\bar{x})$.

Puisque $\nabla_x e(x^q, \beta) \neq 0$ et $\nabla_x e(\bar{x}, \beta) \neq 0$, on obtient à partir du lemme 4.1.1 que :
 $d(x^q) - x^q \neq 0$ et $d(\bar{x}) - \bar{x} \neq 0$.

Comme $d(x^q)$ est continu . Alors, $d(x^q)$ converge vers $d(\bar{x})$ quand $q \rightarrow \infty$.

Et comme $y^q \in A(x^q)$, alors il existe un certain nombre $\mu_q^* \in [0, 1]$ qui satisfait :

$$y^q = x^q + \mu_q^*(d(x^q) - x^q).$$

Du fait que : $d(x^q) - x^q \neq 0$ on aura:

$$\mu_q^* = \frac{\|y^q - x^q\|}{\|d(x^q) - x^q\|},$$

et quand $q \rightarrow \infty$ alors:

$$\mu_q^* \rightarrow \bar{\mu}^* = \frac{\|\bar{y} - \bar{x}\|}{\|d(\bar{x}) - \bar{x}\|} \text{ avec } \bar{\mu}^* \in [0, 1],$$

D'où :

$$\bar{y} = \bar{x} + \bar{\mu}^*(d(\bar{x}) - \bar{x}).$$

D'autre part, on a $y^q \in A(x^q)$ alors on aura :

$$e(y^q, \beta) \leq e(x^q + \mu(d(x^q) - x^q), \beta), \text{ pour tout } \mu \in [0, 1].$$

Cela implique que :

$$e(\bar{y}, \beta) \leq e(\bar{x} + \mu(d(\bar{x}) - \bar{x}), \beta) \text{ pour } \mu \in [0, 1], \text{ qui prouve que :}$$

$$e(\bar{y}, \beta) = \min_{\mu \in [0, 1]} e(\bar{x} + \mu(d(\bar{x}) - \bar{x}), \beta).$$

Selon la définition de $A(x)$ on a bien: $\bar{y} \in A(\bar{x})$.

Puisque X est borné et $x^k \in X$, $k = 1, 2, \dots$, donc on peut extraire un ordre convergent vers x^k , $k = 1, 2, \dots$.

Soit x^{k_j} , $j = 1, 2, \dots$, l'ordre convergent vers l'ordre x^k , $k = 1, 2, \dots$.

Poser x^* la limite de cet ordre. Donc il suffit de montrer que: $x^* \in \Omega$.

C'est clair que quand $k \rightarrow \infty$, $e(x^k, \beta)$ converge vers $e(x^*, \beta)$ car $e(x, \beta)$ est continu sur B et $e(x^{k+1}, \beta) < e(x^k, \beta)$, $k = 0, 1, \dots$.

Soit l'ordre x^{k_j+1} , $j = 1, 2, \dots$.

Noter que :

$$x^{k_j+1} = x^{k_j} + \mu_{k_j}(d(x^{k_j}) - x^{k_j}) \text{ et } e(x^{k_j+1}, \beta) = \min_{\mu \in [0,1]} e(x^{k_j} + \mu(d(x^{k_j}) - x^{k_j}), \beta).$$

Selon la définition de $A(x)$, on aura: $x^{k_j+1} \in A(x^{k_j+1})$.

Du fait que les x^{k_j+1} , $j = 1, 2, \dots$, sont bornés donc on peut extraire un ordre convergent.

Soit x^{k_j+1} , $j \in K$ un tel ordre .

Soit x le point limite de l'ordre x^{k_j+1} , $j \in K$.

Par absurde, supposant que: $x^* \notin \Omega$.

Du fait que $A(x^*)$ est borné, alors on a bien que: $x \in A(x^*)$.

Donc: $e(x, \beta) < e(x^*, \beta)$ ce qui contredit le fait que $e(x^k, \beta)$ converge quand $k \rightarrow \infty$.
Ce qui implique que $x^* \in \Omega$.

D'où la preuve du théorème.

4.4 Application de la méthode sur certains problèmes :

4.4.1 Problème 1 :

Considérons le problème d'optimisation quadratique suivant :

$$\min f(x) = \begin{cases} c^T x - \frac{1}{2} x^T A A^T x \\ \text{avec : } 0 \leq x_i \leq 1; i = 1, \dots, n. \end{cases}$$

où :

A : est une matrice d'ordre $n * m$ avec des entrées étant des nombres dans $[-1, 1]$.

c : est un vecteur dans les composantes sont des nombres dans $[-1,1]$.

On démarre d'un $0 < x_i^0 < 1 ; i = \overline{1,n}$

La méthode se termine quand $\beta < 0.1$.

La solution est donnée par : $z^* = \text{round}(x^k)$.

La méthode produit toujours un minimum global appliquée sur certains nombres de problèmes.

4.4.2 Problème 2 :

Considérons le problème quadratique suivant :

$$\min f(x) = \begin{cases} c^T x + \frac{1}{2} x^T Q x \\ \text{avec : } -1 \leq x_i \leq 1 ; i = 1, \dots, n. \end{cases}$$

où :

Q : est une matrice symétrique.

$$c = -(Q - uI)z^*.$$

u : est la plus petite valeur propre de Q .

I : est une matrice identité.

$$z^* = (1, -1, 1, -1, \dots, 1, -1).$$

La manière de produire ce type de problème est définie par *Pardalos*(1991)^[25].
Il est facile de constater que z^* est un minimum global du problème posé au dessus.

4.4.3 Problème 3 : Recherche de clique maximum dans un graphe :

Soit $G = (V,E)$: un graphe non orienté.

Où :

$V = \{1,2, \dots, n\}$: est l'ensemble des sommets de G .

E : est l'ensemble d'arêtes de G .

(i,j) : est une arête entre le sommet i et le sommet j .

$$\text{Soit } A_G = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} : \text{matrice d'adjasence de } G.$$

$$\text{où: } \begin{cases} a_{ii} = 0, i = \overline{1, n} \\ a_{ij} = \begin{cases} 1 & \text{si } (i, j) \in E \\ 0 & \text{sinon.} \end{cases} \end{cases}$$

On remarque qu'un graphe est complètement déterminé par sa matrice d'adjacence A_G .

Soit \overline{G} le graphe complémentaire de G défini par :
 $\overline{G} = (V, \overline{E})$ avec :

$$\overline{E} = \{(i, j) / (i, j) \notin E \text{ et } i \neq j\}$$

$$\text{Soit } A_{\overline{G}} = \begin{pmatrix} \overline{a}_{11} & \overline{a}_{12} & \dots & \overline{a}_{1n} \\ \overline{a}_{21} & \overline{a}_{22} & \dots & \overline{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{a}_{n1} & \overline{a}_{n2} & \dots & \overline{a}_{nn} \end{pmatrix} : \text{matrice d'adjacence de } \overline{G}.$$

$$\text{où: } \begin{cases} \overline{a}_{ii} = 0, i = \overline{1, n} \\ \overline{a}_{ij} = 1 - a_{ij}, i \neq j. \end{cases}$$

Soit S un sous-ensemble de V .

Un sous-ensemble de G avec les sommets S noté par $G(S)$ est un graphe $G(S) = (S, E(S))$
 où :

$$E(S) = \{(i, j) / (i, j) \in E \text{ et } i, j \in S\}.$$

Un graphe G est complet si $(i, j) \in E$, pour tout $i \neq j$.

Une clique de G est un sous-ensemble C de V tel que $G(C)$ est complet.

Une clique maximum de G est une clique qui a la cardinalité maximum.

Le problème de recherche de clique maximum consiste à la recherche d'une clique de cardinalité maximum.

Pardalos et Rodgers (1992)^[26] ont démontré que le problème de recherche de clique maximum d'un graphe $G = (V, E)$ est équivalent à la résolution du système :

$$\begin{cases} \min f(x) = x^T Q x \\ 0 \leq x \leq 1. \end{cases}$$

où : $Q = A_{\overline{G}} - I$ avec I : matrice d'identité.

On constate que le problème admet un point minimum global puisque toutes les entrées diagonales de Q sont négatives.

La méthode a été employée pour trouver une clique maximum d'un graphe $G = (V, E)$, où A_G est la matrice d'adjacence produite par le procédé suivant :

- Poser p un nombre dans $(0,1)$
- Pour $i = \overline{1, n}$ et $j = \overline{i + 1, n}$ choisir :
 α un nombre $\in [0,1]$: selon la distribution de probabilité uniforme
- Poser :
$$\begin{cases} a_{ij} = 1 \text{ et } a_{ji} = 1 \text{ si } \alpha \leq p \\ a_{ij} = 0 \text{ et } a_{ji} = 0 \text{ si } \alpha > p. \end{cases}$$

4.5 Langage de programmation :

Matlab est un logiciel de calcul numérique produit par MathWorks.

Matlab est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom Matrix laboratory. Ce langage fut développé dans les années 70 pour des applications impliquant des matrices, l'algèbre linéaire et l'analyse numérique.

Pour le calcul numérique, **Matlab** est beaucoup plus concis que les vieux langages (C, Pascal, Fortran, Basic...). Un exemple : plus besoin de programmer des boucles modifier pour un à un les éléments d'une matrice. On peut traiter la matrice comme une simple variable. **Matlab** contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques.

Les programmes développés dans cette thèse, ont été programmés dans l'environnement Matlab 7.10.0 (R2010a).

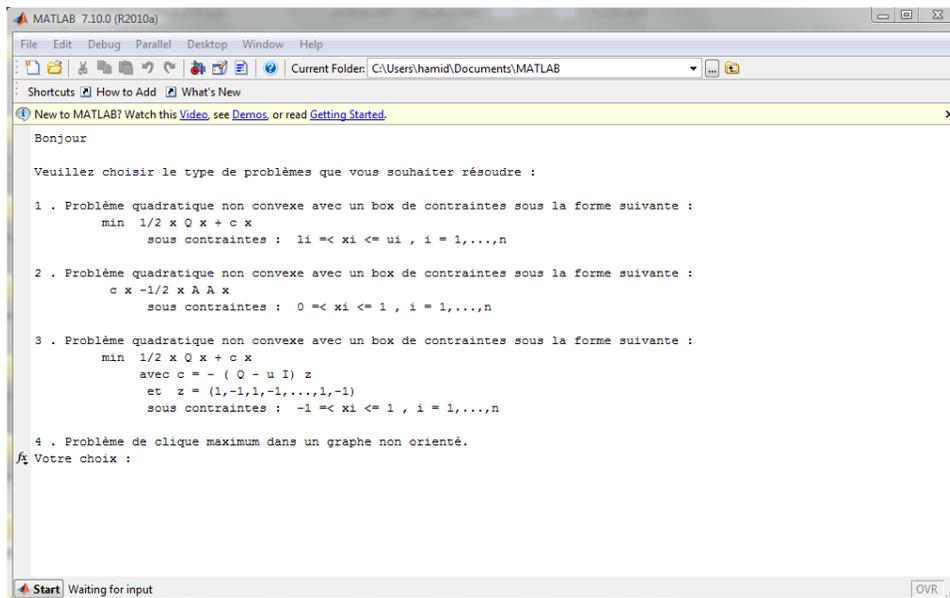


Figure 4.1- L'interface principale de l'algorithme

4.6 Exemples d'application :

Exemple 1 :

Soit le problème suivant :

$$\begin{cases} \min f(x) = \frac{1}{2}(x_1, x_2) \begin{pmatrix} -83.75 & 28.34 \\ 28.34 & -48.28 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (17.72, 15.22) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ \text{S.C: } \begin{cases} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{cases} \end{cases}$$

k	β	x^k	$d(x^k)$	$d(x^k) - x^k$	$\ d(x^k) - x^k\ $
0	200	(0.5,0.5)	(0.5125,0.4934)	(0.0125,-0.0066)	0.0141
1	190	(0.5075,0.4961)	(0.5141,0.4926)	(0.0066,-0.0035)	0.0075
2	180.5	(0.5115,0.4940)	(0.5154,0.4919)	(0.0039,-0.0021)	0.0045
3	171.475	(0.5138,0.4927)	(0.5165,0.4913)	(0.0027,-0.0014)	0.0031
4	162.9012	(0.5155,0.4918)	(0.5177,0.4907)	(0.0022,-0.0012)	0.0025
5	154.7562	(0.5168,0.4911)	(0.5188,0.4901)	(0.0020,-0.0011)	0.0023
10	119.7474	(0.5231,0.4878)	(0.5256,0.4864)	(0.0025,-0.0013)	0.0028
20	71.6972	(0.5445,0.4763)	(0.5500,0.4733)	(0.0055,-0.0029)	0.0062
30	42.9278	(0.5973,0.4473)	(0.6123,0.4389)	(0.0151,-0.0085)	0.0173
40	25.7024	(0.7578,0.3478)	(0.8016,0.3155)	(0.0438,-0.0323)	0.0544
50	15.3890	(0.9676,0.1111)	(0.9803,0.0815)	(0.0128,-0.0296)	0.0322
60	9.2140	(0.9984,0.0148)	(0.9992,0.0095)	(0.0008,-0.0053)	0.0054
70	5.5167	(1.0000,0.0008)	(1.0000,0.0004)	(0.0151,-0.4058) 10^{-3}	$4.0611 \cdot 10^{-4}$
80	3.3031	(1.0000,0.0000)	(1.0000,0.0000)	(0.0025,-0.5528) 10^{-5}	$5.2581 \cdot 10^{-6}$
83	2.8320	(1.0000,0.0000)	(1.0000,0.0000)	(0.0022,-0.9398) 10^{-6}	$9.3976 \cdot 10^{-6}$

Tableau.4.1- Exécution de la méthode de barrière pour l'exemple 1

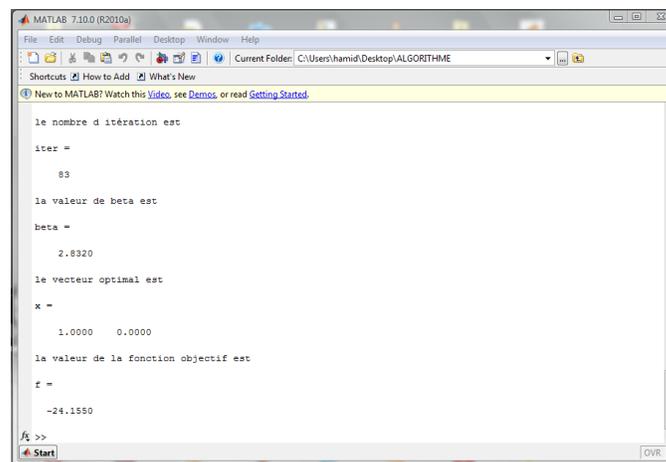


Figure 4.2- Résultat de l'exemple 1

Après exécution de l'algorithme, on aura :
 $x^* = (x_1^*, x_2^*) = (1, 0)$ et $f(x^*) = -24.1550$.

Exemple 2 :

Soit le problème suivant :

$$\begin{cases} \min f(x) = c^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \frac{1}{2}(x_1, x_2, x_3)AA^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\ \text{S.C: } \begin{cases} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \\ 0 \leq x_3 \leq 1 \end{cases} \end{cases}$$

avec : $c = \begin{pmatrix} 2 \\ 3 \\ 3 \\ 4 \\ 7 \\ 9 \end{pmatrix}$ et $A = \begin{pmatrix} -1 & \frac{1}{2} & \frac{3}{4} & -\frac{1}{2} \\ \frac{1}{2} & \frac{4}{4} & -\frac{9}{11} & -\frac{3}{4} \\ \frac{3}{4} & \frac{5}{2} & \frac{3}{4} & -\frac{4}{8} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{3}{4} & -\frac{7}{8} \end{pmatrix}$

Alors : $AA^T = \begin{pmatrix} -1 & \frac{1}{2} & \frac{3}{4} & -\frac{1}{2} \\ \frac{1}{2} & \frac{4}{4} & -\frac{9}{11} & -\frac{3}{4} \\ \frac{3}{4} & \frac{5}{2} & \frac{3}{4} & -\frac{4}{8} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{3}{4} & -\frac{7}{8} \end{pmatrix} \begin{pmatrix} -1 & \frac{1}{2} & \frac{3}{4} \\ \frac{1}{2} & \frac{4}{4} & -\frac{9}{11} \\ \frac{3}{4} & \frac{5}{2} & \frac{3}{4} \\ -\frac{1}{2} & -\frac{3}{4} & -\frac{7}{8} \end{pmatrix} = \begin{pmatrix} 2.0625 & -0.3386 & 0.5 \\ -0.3386 & 2.1219 & 0.8176 \\ 0.5 & 0.8176 & 2.1406 \end{pmatrix}$

k	β	x^k	$d(x^k)$	$d(x^k) - x^k$
0	200	(0.1,0.1,0.1)	(0.4994,0.5.013,0.4995)	(0.3994,0.4013,0.3995)
1	190	(0.3397,0.3408,0.3397)	(0.5001,0.5022,0.5005)	(0.1604,0.1614,0.1608)
2	180.5	(0.4359,0.4376,0.4362)	(0.5004,0.5026,0.5010)	(0.0645,0.0650,0.0648)
3	171.4750	(0.4746,0.4766,0.4751)	(0.5006,0.5029,0.5013)	(0.0259,0.0263,0.0262)
4	162.9012	(0.4902,0.4924,0.4908)	(0.5006,0.5031,0.5014)	(0.0105,0.0107,0.0106)
5	154.7562	(0.4965,0.4988,0.4972)	(0.5007,0.5033,0.5015)	(0.0042,0.0045,0.0044)
6	147.0184	(0.4990,0.5015,0.4998)	(0.5008,0.5035,0.5016)	(0.0017,0.0020,0.0018)
7	139.6675	(0.5001,0.5027,0.5009)	(0.5008,0.5037,0.5017)	(0.7409,0.9811,0.8277) 10^{-3}
8	132.6841	(0.5005,0.5033,0.5014)	(0.5008,0.5039,0.5018)	(0.3401,0.5891,0.4244) 10^{-3}

k	$\ d(x^k) - x^k\ $
0	0.6929
1	0.2787
2	0.1122
3	0.0453
4	0.0184
5	0.0076
6	0.0032
7	0.0015
8	$8.0174 \cdot 10^{-4}$

Tableau.4.2- Exécution de la méthode de barrière pour l'exemple 2

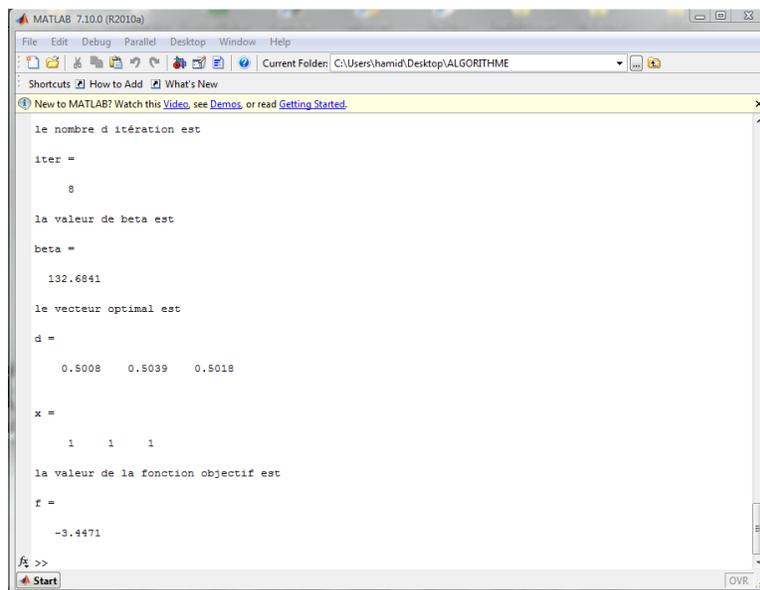


Figure 4.3- Résultat de l'exemple 2

Après exécution de l'algorithme on a : $d = (x_1, x_2, x_3) = (0.5008, 0.5039, 0.5018)$.
 Et comme $x^* = \text{round}(d)$ alors la solution optimale du problème est :
 $x^* = (x_1^*, x_2^*, x_3^*) = (1, 1, 1)$ et la valeur de la fonction objectif est $f(x^*) = -3.4471$.

Exemple 3 :

Soit le problème suivant :

$$\left\{ \begin{array}{l} \min f(x) = c^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \frac{1}{2}(x_1, x_2, x_3, x_4) \begin{pmatrix} -2 & 2 & 1 & -3 \\ 2 & 5 & -4 & 3 \\ 1 & -4 & -5 & -2 \\ -3 & 3 & -2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \\ \text{S.C:} \begin{cases} -1 \leq x_1 \leq 1 \\ -1 \leq x_2 \leq 1 \\ -1 \leq x_3 \leq 1 \\ -1 \leq x_4 \leq 1 \end{cases} \end{array} \right.$$

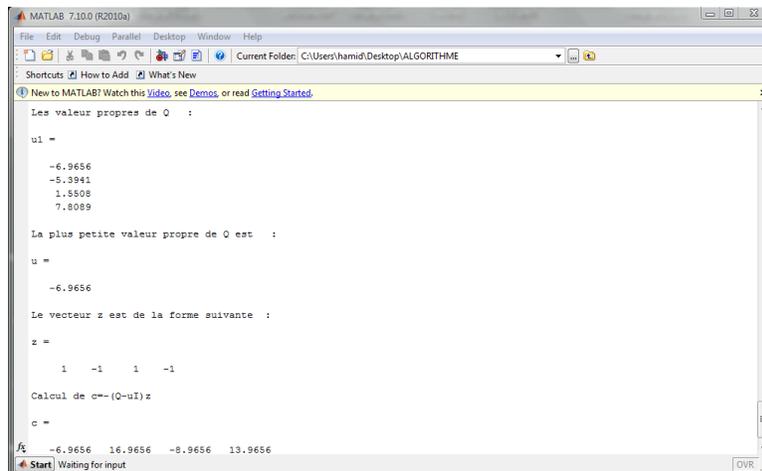


Figure 4.4- Calcul de c

On voit que les valeurs propres de Q sont :
 $u_1 = (-6.9656, -5.3941, 1.5508, 7.8089)$ d'où $u = -6.9656$ est la plus petite valeur propre de Q .

$$z = (1, -1, 1, -1).$$

Donc $c = -(Q - uI)z = (-6.9656, 16.9656, -8.9656, 13.9656)$.

k	β	x^k	$d(x^k)$
0	100	(-0.1,-0.1,-0.1,-0.1)	(-1.6485,-1.4644,-1.6581,-14810)
5	77.3781	(0.0122,-0.1138,0.0152,-0.1062)	(0.0445,-0.1033,0.0540,-0.0881)
10	59.8737	(0.0529,-0.1221,0.0652,-0.1033)	(0.0579,-0.1319,0.0712,-0.1115)
20	35.8486	(0.0886,-0.1945,0.1066,-0.1668)	(0.0963,-0.2095,0.1152,-0.1803)
30	21.4639	(0.1467,-0.3001,0.1697,-0.2644)	(0.1593,-0.3210,0.1828,-0.2843)
40	12.8512	(0.2406,-0.4402,0.2639,-0.4026)	(0.2606,-0.4661,0.2830,-0.4293)
50	7.6945	(0.3863,-0.6023,0.3996,-0.5745)	(0.4163,-0.6298,0.4269,-0.6045)
60	4.6076	(0.5921,-0.7606,0.5901,-0.7472)	(0.6305,-0.7847,0.6271,-0.7732)
70	2.7584	(0.8168,-0.8874,-0.8157,-0.8814)	(0.8492,-0.9045,0.8493,-0.8994)
80	1.6515	(0.9584,-0.9678,0.9593,-0.9662)	(0.9709,-0.9763,0.9716,-0.9753)
90	0.9888	(0.9967,-0.9969,0.9967,-0.9968)	(0.9983,-0.9983,0.9983,-0.9983)
92	0.8924	(0.9983,-0.9984,0.9983,-0.9984)	(0.9992,-0.9992,0.9992,-0.9992)
94	0.8054	(0.9992,-0.9992,0.9992,-0.9992)	(0.9996,-0.9997,0.9996,-0.9997)

k	$d(x^k) - x^k$	$\ d(x^k) - x^k\ $
0	(-1.5485,-1.3644,-1.5581,-1.3810)	2.9316
5	(0.0323,0.0106,0.0388,0.0181)	0.054
10	(0.0050,-0.0098,0.0060,-0.0083)	0.0150
20	(0.0077,-0.0150,0.0086,-0.0135)	0.0232
30	(0.0125,-0.0209,0.0131,-0.0199)	0.0341
40	(0.0200,-0.4661,0.2830,-0.4293)	0.0464
50	(0.0299,-0.0275,0.0273,-0.300)	0.0574
60	(0.0384,-0.0241,0.0370,-0.0260)	0.0640
70	(0.0324,-0.0171,0.0335,-0.0179)	0.0528
80	(0.0125,-0.0085,0.0122,-0.0091)	0.0215
90	(0.0016,-0.0014,0.0015,-0.0015)	0.0030
92	(0.8528,-0.8081,0.8448,-0.8201) 10^{-3}	0.0017
94	(0.4359,-0.4220,0.4333,-0.4258) 10^{-3}	8.5860 10^{-3}

Tableau.4.3- Exécution de la méthode de barrière pour l'exemple 3

```

MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\hamid\Desktop\ALGORITHME
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

le nombre d itération est
iter =
    94
la valeur de beta est
beta =
    0.8054
le vecteur x est
x =
    0.9996 -0.9997 0.9996 -0.9997
Le vecteur optimal est :
z =
    1 -1 1 -1
la valeur de la fonction objectif est
f =
   -37.3526

f >>
Start OVR
    
```

Figure 4.5- Résultat de l'exemple 3

On a trouvé $x = (0.9996, -0.9997, 0.9996, -0.9997)$.

Et comme $x^* = \text{round}(x)$ alors la solution optimale est : $x^* = (x_1^*, x_2^*, x_3^*, x_4^*) = (1, -1, 1, -1)$ d'où $f(x^*) = -37.3526$ est la valeur de la fonction objectif.

Exemple 4 :

Soit le graphe $G1 = (V1, E1)$ non orienté suivant :

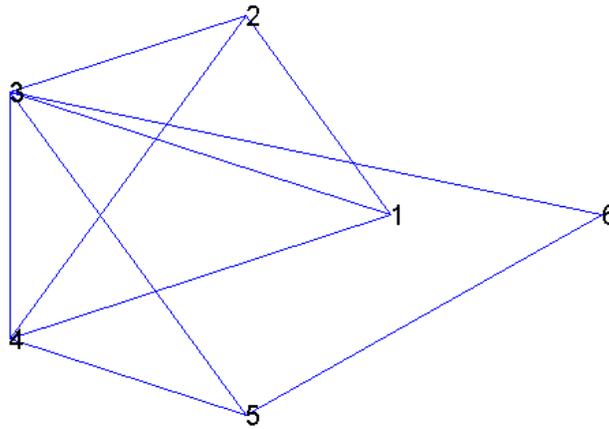


Figure 4.6- **Le graphe** $G1 = (V1, E1)$

Sa matrice d'adjascence est comme suit :

$$A_{G1} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Donc :

$$A_{\overline{G1}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Son graphe est comme suit :

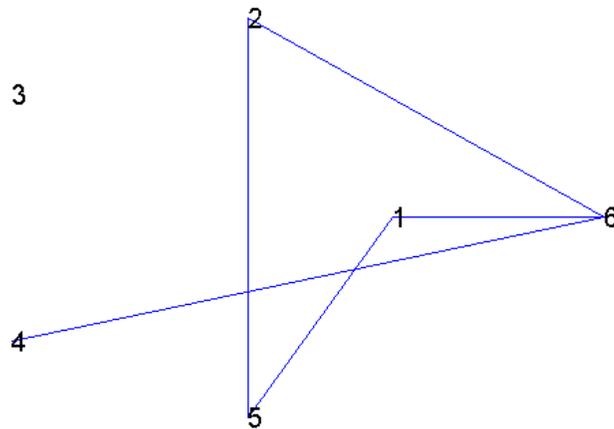


Figure 4.7- Le graphe $\overline{G1}$

En choisissant $\alpha \in [0,1]$ et suivant la distribution de probabilité de la loi uniforme on aura la matrice suivante :

$$A_{G1DBU} = \begin{pmatrix} 0 & Inf & Inf & Inf & -Inf & -Inf \\ 0 & 0 & Inf & Inf & -Inf & -Inf \\ 0 & 0 & 0 & Inf & Inf & Inf \\ 0 & 0 & 0 & 0 & -Inf & -Inf \\ 0 & 0 & 0 & 0 & 0 & Inf \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ensuite, en choisissant un $p \in [0,1]$ et en appliquant la troisième étape du procédé on aura :

$$A_{G1} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Son complémentaire est défini par :

$$A_{\overline{GI}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Et on a : $Q = A_{\overline{GI}} - I$ donc :

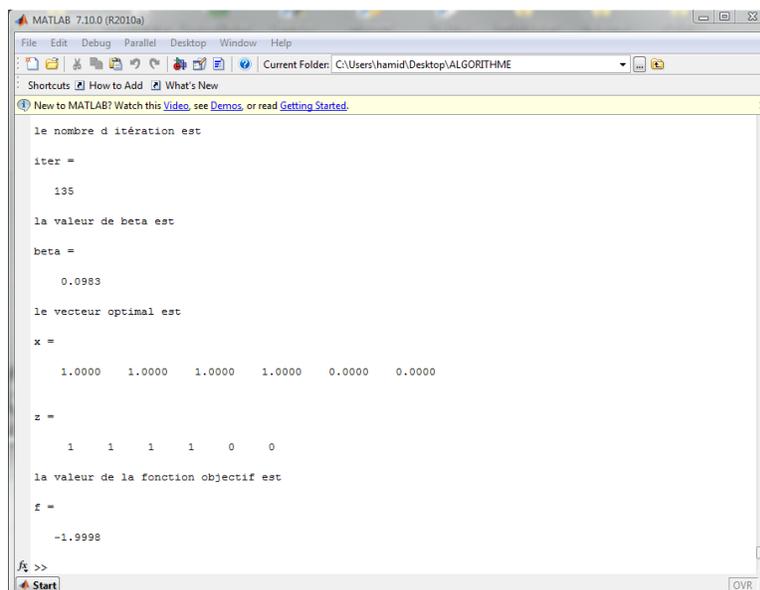
$$Q = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \\ 1 & 1 & 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & -1 \end{pmatrix}$$

k	β	x^k	$d(x^k)$
0	100	(0.5,0.5,0.5,0.5,0.5,0.5)	(0.4988,0.4988,0.5012,0.5000,0.4988,0.4975)
1	95	(0.4993,0.4993,0.5007,0.5000,0.4993,0.4985)	(0.4987,0.4987,0.5013,0.5000,0.4987,0.4974)
5	77.3781	(0.4985,0.4985,0.5015,0.5000,0.4985,0.4971)	(0.4984,0.4984,0.5016,0.5000,0.4984,0.4968)
10	59.8737	(0.4981,0.4981,0.5019,0.5000,0.4981,0.4962)	(0.4979,0.4979,0.5021,0.5000,0.4979,0.4958)
20	35.8486	(0.4968,0.4968,0.5032,0.5000,0.4968,0.4936)	(0.4966,0.4966,0.5035,0.5000,0.4965,0.4930)
30	21.4639	(0.4948,0.4948,0.5054,0.5001,0.4947,0.4893)	(0.4943,0.4943,0.5059,0.5001,0.4942,0.4884)
40	12.8512	(0.4914,0.4914,0.5091,0.5003,0.4912,0.4821)	(0.4906,0.4906,0.5099,0.5004,0.4904,0.4805)
50	7.6945	(0.4859,0.4859,0.5154,0.5008,0.4854,0.4701)	(0.4847,0.4847,0.5167,0.5010,0.4842,0.4675)
60	4.6070	(0.4774,0.4774,0.5261,0.5024,0.4761,0.4500)	(0.4757,0.4757,0.5285,0.5028,0.4740,0.4456)
70	2.7584	(0.4651,0.4651,0.5450,0.5069,0.4608,0.4160)	(0.4628,0.4628,0.5492,0.5082,0.4576,0.4085)
80	1.6515	(0.4505,0.4505,0.5789,0.5206,0.4357,0.3574)	(0.4483,0.4483,0.5868,0.5247,0.4300,0.3443)
90	0.9888	(0.4479,0.4479,0.6422,0.5649,0.3872,0.2518)	(0.4518,0.4518,0.6569,0.5785,0.3742,0.2275)
100	0.5924	(0.5533,0.5533,0.7577,0.7167,0.2347,0.0780)	(0.6002,0.6002,0.7824,0.7430,0.1865,0.0506)
110	0.3545	(0.8905,0.8905,0.9078,0.9056,0.0145,0.0016)	(0.9218,0.9218,0.9283,0.9276,0.0068,0.0005)
120	0.2122	(0.9853,0.9853,0.9853,0.9853,0.0002,0.0000)	(0.9904,0.9904,0.9905,0.9905,0.0001,0.0000)
130	0.1271	(0.9992,0.9992,0.9992,0.9992,0.0000,0.0000)	(0.9996,0.9996,0.9996,0.9996,0.0000,0.0000)
135	0.0983	(0.9999,0.9999,0.9999,0.9999,0.0000,0.0000)	(1.0000,1.0000,1.0000,1.0000,0.0000,0.0000)

k	$d(x^k) - x^k$	$\ d(x^k) - x^k\ $
0	(-0.0012,-0.0012,0.0012,0,-0.0012,-0.0025)	0.0035
1	(-0.0006,-0.0006,0.0006,0.0000,-0.0006,-0.0001)	0.0016
5	(-0.1401,-0.1401,0.1430,0.0019,-0.1411,-0.2841) 10^{-3}	$4.0043 \cdot 10^{-4}$
10	(-0.1660,-0.1660,0.1697,0.0025,-0.1673,-0.3370) 10^{-3}	$4.7484 \cdot 10^{-4}$
20	(-0.2743,-0.2743,0.2846,0.0069,-0.2779,-0.5625) 10^{-3}	$7.9068 \cdot 10^{-4}$
30	(-0.4504,-0.4504,0.4792,0.0193,-0.4605,-0.9397) 10^{-3}	0.0013
40	(-0.0007,-0.0007,0.0008,0.0001,-0.0008,-0.0016)	0.0022
50	(-0.0012,-0.0012,0.0014,0.0002,-0.0012,-0.0026)	0.0036
60	(-0.0018,-0.0018,0.0024,0.0004,-0.0020,-0.0044)	0.0060
70	(-0.0024,-0.0024,0.0043,0.0013,-0.0033,-0.0075)	0.0099
80	(-0.0022,-0.0022,0.0078,0.0041,-0.0056,-0.0132)	0.0171
90	(0.0040,0.0040,0.0147,0.0136,-0.0130,-0.0243)	0.0345
100	(0.0469,0.0469,0.0247,0.0364,-0.0482,-0.0273)	0.0970
110	(0.0313,0.0313,0.0205,0.0220,-0.0077,-0.0011)	0.0540
120	(0.0052,0.0052,0.0051,0.0051,-0.0002,-0.0000)	0.0103
130	(0.4135,0.4135,0.4134,0.4134,-0.0007,-0.0000) 10^{-3}	$8.2690 \cdot 10^{-4}$
135	(0.6432,0.6432,0.6432,0.6432,-0.0002,-0.0000) 10^{-4}	$1.2865 \cdot 10^{-4}$

Tableau.4.4- Exécution de la méthode de barrière pour l'exemple 4

Après l'exécution de l'algorithme on aura les résultats et le graphe suivant :



```

MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\hamid\Desktop\ALGORITHME
Shortcuts How to Add What's New
New to MATLAB? Watch this Video, see Demos or read Getting Started.
le nombre d itération est
iter =
    135
la valeur de beta est
beta =
    0.0983
le vecteur optimal est
x =
    1.0000    1.0000    1.0000    1.0000    0.0000    0.0000
z =
     1     1     1     1     0     0
la valeur de la fonction objectif est
f =
   -1.9998
fx >>
Start OVR
    
```

Figure 4.8- Résultat de l'exemple 4

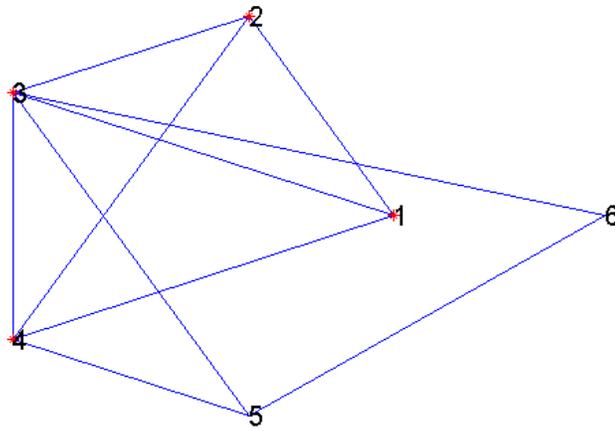


Figure 4.9 La clique maximum du graphe $G1$

Les sommets coloriés en rouge présentent la clique maximum du graphe c'est-à-dire : $C = \{1,2,3,4\}$ est la clique maximum du graphe $G1$.

Exemple 5 :

Soit $G2 = (V2, E2)$ le graphe non orienté suivant :

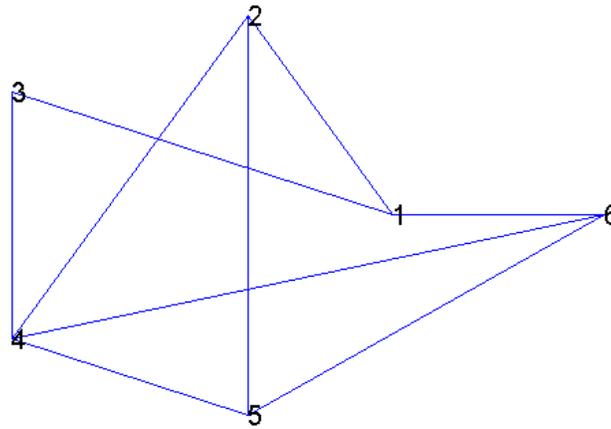


Figure 4.10 **Le graphe** $G2 = (V2, E2)$

Sa matrice d'adjascence est comme suit :

$$A_{G2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Donc :

$$A_{\overline{G2}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Son graphe est comme suit :

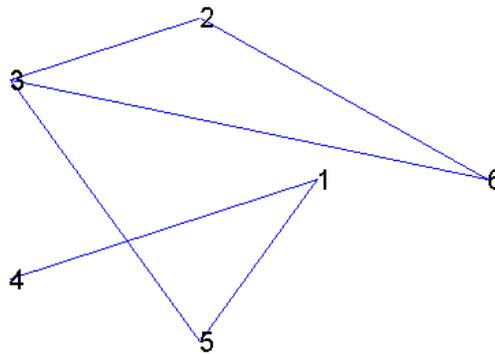


Figure 4.11- Le graphe $\overline{G2}$

En choisissant $\alpha \in [0,1]$ et suivant la distribution de probabilité de la loi uniforme on aura la matrice suivante :

$$A_{G2DBU} = \begin{pmatrix} 0 & Inf & Inf & -Inf & -Inf & Inf \\ 0 & 0 & -Inf & Inf & Inf & -Inf \\ 0 & 0 & 0 & Inf & -Inf & -Inf \\ 0 & 0 & 0 & 0 & Inf & Inf \\ 0 & 0 & 0 & 0 & 0 & Inf \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ensuite, en choisissant un $p \in [0,1]$ et en appliquant la troisième étape du procédé on aura :

$$A_{G2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Son complémentaire est défini par :

$$A_{\overline{G^2}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Et on a : $Q = A_{\overline{G}} - I$ donc :

$$Q = \begin{pmatrix} -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \end{pmatrix}$$

k	β	x^k	$d(x^k)$
0	100	(0.9,0.9,0.9,0.9,0.9,0.9)	(0.4988,0.4988,0.5012,0.5000,0.4988,0.4975)
5	77.3781	(0.5025,0.5025,0.5010,0.5041,0.5025,0.5025)	(0.4984,0.4984,0.4967,0.5000,0.4984,0.4984)
15	46.3291	(0.4975,0.4975,0.4951,0.5000,0.4975,0.4975)	(0.4973,0.4973,0.4946,0.5000,0.4973,0.4973)
30	21.4639	(0.4946,0.4948,0.4893,0.5001,0.4948,0.4948)	(0.4942,0.4943,0.4884,0.5001,0.4943,0.4943)
45	9.9440	(0.4884,0.4889,0.4771,0.5003,0.4889,0.4889)	(0.4874,0.4880,0.4751,0.5003,0.4880,0.4880)
60	4.6070	(0.4749,0.4773,0.4512,0.5012,0.4774,0.4773)	(0.4727,0.4755,0.4470,0.5014,0.4757,0.4755)
75	2.1344	(0.4447,0.4566,0.3966,0.5060,0.4577,0.4566)	(0.4395,0.4537,0.3878,0.5072,0.4552,0.4537)
90	0.9888	(0.3652,0.4313,0.2805,0.5348,0.4464,0.4313)	(0.3491,0.4296,0.2611,0.5428,0.4498,0.4296)
105	0.4581	(0.0764,0.4630,0.0499,0.7716,0.7154,0.4630)	(0.0440,0.4728,0.0301,0.8202,0.7834,0.4728)
120	0.2122	(0.0002,0.4995,0.0002,0.9853,0.9852,0.4995)	(0.0001,0.4997,0.0001,0.9904,0.9904,0.4997)
135	0.0983	(0.0000,0.5000,0.0000,0.9999,0.9999,0.5000)	(0.0000,0.5000,0.0000,1.0000,1.0000,0.5000)
150	0.0456	(0.0000,0.0326,0.0000,1.0000,1.0000,0.9674)	(0.0000,0.0000,0.0000,1.0000,1.0000,1.0000)
160	0.0273	(0.0000,0.0000,0.0000,1.0000,1.0000,1.0000)	(0.0000,0.0000,0.0000,1.0000,1.0000,1.0000)
162	0.0246	(0.0000,0.0000,0.0000,1.0000,1.0000,1.0000)	(0.0000,0.0000,0.0000,1.0000,1.0000,1.0000)

k	$d(x^k) - x^k$	$\ d(x^k) - x^k\ $
0	(-0.0012,-0.0012,0.0012,0,-0.0012,-0.0025)	0.0035
5	(-0.0042,-0.0042,-0.0042,-0.0041,-0.0042,-0.0042)	0.0102
15	(-0.2180,-0.2139,-0.4335,0.0016,-0.2139,-0.2139) 10^{-3}	$6.1052 \cdot 10^{-4}$
30	(-0.4700,-0.4507,-0.9300,0.0097,-0.4504,-0.4507) 10^{-3}	0.0013
45	(-0.0010,-0.0009,-0.0020,0.0000,-0.0009,-0.0009)	0.0027
60	(-0.0022,-0.0018,-0.0042,0.0002,-0.0018,-0.0018)	0.0057
75	(-0.0052,-0.0029,-0.0088,0.0012,-0.0025,-0.0029)	0.0113
90	(-0.0161,-0.0017,-0.0193,0.0080,0.0034,-0.0017)	0.0267
105	(-0.0324,0.0098,-0.0199,0.0486,0.0681,0.0098)	0.0929
120	(-0.0002,0.0003,-0.0001,0.0052,0.0052,0.0003)	0.0074
135	(-0.0002,0.0023,-0.0002,0.6432,0.6432,0.6432) 10^{-4}	$9.0969 \cdot 10^{-5}$
150	(-0.0000,-0.0326,-0.0000,0.0000,0.0000,0.0326)	0.0462
160	(-0.0000,-0.3422,-0.0000,0.0000,0.0000,0.3422) 10^{-5}	$4.8398 \cdot 10^{-6}$
162	(-0.0000,-0.5476,-0.0000,0.0000,0.0000,0.5476) 10^{-6}	$7.7437 \cdot 10^{-7}$

Tableau.4.5- Exécution de la méthode de barrière pour l'exemple 5

Après l'exécution de l'algorithme on aura les résultats et le graphe suivant :

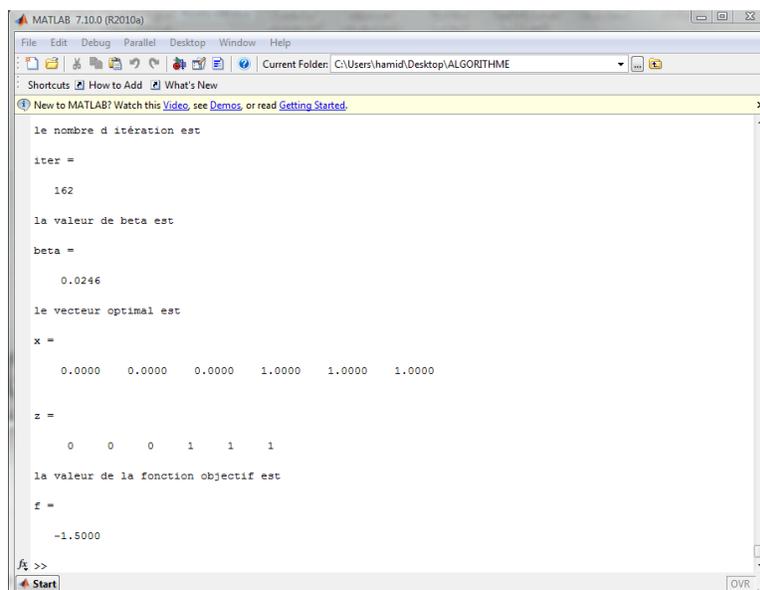


Figure 4.12- Résultat de l'exemple 5

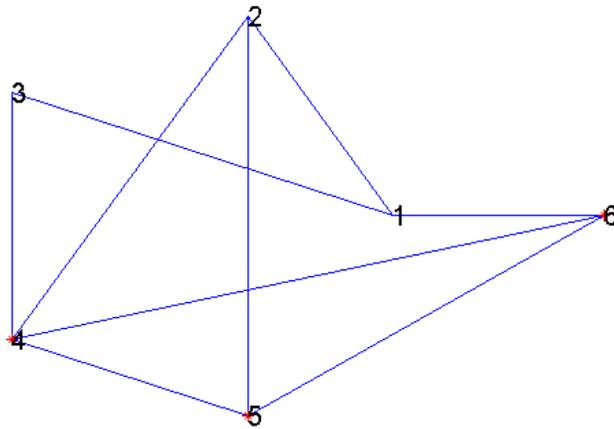


Figure 4.13 La clique maximum du graphe G_2

Les sommets coloriés en rouge présentent la clique maximum du graphe c'est-à-dire : $C = \{4,5,6\}$ est la clique maximum du graphe G_2 .

Conclusion

En conséquence, et après avoir résolu d'autres exemples, on conclut que l'algorithme décrit précédemment peut être utilisé pour résoudre les différents types de problèmes posés. La méthode définie converge vers la solution des problèmes quadratiques non convexes définis sur un box.

Conclusion générale

A travers ce mémoire, notre objectif était l'étude théorique puis l'implémentation de la méthode de la fonction barrière pour les problèmes quadratiques non convexes définis sur un box. Pour ce faire, nous avons étudié et implémenté une approche de résolution. Pour cela, nous avons d'abord introduit quelques classiques importants sur la convexité. Afin d'analyser ou de résoudre de manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer de conditions d'optimalité. En effet, celles-ci nous servent non seulement à vérifier la validité des solutions obtenues, mais souvent l'étude de ces conditions aboutit au développement des algorithmes de résolution eux-mêmes.

Nous nous sommes intéressés aux différentes méthodes de pénalité intérieure avec un aperçu général sur la méthode de *Karmarkar*. Nous avons présenté ensuite la méthode de la fonction barrière pour les problèmes quadratiques non convexes définis sur un box. Elle est basée sur la diminution du paramètre de barrière qui consiste simplement à suivre une direction de descente de façon itérative jusqu'à l'obtention d'un bon minimiseur.

Nous avons finalisé notre travail par l'élaboration du programme de cette méthode, son application sur certains problèmes et son implémentation nous a permis de résoudre différents exemples qui nous ont prouvé que la méthode semble efficace et décisive mais il est difficile de montrer théoriquement que la méthode produit toujours une solution optimale ou approchée.

Ce sujet passionnant n'a bien sûr pas été épuisé à travers ce projet et de nombreux défis restent ouverts, en particulier la combinaison de la méthode de la fonction barrière et l'algorithme de branch-and-bound, afin de fournir une approche efficace pour la résolution du problème de clique maximum.

Bibliographie

- [1] ANTOINE. X, DREYFUSS. P ET PRIVAT. Y. *Introduction à l'optimisation: aspects théoriques, numériques et algorithmes*. ENSMN-ENSEM2A (2006-2007).(NANCY, LE 19 JUILLET 2007).
- [2] BERTSEKAS .D.P. *Nonlinear programming*. ATHENA SCIENTIFIC, (1999).
- [3] CHUANGYIN DANG AND LEI XU. *A Barrier Function Method for the Nonconvex Quadratic Programming Problems with Box Constraints*. Journal of Global Optimization 18: 165-188. (2000).
- [4] CHARLES GILBERT .J. *Optimisation Différentiable*. (19 FÉVRIER 2008).
- [5] CHRISTOS PAPADIMITRIOU. *Computational Complexity*. ADDISON-WESLEY.(1993: ISBN 0-201-53082-1).
- [6] DIETER VANDENBUSSCHE, GEORGE L. NEMHAUSER. *A branch-and-cut algorithm for nonconvex quadratic programs with box constraints*. MATH. PROGRAM, SER. A 102: 559-575. (2005).
- [7] DAVID YANG GAO. *Solutions and Optimality criteria to box constrained nonconvex minimization problems*. JOURNAL OF INDUSTRY AND MANAGEMENT OPTIMIZATION: 293-304. (2007).
- [8] GUY COHEN. *Convexité et Optimisation*. ECOLE NATIONALE DES PONTS ET CHAUSSÉES ET INRIA. EDITION (2000-2006).
- [9] GAUVIN. J. *Théorie de la programmation mathématique non convexe*. LES PUBLICATIONS CRM, MONTRÉAL. (1992)
- [10] HOPFIELD. J.J. *Neurons with graded response have collective computational properties like those of two-state neurons*. PROC. NAT. ACAD. SCI U.S.A.81: 3088-3092. (1984).

- [11] ILANADLER. *An implementation of Karmarkar's algorithm for linear programming*. MATHEMATICAL PROGRAMMING (1995).
- [12] JEAN-CHRISTOPHE CULIOLI. *Introduction à l'optimisation*. ECOLE DES MINES DE PARIS. EDITION MARKETING. ISBN 2-7298-9428-4. (1994).
- [13] KELLEY. C.T. *Iterative Methods for Optimization*. SIAM PUBLICATION, PHILADELPHIA. (1999).
- [14] LUENBERGER. D. *Introduction to linear and nonlinear programming*. ADDISON-WESLEY, 1973.
- [15] MINOUX. M. *Programmation mathématique*. (TOME 2), DUNOD (1983).
- [16] MINOUX MICHEL. *Programmation Mathématique Théorie et Algorithmes*. COLLECTION TECHNIQUE ET SCIENTIFIQUE DES TÉLÉCOMMUNICATIONS, VOLUME 1. ANNÉE 1983.
- [17] MIKAEL ADLERS. *Sparse Least Squares Problems with Box Constraints*. PRINTED IN SUEDEN BY UNITRYCK, LINKÖPNIG, (1998)
- [18] MINOUX. M. *Mathematical Programming: Theory and Algorithms*. JOHN WILEY AND SONS. (1986).
- [19] MURTY. K.G AND KABADI. S.N. *Some NP-complete problems in quadratic and nonlinear programming*. MATH. PROG. 39: 117-129. (1987).
- [20] NESTEROV. Y.E, NEMIROVSKII. A.S. *An interior point method for generalized linear-fractional programming*. RAPPORT DE RECHERCHE: 461, 481, 496. (1993).
- [21] NESTEROV. Y.E, NEMIROVSKII. A.S. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM STUDIES IN APPLIED MATHEMATICS 13. SIAM, PHILADELPHIA. (1994).
- [22] NORBERT. Y, OUELLET. R ET PARENT. R. *La recherche opérationnelle*. GAËTAN MORIN EDITEUR, MONTRÉAL-PARIS. (1995).
- [23] PARDALOS. P.M AND ROSEN. J.B. *Constrained Global Optimization, Algorithms and Applications*. LECTURE NOTES IN COMPUTER SCIENCES 268, SPRINGER. (1987).

- [24] PARDALOS. P.M AND RODGERS. G.P. *Computational aspects of a branch and bound algorithm for quadratic zero-one programming.* COMPUTING 45: 131-144. (1990).
- [25] PARDALOS. P.M. *Construction of test problems in quadratic bivalent programming.* ACM TRANSACT MATH SOFTWARE 17: 74-87. (1991).
- [26] PARDALOS. P.M AND RODGERS. G.P. *A branch and bound algorithm for the maximum clique problem.* COMPUT. OPER. RES. 19: 363-375 (1992).
- [27] RENEGAR. J. *A Mathematical View of Interior-Point Methods in Convex Optimization.* MPS/SIAM SERIES ON OPTIMIZATION3. SIAM. (2001).
- [28] RON. S, DEMBO AND ULRICH TULOWITZKI. *On the minimization of the quadratic functions subject to box constraints.* THE U.S. DEPARTMENT OF TRANSPORTATION UNDER GRANT TO NO: CT-06-0011.
- [29] TERLAKY. T. *Interior Point Methods of Mathematical Programming.* APPLIED OPTIMIZATION SERIES, KLUWER ACADEMIC PUBLISHERS. (1996).
- [30] VERWAERDE. D ET LAURENT-GENGOUX. P. *Optimisation.* ECOLE CENTRALE PARIS (2006-2007).
- [31] WRIGHT. M.H. *Interior Methods for Constrained Optimization.* ACTA NUMERICA, CAMBRIDGE UNIVERSITY PRESS, NEW YORK: 341-407. (1992).
- [32] YE. Y. *On affine scaling algorithms for nonconvex quadratic programming.* MATHEMATICAL PROGRAMMING 56: 285-300. (1992). NORTH-HOLLAND.