

Table des matières

0	INTRODUCTION	4
1	OPTIMISATION GLOBALE LIPSCHITZIENNE	9
1.1	Introduction	9
1.2	Optimisation lipschitzienne unidimensionnelle	10
1.2.1	Enveloppes en dents de scie	12
1.2.2	Méthode d'Evtushenko	15
1.2.3	Méthode de Piyavskii	17
1.2.4	Méthode de Shoen	20
1.3	Méthode Branch-and-Bound	22
1.3.1	Un prototype de la méthode Branch-and-Bound	23
1.3.2	Méthode Branch-and-Bound pour minimiser une fonction lipschitzienne sur un pavé de \mathbb{R}^n	27
1.4	Méthode de l'indicateur de relief	30
1.4.1	Séparateurs pour la fonction f sur l'ensemble D	31
1.4.2	Un critère d'optimalité globale	34
1.4.3	La méthode de l'indicateur de relief	38
1.5	Conclusion	42
2	LA METHODE ALIENOR COUPLEE AVEC L'ALGORITHME DE BRENT	44

2.1	Introduction	44
2.2	La méthode Aliénor de base	45
2.2.1	La première transformation réductrice	46
2.2.2	Complexité de la méthode Aliénor	49
2.2.3	Quelques variantes de la méthode Aliénor	50
2.3	La méthode de Brent	51
2.3.1	Algorithme de Brent	52
2.4	La méthode Aliénor couplée avec l'algorithme de Brent	53
2.4.1	Algorithme Aliénor-Brent	55
2.4.2	Convergence de la méthode	58
2.4.3	Tests numériques	59
2.5	Conclusion	62
3	UNE NOUVELLE FONCTION FILLED	63
3.1	Introduction	63
3.2	Méthode de la fonction Filled	64
3.2.1	Analyse du facteur poids	67
3.2.2	L'algorithme de la fonction Filled	68
3.3	Une nouvelle fonction Filled	70
3.4	Etude comparative entre les fonctions Filled	74
3.4.1	Comparaisons entre la fonction K et la fonction Q	74
3.4.2	Comparaisons entre la fonction K et la fonction H	75
3.5	Conclusion	76

Chapitre 0

INTRODUCTION

Dans la vie courante, nous sommes fréquemment confrontés à des problèmes d'optimisation plus au moins complexes. Cela peut commencer au moment où l'on tente de ranger son bureau, de placer son mobilier, et aller jusqu'à un processus industriel, par exemple pour la planification des différentes tâches. Ces problèmes peuvent être exprimés sous la forme générale d'un "problème d'optimisation". On définit alors une fonction objectif (fonction de coût ou fonction profit), que l'on cherche à optimiser (minimiser ou maximiser) par rapport à tous les paramètres (contraintes) concernés. Une telle fonction 'objectif ' présente généralement un grand nombre de solutions non optimales.

Il existe un grand nombre de méthodes d'optimisation, qu'on peut partager en deux catégories, celles qui permettent de déterminer un minimum (maximum) local, ces méthodes sont appelées méthodes locales, et celles qui s'efforcent de déterminer un optimum global, ces méthodes sont appelées méthodes globales.

Ces deux types de méthodes ne s'excluent pas mutuellement. Afin d'améliorer les performances d'une méthode on peut combiner les deux types d'algorithmes. Une recherche globale permet de bien explorer l'espace de recherche, et une recherche locale permet de bien exploiter une zone susceptible de contenir un minimum global, localisé lors de l'exploration du domaine de recherche.

Dans la majorité des problèmes pratiques d'optimisation, il est important de déter-

miner les extréma absolus. Ce n'est que rarement que les solutions locales suffisent. La minimisation d'une fonction convexe, par exemple, sur un ensemble convexe peut être résolue en utilisant les méthodes locales ; puisque dans ce cas le minimum local coïncide avec le minimum global. Les méthodes numériques qui permettent de chercher les solutions globales des problèmes à plusieurs variables, malgré leur importance, n'avaient pas été jusqu'alors très efficaces étant donné leur complexité et la longueur du temps de calcul.

On peut classer les méthodes d'optimisation globale en deux grandes familles. La première est la classe des méthodes déterministes, comme leur nom l'indique, elles ne laissent aucune place au hasard et conduiront pour un contexte initial donné à une même solution finale. Pour ces méthodes, l'exploration de l'espace des solutions se fait grâce à des procédures de recherche qui sont élaborées à partir de la constante de Lipschitz, des dérivées ou d'autres informations concernant la fonction à optimiser. Appartenant à cette classe, les méthodes de recouvrement ont la réputation d'être efficaces en dimension 1, mais la généralisation des mêmes idées au cas multidimensionnel pose d'énormes difficultés d'ordre pratique.

La deuxième classe est celle des méthodes stochastiques qui se basent sur la recherche aléatoire (au hasard) d'un optimum. Ainsi, partant du même point initial, plusieurs exécutions successives de ces méthodes, pourront conduire à des résultats différents telles que la méthode du recuit simulé, la méthode Monte-Carlo etc. Ces méthodes présentent un inconvénient majeur ; on ne peut garantir leur convergence que d'une manière asymptotique ou pas du tout, et l'obtention de l'optimum global n'est pas garantie ; il est seulement repéré avec une probabilité proche de 1.

Cependant, certaines de ces méthodes possèdent un avantage, elles ne requièrent en effet comme seule donnée du problème que les évaluations de la fonction à optimiser. Cela est très intéressant dans le cas des problèmes à paramètres discrets ou lorsque la valeur de la fonction nécessite une résolution numérique.

Dans cette thèse, on s'intéresse aux méthodes déterministes et notre attention sera

portée sur une approche parmi les plus prometteuses, il s'agit de la méthode liée à la transformation réductrice Aliénor. La plupart des algorithmes mis en oeuvre à partir de cette méthode ont prouvé leur efficacité dans de différentes situations [2], [4], [7]. Elle est basée sur l'idée de ramener l'optimisation d'une fonction multivariable à celle d'une fonction d'une seule variable ; pour ainsi disposer de toutes les méthodes performantes et les techniques connues faisant intervenir une seule variable [3], [17], [33], [35].

Dans le premier chapitre, on décrit quelques méthodes pour l'optimisation globale lipschitzienne. Ces méthodes ont été construites pour exploiter théoriquement le fait que la fonction objectif soit lipschitzienne, notamment :

- Les méthodes basées sur la construction d'enveloppe en dents de scie pour l'optimisation globale unidimensionnelle lipschitzienne, telles que la méthode d'Evtushenko, la méthode de Piyavskii-Shubert, la méthode de Shoen etc.

- La méthode Branch-and-Bound, dont la stratégie consiste à déterminer les sous-ensembles qui ne peuvent pas contenir un minimiseur global, les éliminer et continuer la recherche dans la partie restante du domaine d'optimisation.

- La méthode de l'indicateur de relief qui a été donnée par Thach et Tuy (1990), elle est destinée à aller au-delà de l'optimalité locale dans l'optimisation globale générale, et procurer des procédures de solution pour certaines classes de problèmes spéciaux.

En plus de la constante de Lipschitz, on fait appel parfois aux dérivées premières et secondes de f lorsqu'elles existent.

Dans le second chapitre, on propose une nouvelle variante d'Aliénor couplée avec l'algorithme de Brent. Nous verrons que cette méthode mixte offre des perspectives très intéressantes pour diminuer le temps de calcul. D'abord on présentera la méthode de la transformation réductrice Aliénor de base, qui a été élaborée par Y. Cherruault et A. Guillez (dans les années 80). Elle a été essentiellement développée pour l'optimisation globale, mais elle s'est révélée être une approche universelle, car elle peut être utilisée pour approximer des fonctions à plusieurs variables, pour résoudre des systèmes d'équations fonctionnelles et autres problèmes multidimensionnels [1], [3], [15].

L'idée de la méthode consiste à effectuer, à l'aide des courbes α -denses, une transformation qui permet de ramener le problème multidimensionnel à un problème unidimensionnel, pour pouvoir utiliser les méthodes unidimensionnelles connues pour leur efficacité et performance. La plupart des algorithmes mis en oeuvre à partir de cette méthode ont prouvé leur efficacité dans de différentes situations [2], [4], [7]. Ensuite, on donnera, brièvement, l'algorithme de Brent, suivi par une description détaillée de la nouvelle méthode mixte Aliénor-Brent, avec une étude théorique complète et on terminera ce chapitre par une série d'applications sur des fonctions tests.

Dans le troisième chapitre, nous présenterons d'abord la méthode de la fonction Filled en donnant les fonctions Filled les plus connues et les plus utilisées. Les méthodes de la fonction Filled sont utilisées pour résoudre les problèmes d'optimisation globale multidimensionnelle, plusieurs fonctions de ce type ont été reportées dans la littérature. Cependant ces fonctions traditionnelles nécessitent beaucoup de calculs ; ceci est dû au terme exponentiel ou au multiples paramètres intervenant dans leurs formulations.

Ensuite, nous tenterons de construire une nouvelle fonction Filled à partir d'une fonction objectif lipschitzienne. Nous verrons que cette fonction offre des perspectives intéressantes pour améliorer les méthodes des fonctions Filled déjà existantes.

NOTATIONS

$\overline{\mathbb{R}}$	$\mathbb{R} \cup \{-\infty, +\infty\}$
$\ x\ $	norme euclidienne de x .
$[x]$	partie entière de x .
$\text{int } X = \overset{\circ}{X}$	intérieur de l'ensemble X .
$\text{cl } X = \overline{X}$	clôture de l'ensemble X .
∂X	frontière de X .
∇f	gradient de la fonction f .
$H = \nabla^2 f$	le hessien de la fonction f .
$\min f(X)$	minimum global de la fonction f sur l'ensemble X .
$\arg \min f(X)$	ensemble des minimiseurs globaux de f sur l'ensemble X .

Chapitre 1

OPTIMISATION GLOBALE LIPSCHITZIENNE

1.1 Introduction

Les problèmes d'optimisation remontent à des temps aussi anciens que les mathématiques elles mêmes. Pour mémoire, les premières techniques de résolution sont attachées à des noms comme Newton, Lagrange ou Cauchy. Cependant, durant ces 30 ou 40 dernières années, d'importants progrès ont été réalisés grâce notamment à l'arrivée des ordinateurs. Ceux-ci ont permis de résoudre dans des temps relativement courts des problèmes de plus en plus complexes dans de très nombreux domaines de la science.

L'optimisation globale a pour but la résolution du problème général suivant :

$$(P) \quad \begin{cases} \min f(x) \\ g_i(x) \leq 0, i = 1, 2, \dots, m \\ x \in X \end{cases}$$

Où $X \subset \mathbb{R}^n$ est un ensemble compact appelé région faisable et $f : X \rightarrow \mathbb{R}$ une fonction appelée objectif, et $g_i : X \rightarrow \mathbb{R}$ ($i = 1, 2, \dots, m$). On a besoin alors, de déterminer la valeur $f^* = \min f(x)$ et usuellement aussi un point x^* dans X tel que $f(x^*) = f^*$.

- Si $m = 0$ et $X = \prod_{i=1}^n [a_i, b_i]$ un pavé de \mathbb{R}^n , alors le problème (P) est dit problème multidimensionnel non contraint.

- Si $n = 1$, le problème (P) est unidimensionnel.

- Si f et g_i ($i = 1, 2, \dots, m$) sont lipschitziennes sur le compact $X \subset \mathbb{R}^n$, on dit que le problème (P) est lipschitzien.

Rappelons qu'une fonction $h : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$, est dite lipschitzienne si elle vérifie la condition suivante :

$$\forall x \in X, \forall y \in X; |h(x) - h(y)| \leq L \|x - y\|,$$

où L est une constante dite la constante de Lipschitz, et $\|\cdot\|$ indique la norme euclidienne (on peut aussi considérer d'autres normes).

1.2 Optimisation lipschitzienne unidimensionnelle

On considère une fonction f , à une seule variable, lipschitzienne avec la constante de Lipschitz L , définie sur l'intervalle $[a, b]$, i.e.,

$$\forall x, y \in [a, b], |f(x) - f(y)| \leq L |x - y|.$$

Le but est de résoudre le problème (PL) suivant :

$$(PL) \begin{cases} \min f(x) \\ x \in [a, b] \end{cases}$$

pour trouver un point x^* tel que

$$f^* = f(x^*) = \min \{f(x) / x \in [a, b]\}.$$

Ce problème, relativement simple, est intéressant parce qu'il se présente dans plusieurs applications et puisque aussi quelques algorithmes pour résoudre (PL) peuvent

être facilement prolongés au cas multidimensionnel.

Il est clair qu'il n'existe pas d'algorithme qui peut résoudre (PL) en un nombre fini d'évaluations de la fonction objectif f .

En effet, supposons qu'il existe un algorithme A à convergence finie, qui trouve un minimiseur global x^* après k itérations; i.e., on a

$$f^* = f(x^*) = \min \{f(x_i) / i = 1, 2, \dots, k \text{ et } k > 1\}.$$

Posons $X_k = \{x_1, x_2, \dots, x_k\}$ et soit $f(X_k)$ l'ensemble des valeurs de la fonction correspondantes, i.e.,

$$f(X_k) = \{f(x_1), f(x_2), \dots, f(x_k)\}.$$

Soit $x_j \in X_k \setminus \{x^*\}$ le point d'évaluation différent de x^* et qui est proche de x^* à gauche (si un tel point n'existe pas, alors une argumentation similaire se fait pour le point dans $X_k \setminus \{x^*\}$ différent de x^* et qui est proche de x^* à droite).

Considérons la fonction

$$f^1(x) = \begin{cases} f(x) & , x \in [a, x_j] \cup [x^*, b] \\ \max \{f(x_j) - L(x - x_j), f(x^*) - L(x^* - x)\} & , x \in [x_j, x^*] \end{cases}$$

Il est clair que f^1 est une fonction lipschitzienne sur $[a, b]$, et il est facile de voir, directement, par un raisonnement géométrique que le minimum global de f^1 est atteint au point

$$\bar{x} = \frac{x_j + x^*}{2} + \frac{f(x_j) - f(x^*)}{2L},$$

avec

$$f^1(\bar{x}) = \frac{1}{2} (f(x_j) + f(x^*) - L(x^* - x_j)) < f(x^*),$$

quand $f(x^*) > f(x_j) - L(x_j - x^*)$.

Cependant, la stratégie de l'algorithme A dépend seulement de L , X_k et $f(X_k)$ qui

coïncident pour f et f^1 . Alors, on a

$$f(x^*) = \min_{i=1,2,\dots,k} f(x_i) = \min_{i=1,2,\dots,k} f^1(x_i),$$

et l'algorithme A arrive à la conclusion que x^* est aussi un minimiseur global de f^1 , ce qui est contradictoire.

1.2.1 Enveloppes en dents de scie

Soit le problème (PL)

$$(PL) \quad \begin{cases} \min f(x) \\ x \in [a, b] \end{cases}$$

Généralement, au lieu de résoudre le problème (PL) , i.e., trouver un minimiseur global x^* tel que $f^* = f(x^*)$, on résout le problème (PL_ε) qui consiste à trouver un point $x_\varepsilon^* \in [a, b]$ tel que pour un petit $\varepsilon > 0$ on a

$$f_\varepsilon^* = f(x_\varepsilon^*) \leq f^* + \varepsilon, \quad (1)$$

Il est clair que le problème (PL_ε) peut être résolu par un algorithme fini.

En effet, la première possibilité consiste à faire évaluer f en des points prédéterminés de la forme

$$x_i = a + \frac{(2i-1)\varepsilon}{L} \quad (i = 1, 2, \dots, k), \quad (2)$$

où $k = \left\lceil \frac{L(b-a)}{2\varepsilon} \right\rceil + 1$, avec $[\cdot]$ désigne la partie entière d'un nombre réel ; dans ce cas, k est le plus petit nombre naturel vérifiant $k \geq \frac{L(b-a)}{2\varepsilon}$, et on aura un point vérifiant (1).

Un tel algorithme, où les points d'évaluations sont prédéterminés est appelé algorithme passif, puisque le pas de déplacement est fixé et ne dépend pas des valeurs de la fonction. En contrepartie, il y a l'algorithme séquentiel (itératif), dans lequel le choix

des nouveaux points d'évaluations dépend des informations acquises dans les itérations précédentes.

Pour la plupart des fonctions f , le nombre de points d'évaluations exigé pour résoudre (PL_ε) , sera beaucoup plus petit en utilisant un algorithme séquentiel convenable, qu'avec un algorithme passif [14].

Il a été montré, indépendamment, par Surkharev et Ivanov que le nombre des points d'évaluations nécessaire pour résoudre (PL_ε) est le même pour un algorithme passif et un algorithme séquentiel, et cela dans le pire des cas, i.e., dans le cas où f est une fonction constante sur $[a, b]$ [13].

Soient $X_k = \{x_1, x_2, \dots, x_k\}$ et $f(X_k)$ l'ensemble des valeurs de la fonction correspondantes, $f(X_k) = \{f(x_1), f(x_2), \dots, f(x_k)\}$ et soit L la constante de Lipschitz de f , il est naturel de borner inférieurement f^* par une fonction linéaire par morceaux avec une pente $+L$ ou $-L$, ce qui permet d'exploiter les bornes lipschitziennes données par :

$$f(x) \geq f(x_i) - L|x - x_i|, \quad \forall x \in [a, b]. \quad (3)$$

Evidemment, pour un ensemble fixé X_k , la meilleure fonction qui borne inférieurement (sous estime) la fonction f en utilisant les informations précédentes est donnée par :

$$F_k(x) = \max_{i=1, \dots, k} \{f(x_i) - L|x - x_i|\}. \quad (4)$$

A cause de sa forme géométrique, la fonction F_k est dite *enveloppe en dents de scie* de la fonction f .

Soit l'ensemble X_k ordonné tel que

$$a \leq y_1 \leq y_2 \leq \dots \leq y_k \leq b,$$

où $\{y_1, y_2, \dots, y_k\} = \{x_1, x_2, \dots, x_k\}$.

La restriction de F_k sur l'intervalle $[y_i, y_{i+1}]$ est dite *la dent* sur $[y_i, y_{i+1}]$.

Un calcul simple montre que la dent sur $[y_i, y_{i+1}]$ atteint sa valeur minimale (pic ou sommet descendant) au point :

$$x_{p,i} = \frac{y_i + y_{i+1}}{2} + \frac{f(y_i) - f(y_{i+1})}{2L}, \quad (5)$$

avec

$$F_k(x_{p,i}) = \frac{f(y_i) + f(y_{i+1})}{2} - \frac{L(y_{i+1} - y_i)}{2}. \quad (6)$$

L'efficacité d'une méthode dépend du nombre d'évaluations de la fonction, nécessaire pour résoudre (PL_ε) , alors le nombre minimal des points d'évaluations exigé pour trouver x_ε^* a été étudié par Danilin (1971)[14]. Ceci peut être fait en construisant une enveloppe en dents de scie *référence* F_β pour résoudre le problème (PL_ε) avec un nombre minimal k_β d'évaluations de la fonction, tel que

$$F_\beta(x) = \max_{i=1, \dots, k_\beta} \{f(x_i) - L|x - x_i|\}.$$

Une telle enveloppe référence est construite avec f^* supposée connue. Elle est désignée pour donner un nombre référence des points d'évaluations nécessaires pour résoudre (PL_ε) , dans le but d'étudier l'efficacité d'autres algorithmes où f^* est inconnue.

Il est facile de voir que l'enveloppe référence F_β peut être obtenue comme suit :

Posons $F_\beta(a) = f^* - \varepsilon$. Alors, le premier point d'évaluation x_1 est le point d'intersection de la droite $(f^* - \varepsilon) + L(x - a)$ avec la courbe de $f(x)$, le prochain pic descendant (en bas) est au point :

$$x_{p,1} = x_1 + \frac{f(x_1) - (f^* - \varepsilon)}{L},$$

et il vérifie $F_\beta(x_{p,1}) = f^* - \varepsilon$.

En procédant de cette façon, on construit une enveloppe en dents de scie $F_\beta(x)$, pour laquelle la plus petite valeur du pic descendant est $f^* - \varepsilon$. Une telle enveloppe où la hauteur des dents est la même (égale à $f^* - \varepsilon$) est dite enveloppe *régulière*.

Algorithme (enveloppe référence en dents de scie)

Initialisation :

Poser $k = 1$;

x_1 est la solution de l'équation $f(x) = (f^* - \varepsilon) + L(x - a)$;

Enveloppe référence en dents de scie :

Etape $k = 1, 2, \dots$;

Poser $x_{p,k} = x_k + \frac{f(x_k) - (f^* - \varepsilon)}{L}$;

Trouver x_{k+1} solution de l'équation $f(x) = (f^* - \varepsilon) + L(x - x_{p,k})$;

Si $x_{k+1} \geq b$, alors arrêter et poser $k_\beta = k$;

Sinon, poser $k = k + 1$;

Aller à l'étape k .

Maintenant, on va donner d'autres algorithmes qui utilisent des enveloppes en dents de scie.

1.2.2 Méthode d'Evtushenko

L'algorithme d'Evtushenko (1971) a été désigné pour l'optimisation des fonctions lipschitziennes; c'est un algorithme itératif ordonné, i.e., les points d'évaluations à des itérations successives sont des valeurs croissantes de x appartenant à l'intervalle $[a, b]$.

Considérons le problème (PL_ε) . Soient x_k le dernier point d'évaluation et f_ε la meilleure valeur courante connue de la fonction f . On va essayer de trouver x_{k+1} tel que le pas de déplacement $x_{k+1} - x_k$ soit maximal sous la condition que :

Si $f(x_{k+1}) \geq f_\varepsilon$, alors on a

$$f_\varepsilon - F_k(x_{p,k}) \leq \varepsilon. \quad (7)$$

On remplace $F_k(x_{p,k})$ par sa formule donnée dans (6), on obtient :

$$\begin{aligned}
f_\varepsilon - F_k(x_{p,k}) &= f_\varepsilon - \frac{f(x_k)+f(x_{k+1})}{2} + \frac{L(x_{k+1}-x_k)}{2} \leq \varepsilon \\
\Rightarrow 2f_\varepsilon - f(x_k) - f(x_{k+1}) + L(x_{k+1} - x_k) &\leq 2\varepsilon \\
\Rightarrow L(x_{k+1} - x_k) &\leq 2\varepsilon + f(x_k) + f(x_{k+1}) - 2f_\varepsilon
\end{aligned}$$

ce qui implique

$$(x_{k+1} - x_k) \leq \frac{1}{L} (2\varepsilon + f(x_k) + f(x_{k+1}) - 2f_\varepsilon), \quad (8)$$

et à cause de la condition $f(x_{k+1}) \geq f_\varepsilon$, on aura

$$x_{k+1} = x_k + \frac{1}{L} (2\varepsilon + f(x_k) - f_\varepsilon). \quad (9)$$

Ceci est l'essentiel de la procédure d'Evtushenko (1971).

On donne maintenant l'algorithme correspondant.

Algorithme (enveloppe en dents de scie d'Evtushenko)

Initialisation :

Poser $k = 1$;

$$x_1 = a + \frac{\varepsilon}{L};$$

$$x_\varepsilon = x_1;$$

$$f_\varepsilon = f(x_\varepsilon);$$

Enveloppe en dents de scie d'Evtushenko :

Etape $k = 1, 2, \dots$:

Si $x_{k+1} > b$, alors arrêter.

Sinon, poser

$$x_{k+1} = x_k + \frac{1}{L} (2\varepsilon + f(x_k) - f_\varepsilon);$$

Si $f(x_{k+1}) < f_\varepsilon$, alors poser

$$f_\varepsilon = f(x_{k+1});$$

$$x_\varepsilon = x_{k+1};$$

Poser $k = k + 1$;

Aller à l'Etape k .

A partir de la formule (9), dans le pire des cas d'une fonction constante ou décroissante d'une façon monotone, on a $f(x_\varepsilon) = f_\varepsilon$ pour tout k , d'où $x_{k+1} - x_k = \frac{2\varepsilon}{L}$, donc on aura l'algorithme passif avec un pas de déplacement égale $\frac{2\varepsilon}{L}$.

Le nombre minimum de points d'évaluations exigé par l'algorithme d'Evtushenko est $2 + \left\lceil \log \left(1 + \frac{L(b-a)}{2\varepsilon} \right) \right\rceil$, il est atteint quand f est une fonction affine sur l'intervalle $[a, b]$ avec une pente L [14].

L'efficacité de la procédure dépend beaucoup de la position de la solution optimale x^* , et elle tend à être mauvaise quand $x^* \rightarrow b$.

L'enveloppe en dents de scie d'Evtushenko peut être considérablement différente de l'enveloppe référence, particulièrement quand x^* est loin de la borne inférieure a de l'intervalle.

L'algorithme d'Evtushenko ne demande pas un grand espace-mémoire, ni des calculs auxiliaires importants. Toutefois, si dans le cas unidimensionnel sa mise en oeuvre est assez simple, sa réalisation dans le cas multidimensionnel est très encombrante et compliquée [29], [31]. Les résultats numériques ont montré d'ailleurs que l'algorithme nécessite un nombre d'évaluations assez élevé pour une dimension $n > 3$.

1.2.3 Méthode de Piyavskii

Le premier algorithme le mieux connu et le plus étudié pour l'optimisation unidimensionnelle lipschitzienne est dû à Piyavskii (1967, 1972) [24], [25]. Il a été indépendamment redécouvert par Shubert (1972) [28]. C'est un algorithme itératif pour résoudre le problème (PL_ε) , il construit d'une façon itérative une enveloppe en dents de scie de la fonction objectif f de plus en plus raffinée sur le domaine d'optimisation $[a, b]$ et évalue f au point qui correspond au minimum de cette enveloppe en dents de scie. Ainsi, à chaque itération une basse dent est trouvée et la fonction f est évaluée au pic descendant de cette dent, qui est alors divisée en deux dents plus petites. L'algorithme s'arrête quand la

différence entre la hauteur de l'enveloppe en dents de scie et la valeur imposée ne dépasse pas ε .

En effet, cette enveloppe est construite comme suit :

Commençant par $x_1 = \frac{a+b}{2}$, et $f(x_1)$.

La première enveloppe en dents de scie

$$F_1(x) = f(x_1) - L|x - x_1|,$$

est minimisée sur $[a, b]$, dans le but d'obtenir le plus petit pic (d'en bas) au point x_2 tel que $x_2 \in \arg \min F_1([a, b])$. Alors, la fonction f est évaluée à ce point pic x_2 et la dent correspondante est divisée en d'autres dents plus petites pour obtenir une nouvelle enveloppe

$$F_2(x) = \max_{j=1,2} \{f(x_j) - L|x - x_j|\},$$

qui sera minimisée sur $[a, b]$.

Ainsi, la procédure est gouvernée par la formule

$$F_k(x) = \max_{j=1,2,\dots,k} \{f(x_j) - L|x - x_j|\}, \quad (10)$$

et

$$x_{k+1} = \arg \min_{x \in [a,b]} F_k(x). \quad (11)$$

Enfin, on donne l'algorithme de Piyavskii, pour résoudre le problème (PL_ε). On note f_ε la meilleure valeur courante de la fonction f et F_ε la valeur minimale de l'enveloppe en dents de scie courante.

Algorithme (enveloppe en dents de scie de Piyavskii)

Initialisation :

Poser $k = 1$;

$$x_1 = \frac{a+b}{2};$$

$$x_\varepsilon = x_1;$$

$$f_\varepsilon = f(x_\varepsilon);$$

$$F_\varepsilon = f_\varepsilon - \frac{L(b-a)}{2};$$

$$F_1 = f(x_1) - L|x - x_1|;$$

Enveloppe en dents de scie de Piyavskii :

Etape $k = 1, 2, \dots$:

Si $f_\varepsilon - F_\varepsilon \leq \varepsilon$, alors arrêter.

Sinon, déterminer

$$x_{k+1} \in \arg \min F_k([a, b]);$$

Si $f(x_{k+1}) \leq f_\varepsilon$, alors poser $f_\varepsilon = f(x_{k+1})$, $x_\varepsilon = x_{k+1}$;

Poser $F_{k+1}(x) = \max_{j=1, \dots, k+1} \{f(x_j) - L|x - x_j|\}$;

et $F_\varepsilon = \min F_{k+1}([a, b])$;

Poser $k = k + 1$;

Aller à l'Etape k .

Evidemment, le second et le troisième point d'évaluation dans l'algorithme de Piyavskii sont a et b , $x_2 = a$ implique $x_3 = b$ et $x_2 = b$ implique $x_3 = a$.

Notons qu'il existe plusieurs variantes de l'algorithme de Piyavskii, telle que celle proposée par Shoen (1982) qui au lieu de choisir le plus petit des points de pic d'en bas comme le prochain point d'évaluation, il prend le point d'évaluation de la stratégie passive, qui est le plus proche de ce point de pic [13]. Shen et Zhu (1987) ont donné aussi une autre version plus simplifiée, dans laquelle à chaque itération le nouveau point d'évaluation est au centre d'un sous intervalle dont les bornes sont deux points consécutifs des évaluations précédentes. Hansen et al (1989 et 1991) ont présenté une étude complète sur l'algorithme unidimensionnel de Piyavskii [13],[14].

Dans Horst et Tuy [13], il est montré que l'algorithme de Piyavskii, comme d'ailleurs d'autres algorithmes, peut être vu comme un algorithme de type Branch-and-Bound. Dans cette reformulation Branch-and-Bound, le sous-problème correspond à une dent de

l'enveloppe courante, ainsi la sélection est mieux bornée. L'action Branching est faite en divisant la dent choisie à son point de pic, et les dents non prometteuses dont on n'a pas besoin seront éliminées pour réduire l'utilisation de l'espace mémoire.

L'algorithme de Piyavskii peut prendre plus d'itérations que l'algorithme passif dans le pire des cas, i.e, pour les fonctions assez plates ou constantes. En effet, si ε est différent de $\frac{L(b-a)}{2^n}$ pour un entier n , alors certains points d'évaluations de l'algorithme de Piyavskii ne coïncident pas avec ceux de l'algorithme passif. Comme toutes les dents de l'enveloppe finale doivent avoir une hauteur qui ne dépasse pas ε , on aura besoin de plus d'itérations pour l'algorithme de Piyavskii que pour l'algorithme passif.

Notons qu'une extension directe au cas multidimensionnel a été proposée par Piyavskii (1967, 1971) [25]; en prenant le domaine d'optimisation le pavé $\prod_{i=1}^n [a_i, b_i] \subset \mathbb{R}^n$, on obtient le même algorithme que précédemment mais avec la fonction $F(x)$ remplacée par :

$$F_k(x) = \max_{j=1, \dots, k} \{f(x_j) - L \|x - x_j\|\} ; \quad (10')$$

à la k -ième itération, le point d'évaluation est déterminé sous la condition

$$x_{k+1} \in \arg \min F_k\left(\prod_{i=1}^n [a_i, b_i]\right). \quad (11')$$

Cependant, cette extension au cas multidimensionnel est numériquement inefficace car le problème (11') constitue d'énormes difficultés pour la mise en oeuvre de l'algorithme.

Pour le cas multidimensionnel, il y a des généralisations qui ont été proposées par d'autres auteurs [17], [21], [22], mais celles-ci sont toutes compliquées dans la réalisation pratique.

1.2.4 Méthode de Shoen

Comme il a été mentionné ci-dessus, le nombre de points d'évaluations d'un algorithme séquentiel coïncide, dans le pire des cas avec celui de l'algorithme passif. Shoen (1982) a proposé un algorithme proche de l'algorithme de Piyavskii ; la seule différence est qu'au

lieu de choisir le point de pic $x_{p,k}$ de la dent la plus basse comme le prochain point d'évaluation, il choisit un point de la stratégie passive le plus proche, qui est facile à trouver.

En effet, soit $x_{p,k} \in [\alpha_j, \beta_j]$; les points de la stratégie passive dans cet intervalle sont : $\alpha_j + \frac{\varepsilon}{L}, \alpha_j + \frac{3\varepsilon}{L}, \dots$

Le point x_{k+1} est parmi ces points, le plus proche de $x_{p,k}$ à gauche si

$$\left(\frac{L(x_{p,k} - \alpha_j)}{2\varepsilon} - \frac{1}{2} \right) \bmod 1 \leq \frac{1}{2},$$

et le plus proche de $x_{p,k}$ à droite sinon.

Donc,

$$x_{k+1} = \alpha_j + \frac{\varepsilon}{L} + \frac{2\varepsilon}{L} \left(\left[\frac{L(x_{p,k} - \alpha_j)}{2\varepsilon} - \frac{1}{2} \right] + \gamma \right),$$

où $\gamma = 0$ si $\left(\frac{L(x_{p,k} - \alpha_j)}{2\varepsilon} - \frac{1}{2} \right) \bmod 1 \leq \frac{1}{2}$ et $\gamma = 1$ sinon.

L'algorithme de Shoen est donné comme suit, avec $[\cdot]$ désignant la partie entière d'un nombre réel.

Algorithme (enveloppe en dents de scie de Shoen)

Initialisation :

Poser $k = 1$;

$$l = \frac{L(b-a)}{2\varepsilon} - \frac{1}{2};$$

Si $l - [l] > \frac{1}{2}$, alors poser $l = l + 1$;

$$x_1 = a + (2l + 1)\frac{\varepsilon}{L};$$

$$x_\varepsilon = x_1;$$

$$f_\varepsilon = f(x_\varepsilon);$$

$$F_\varepsilon = f_\varepsilon - L \max \{x_\varepsilon - a, b - x_\varepsilon\};$$

Enveloppe en dents de scie de Shoen :

Etape $k = 1, 2, \dots$:

Si $f_\varepsilon - F_\varepsilon \leq \varepsilon$, alors arrêter.

Sinon, déterminer

$$x_{p,k} \in \arg \min F_k([a, b]);$$

$$\alpha_j = x_{p,k} - \frac{f_\varepsilon - F_k(x_{p,k})}{L};$$

$$\beta_j = 2x_{p,k} - \alpha_j;$$

$$l = \frac{L(x_{p,k} - \alpha_j)}{2\varepsilon} - \frac{1}{2};$$

Si $l - [l] > \frac{1}{2}$, alors poser $l = l + 1$;

$$x_{k+1} = \alpha_j + (2l + 1) \frac{\varepsilon}{L};$$

Si $f(x_{k+1}) \leq f_\varepsilon$, alors poser $f_\varepsilon = f(x_{k+1})$, $x_\varepsilon = x_{k+1}$;

$$\text{Poser } F_{k+1}(x) = \max_{j=1, \dots, k+1} \{f(x_j) - L|x - x_j|\};$$

$$\text{et } F_\varepsilon = \min F_{k+1}([a, b]);$$

Poser $k = k + 1$;

Aller à l'Etape k .

Remarque.

Les algorithmes donnés pour résoudre le problème (PL_ε) supposent que la valeur de la constante de Lipschitz L est connue à priori. On voudrait mentionner que Strongin (1973, 1978) a proposé un algorithme qui, au lieu de L , utilise une estimation de L , qui est une multiplication des plus grandes valeurs absolues des pentes des lignes qui joignent les points des évaluations successives. La convergence vers un minimum global ($\varepsilon = 0$) est garantie, si cette estimation est une borne supérieure de L , suffisamment grande [14].

1.3 Méthode Branch-and-Bound

La méthode Branch-and-Bound est largement utilisée pour résoudre plusieurs classes de problèmes difficiles en optimisation. Elle a été appliquée, avec succès, à certains types de problèmes, tels que la minimisation d'une fonction concave sur un ensemble convexe, la minimisation de fonctions lipschitziennes et la minimisation de la différence de deux fonctions convexes (l'optimisation d. c.) [13]. La stratégie de cette méthode consiste à partager le domaine d'optimisation en plusieurs sous-ensembles (l'action branching),

dans lesquels on détermine les bornes inférieures, souvent aussi les bornes supérieures de la fonction objectif (l'action bounding), ensuite on détermine les sous-ensembles qui ne peuvent pas contenir un minimiseur global, on les élimine et on continue la recherche dans la partie restante du domaine d'optimisation.

Maintenant, on présente un prototype de la méthode Branch-and-Bound.

1.3.1 Un prototype de la méthode Branch-and-Bound

Soit le problème d'optimisation globale suivant :

$$(P_1) \quad \begin{cases} \min f(x) \\ x \in X \end{cases}$$

où $X \subset \mathbb{R}^n$ et $f : X \rightarrow \mathbb{R}$ la fonction objectif.

On suppose que $\min f(X)$ existe.

L'idée des méthodes B.B en optimisation globale est très simple :

- Commencer par un ensemble réalisable $M_0 \supset X$, est partager M_0 en un nombre fini de sous-ensembles $M_i, i \in I$ (faire une partition).
- Pour chaque M_i , déterminer une borne inférieure $\beta(M_i)$ et si possible une borne supérieure $\alpha(M_i)$ pour la fonction f vérifiant :

$$\beta(M_i) \leq \inf f(M_i \cap X) \leq \alpha(M_i).$$

Alors, $\beta = \min_{i \in I} \beta(M_i)$, $\alpha = \min_{i \in I} \alpha(M_i)$ sont les bornes globales, i.e., on a

$$\beta \leq \min f(X) \leq \alpha.$$

- Si $\alpha = \beta$ (ou $\alpha - \beta \leq \varepsilon$, pour un $\varepsilon > 0$ donné), alors arrêter.
- Sinon, choisir quelques sous-ensembles M_i , et partager ces sous-ensembles pour obtenir une partition plus raffinée de M_0 . Déterminer de nouvelles bornes meilleures sur les éléments de la nouvelle partition et répéter le même processus.

Un des avantages de la méthode B.B est que dans une itération on peut éliminer les sous-ensembles de X qui ne peuvent pas contenir un minimiseur global.

Un des inconvénients de cette méthode est que la précision d'une solution approchée peut être seulement mesurée par la différence $\alpha - \beta$ des bornes courantes. Donc, un "bon" point réalisable trouvé, ne peut être détecté comme "bon", qu'après plusieurs raffinements.

Définition 1.3.1

Soient B un ensemble de \mathbb{R}^n et I un ensemble fini d'indices.

Un ensemble $\{M_i, i \in I\}$ de sous-ensembles de B est dit une partition de B si

$$B = \bigcup_{i \in I} M_i \text{ et } M_i \cap M_j = \partial M_i \cap \partial M_j \quad \forall i, j \in I, i \neq j$$

où ∂M_i est la frontière de M_i .

Pour les ensembles M d'une partition, il est naturel d'utiliser les plus simples des polytopes ou des ensembles polyédriques convexes tels que les simplexes et les pavés.

Définition 1.3.2

Un ensemble M d'une partition vérifiant $M \cap X = \emptyset$ est dit non réalisable.

Un ensemble M d'une partition vérifiant $M \cap X \neq \emptyset$ est dit réalisable.

Un ensemble M d'une partition est dit incertain si on ne sait pas s'il est réalisable ou non, i.e., $M \cap X$ est non déterminé.

Maintenant, on donne un prototype de la procédure B.B.

Algorithme (Branch-and-Bound)

1) **Initialisation :**

Choisir :

$M_0 \supset X$,

$S_{M_0} \subset X$ (S_{M_0} un ensemble fini de points de M_0),

$-\infty < \beta_0 \leq \min f(X)$ (β_0 un minorant de f sur M_0);

Poser :

$$\mathcal{M}_0 = \{M_0\}, \alpha_0 = \min f(S_{M_0}), \beta(M_0) = \beta_0;$$

Si $\alpha_0 < +\infty$, alors déterminer $x_0 \in \arg \min f(S_{M_0})$ (i.e, $f(x_0) = \alpha_0$);

Si $\alpha_0 - \beta_0 \leq \varepsilon$, alors arrêter (x_0 est une solution optimale approchée);

Sinon, aller à l'étape 2);

2) Etape k=1, 2,.....

Au début de l'étape k, nous avons la partition actuelle \mathcal{M}_{k-1} d'un sous-ensemble de M_0 qui est digne d'intérêt. En plus, pour tout $M \in \mathcal{M}_{k-1}$, on a un ensemble $S_M \subseteq M \cap X$ et des bornes $\beta(M), \alpha(M)$ vérifiant :

$$\begin{cases} \beta(M) \leq \inf f(M \cap X) \leq \alpha(M) & , \text{ si } M \cap X \neq \emptyset \\ \beta(M) \leq \inf f(M) & , \text{ si } M \cap X \text{ est non déterminé} \end{cases}$$

On a aussi, les bornes $\beta_{k-1}, \alpha_{k-1}$ vérifiant :

$$\beta_{k-1} \leq \inf f(X) \leq \alpha_{k-1}.$$

Finalement, si $\alpha_{k-1} < +\infty$, alors on a le point $x_{k-1} \in X$, vérifiant $f(x_{k-1}) = \alpha_{k-1}$ (le meilleur point réalisable obtenu jusqu'à maintenant).

k.1. Eliminer tout ensemble $M \in \mathcal{M}_{k-1}$ vérifiant :

$$\beta(M) \geq \alpha_{k-1}.$$

Soit \mathcal{R}_k la collection des sous-ensembles restants de \mathcal{M}_{k-1}

k.2. Choisir une collection non vide d'ensembles $\mathcal{P}_k \subset \mathcal{R}_k$, et construire une partition de chaque élément de \mathcal{P}_k .

Soit \mathcal{P}'_k la collection de tous les éléments des nouvelles partitions.

k.3. Eliminer chaque $M \in \mathcal{P}'_k$ qui vérifie $M \cap X = \emptyset$.

Soit \mathcal{M}'_k la collection de tous les ensembles restants de \mathcal{P}'_k .

k.4. Déterminer pour chaque $M \in \mathcal{M}'_k$ un ensemble S_M et un nombre $\beta(M)$ vérifiant :

$$S_M \subseteq M \cap X,$$

$$\begin{cases} \beta(M) \leq \inf f(M \cap X) & , \text{ si } M \cap X \neq \emptyset \\ \beta(M) \leq \inf f(M) & , \text{ si } M \cap X \text{ est non déterminé} \end{cases}$$

En plus, on a

$$S_M \supseteq M \cap S_{M'} , \beta(M) \geq \beta(M') \text{ pour } M \subset M' \in \mathcal{M}_{k-1}.$$

Poser $\alpha(M) = \min f(S_M)$.

k.5. Poser $\mathcal{M}_k = (\mathcal{R}_k \setminus \mathcal{P}_k) \cup \mathcal{M}'_k$.

Calculer

$$\alpha_k = \min \{ \alpha(M) : M \in \mathcal{M}_k \},$$

$$\beta_k = \min \{ \beta(M) : M \in \mathcal{M}_k \}.$$

Si $\alpha_k < +\infty$, alors déterminer $x_k \in X$, tel que $f(x_k) = \alpha_k$.

k.6. Si $\alpha_k - \beta_k \leq \varepsilon$, alors arrêter (x_k est une solution optimale approchée).

Sinon, poser $k = k + 1$.

Aller à l'étape 2).

Remarques.

i) On dit qu'un élément M de la partition \mathcal{M}_{k-1} est exclu, si $\beta(M) \geq \alpha_{k-1}$, un tel élément sera éliminé dans l'étape k . Le critère d'arrêt $\alpha_k = \beta_k$, signifie que tous les éléments de la partition sont exclus.

ii) Pour chaque ensemble M d'une partition, $S_M \subset M \cap X$ est l'ensemble des points réalisables dans M . On suppose que $\min f(S_M)$ existe ou $S_M = \emptyset$. Dans les applications, les ensembles S_M sont souvent finis. Cependant, l'algorithme précédent reste valable

même si $S_M = \emptyset$, dans ce cas on prend $\alpha_k = +\infty$.

iii) La suite (x_k) , où $f(x_k) = \alpha_k$ converge vers une approximation de la solution optimale de la fonction objectif f , et la différence $\alpha_k - \beta_k$ mesure la proximité de la meilleure solution actuelle x_k d'un minimiseur global.

iv) La majorité des algorithmes B.B pour l'optimisation globale de fonctions lipschitziennes utilisent des partitions de l'ensemble X en hyperrectangles (pavés de \mathbb{R}^n), de la forme $I_k = [a_1^k, b_1^k] \times [a_2^k, b_2^k] \times \dots \times [a_n^k, b_n^k]$. On peut aussi utiliser d'autres partitions.

La convergence et l'efficacité de l'algorithme Branch-and-Bound dépend des trois importantes opérations de base suivantes :

- La détermination de $\alpha(M)$ et $\beta(M)$ (l'action bounding).
- Le choix de \mathcal{P}_k (la sélection).
- La construction des partitions (raffinement).

Le problème de convergence et les aspects de l'implémentation ont été étudiés sous différentes conditions [13], [14], [15],[23]. La convergence vers le minimum global est assurée, mais la réalisation pratique de la méthode est très compliquée. Ceci est dû à la nécessité du stockage d'un nombre énorme de sous-ensembles de X ainsi qu'à la difficulté liée à leur classement et au choix de l'un d'entre eux pour procéder à une nouvelle répartition.

1.3.2 Méthode Branch-and-Bound pour minimiser une fonction lipschitzienne sur un pavé de \mathbb{R}^n

Soit le problème d'optimisation globale suivant :

$$(P'_1) \quad \begin{cases} \min f(x) \\ x \in X \end{cases}$$

où l'ensemble réalisable X est un pavé de \mathbb{R}^n , i.e., $X = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbb{R}^n$ et la fonction objectif $f : X \rightarrow \mathbb{R}$ est lipschitzienne avec une constante de Lipschitz L .

Soient les deux vecteurs $a = (a_1, a_2, \dots, a_n)$ et $b = (b_1, b_2, \dots, b_n)$, alors on peut définir l'ensemble X comme suit :

$$X = \{x \in \mathbb{R}^n : a \leq x \leq b\}.$$

Soit $L' > L$ un nombre supérieur à la constante de Lipschitz optimale L .
Soit M le pavé de \mathbb{R}^n défini par

$$M = \{x \in \mathbb{R}^n : a_M \leq x \leq b_M\}.$$

Donnons, maintenant, l'algorithme B.B pour résoudre le problème (P'_1) .

Algorithme Branch-and-Bound pour résoudre (P'_1)

1) **Initialisation :**

Poser :

$$M_0 = X, \mathcal{M}_0 = \{M_0\},$$

$$\alpha_0 = \min \{f(a), f(b)\}, x_0 \in \{a, b\} \text{ tel que } \alpha_0 = f(x_0),$$

$$\beta_0 = \max \{f(a), f(b)\} - L' \|b - a\|;$$

Aller à l'étape 2)

2) **Etape k=1, 2,.....**

Au début de l'étape k, nous avons la partition actuelle (un pavé) \mathcal{M}_{k-1} d'un sous-ensemble de M_0 qui est digne d'intérêt.

En plus, pour tout $M \in \mathcal{M}_{k-1}$, on a des bornes $\beta(M), \alpha(M)$ vérifiant :

$$\beta(M) \leq \min f(M) \leq \alpha(M).$$

On a aussi, les bornes inférieure et supérieure $\beta_{k-1}, \alpha_{k-1}$ qui vérifient :

$$\beta_{k-1} \leq \min f(X) \leq \alpha_{k-1},$$

et un point x_{k-1} tel que $f(x_{k-1}) = \alpha_{k-1}$.

k.1. Eliminer tout ensemble $M \in \mathcal{M}_{k-1}$ vérifiant :

$$\beta(M) \geq \alpha_{k-1}.$$

Soit \mathcal{R}_k la collection des sous-ensembles restants de \mathcal{M}_{k-1}

k.2. Choisir une collection non vide d'ensembles $\mathcal{P}_k \subset \mathcal{R}_k$ vérifiant :

$$\mathcal{P}_k \cap \arg \min \{\beta(M) : M \in \mathcal{R}_k\} \neq \emptyset.$$

Pour chaque $\tilde{M} \in \mathcal{P}_k$ choisir

$$x_{\tilde{M}} = \frac{1}{2}(a_{\tilde{M}} + b_{\tilde{M}}) + \frac{f(a_{\tilde{M}}) + f(b_{\tilde{M}})}{2L' \|b_{\tilde{M}} - a_{\tilde{M}}\|} (b_{\tilde{M}} - a_{\tilde{M}}),$$

et partager \tilde{M} en deux sous-pavés, en utilisant l'hyperplan qui contient $x_{\tilde{M}}$ et qui est orthogonal à l'un des côtés les plus longs de \tilde{M} .

Soit \mathcal{M}'_k la collection des éléments des nouvelles partitions.

Pour chaque $M' \in \mathcal{M}'_k$, notons par $\tilde{M} \in \mathcal{P}_k$ l'hyperrectangle dont la subdivision génère M' .

k.3. Pour chaque $M' \in \mathcal{M}'_k$, poser

$$\begin{aligned} \alpha(M') &= \min \{f(a_{M'}), f(b_{M'}), f(x_{\tilde{M}'})\}, \\ \beta(M') &= \max \left\{ \beta(\tilde{M}'), \max \{f(a_{M'}), f(b_{M'}), f(x_{\tilde{M}'})\} - L' \|b_{M'} - a_{M'}\| \right\}. \end{aligned}$$

k.4. Poser $\mathcal{M}_k = (\mathcal{R}_k \setminus \mathcal{P}_k) \cup \mathcal{M}'_k$.

Calculer

$$\alpha_k = \min \{\alpha(M) : M \in \mathcal{M}_k\},$$

$$\beta_k = \min \{ \beta(M) : M \in \mathcal{M}_k \}.$$

Déterminer $x_k \in X$ tel que $f(x_k) = \alpha_k$.

k.5. Si $\alpha_k - \beta_k \leq \varepsilon$, alors arrêter, x_k est une solution optimale (approchée).

Sinon, poser $k = k + 1$.

Aller à l'étape 2).

Remarques.

i) Il est montré dans [13], qu'en utilisant l'algorithme précédent on a

$$\lim_{k \rightarrow +\infty} \beta_k = \min f(X) = \lim_{k \rightarrow +\infty} \alpha_k,$$

et chaque point d'accumulation x^* de la suite $\{x_k\}$ vérifie $f(x^*) = \min f(X)$.

ii) En pratique, un pavé M ne sera pas subdivisé si son diamètre $\|b_M - a_M\|$ est inférieur à un paramètre fixé $\delta > 0$.

iii) Si $\alpha_k - \beta_k \leq \varepsilon$ ou $\|b_M - a_M\| < \delta$ pour tous les ensembles M qui sont d'intérêt, alors l'algorithme s'arrête à un point x_k . Dans ce cas, une recherche locale partant de x_k peut nous conduire à une approximation du minimum global.

1.4 Méthode de l'indicateur de relief

Cette méthode est due essentiellement à Thach et Tuy (1990), dont l'idée consiste à introduire dans la stratégie de l'optimisation globale une procédure qui permet de dépasser les optima locaux. Ceci a permis de résoudre certaines classes de problèmes spéciaux.

On considère le problème suivant :

$$(Pc) \begin{cases} \min f(x) \\ x \in D \end{cases}$$

où $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ est une fonction continue, et D un compact de \mathbb{R}^n .

De plus, on suppose que

$$\inf \{f(x) / x \in D\} = \inf \{f(x) / x \in \text{int}D\} \quad (12)$$

Cette supposition est satisfaite, par exemple, si D est robuste.

Rappelons qu'un sous ensemble fermé $D \subset \mathbb{R}^n$ est dit robuste s'il est la clôture d'un ensemble ouvert, i.e, $D = \overline{O}$, où O est un ensemble ouvert de \mathbb{R}^n .

Définition 1.4.1

Soit $C \subset \mathbb{R}^n$ un ensemble convexe. Une fonction $h : C \longrightarrow \mathbb{R}$ est dite une fonction d. c. sur C , s'il existe deux fonctions convexes : $p : C \longrightarrow \mathbb{R}$, et $q : C \longrightarrow \mathbb{R}$, telles que

$$\forall x \in C, h(x) = p(x) - q(x)$$

Dans cette définition d. c. est l'abréviation de 'différence de deux fonctions convexes'.

L'idée dans le développement qui suit est d'associer à f et D et à chaque $\alpha \in \mathbb{R}$, une fonction d. c. $\varphi_\alpha(x)$ telle que \bar{x} est un minimiseur global de f sur D si, et seulement, si :

$$O = \min \{\varphi_{\bar{\alpha}}(x) / x \in \mathbb{R}^n\}$$

où $\bar{\alpha} = f(\bar{x})$.

Basée sur ce critère d'optimalité, une méthode va être obtenue pour résoudre le problème (P_c) dans le sens mentionné ci-dessus.

1.4.1 Séparateurs pour la fonction f sur l'ensemble D

Une fonction $r(\alpha, x)$ est définie comme séparateur pour f sur l'ensemble D dans le sens suivant :

Soit $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$, et pour tout $\alpha \in \overline{\mathbb{R}}$, on considère les ensembles de niveau suivants :

$$D_\alpha = \{x \in D / f(x) < \alpha\},$$

$$\overline{D}_\alpha = \{x \in D / f(x) \leq \alpha\}.$$

En plus, soit

$$d_A(x) = \inf \{\|x - y\| / y \in A\},$$

la distance entre $x \in \mathbb{R}^n$ et l'ensemble A de \mathbb{R}^n .

Définition 1.4.2

Une fonction $r(\alpha, x)$ à valeur réelle, définie sur $\overline{\mathbb{R}} \times \mathbb{R}^n$ est dite séparateur pour la fonction f sur l'ensemble D si elle satisfait les conditions suivantes :

- i) $0 \leq r(\alpha, x) \leq d_{\overline{D}_\alpha}(x), \forall \alpha \in \overline{\mathbb{R}}, \forall x \in \mathbb{R}^n$.
- ii) Pour chaque $\alpha \in \mathbb{R}$ fixé, $x_k \xrightarrow[k \rightarrow +\infty]{} \tilde{x} \notin \overline{D}_\alpha$ implique que $\lim_{k \rightarrow +\infty} r(\alpha, x) > 0$.
- iii) $r(\alpha, x)$ n'est pas une fonction croissante de α , i.e., $\alpha \geq \alpha'$ implique que $r(\alpha, x) \leq r(\alpha', x), \forall x \in \mathbb{R}^n$.

Exemple 1:

La fonction distance $d_{\overline{D}_\alpha}(x)$ est un séparateur pour la fonction f sur l'ensemble D .

Exemple 2 :

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction deux fois continûment différentiables dont la dérivée seconde est bornée, i.e., il existe une constante $M > 0$ tel que

$$|f''(x)| \leq M, \forall x \in \mathbb{R}.$$

Pour tout $\alpha \in \overline{\mathbb{R}}$, et $x \in \mathbb{R}$, posons

$$\rho(\alpha, x) = \begin{cases} 0 & , \text{ si } f(x) \leq \alpha \\ -\frac{1}{M} \left[|f'(x)| + (|f'(x)|^2 + 2M(f(x) - \alpha))^{\frac{1}{2}} \right] & , \text{ si } f(x) > \alpha \end{cases} \quad (13)$$

Alors, un séparateur pour f sur l'ensemble $D = \{x \in \mathbb{R} / |x| \leq c\}$, est donné par :

$$r(\alpha, x) = \max \{\rho(\alpha, x), |x| - c\}. \quad (14)$$

Notons que la seconde expression dans (14) décrit l'unique solution positive $t = \rho(\alpha, x)$ de l'équation

$$\frac{M}{2}t^2 + |f'(x)|t = f(x) - \alpha.$$

Les conditions *ii*) et *iii*) dans la Définition 1.4.2 sont vérifiées par (14).

Pour démontrer *i*) il suffit de montrer que :

$$|y| \leq c \text{ et } f(y) \leq \alpha \text{ implique que } r(\alpha, x) \leq |x - y|.$$

Mais, puisque $|y| \leq c$ implique que $|x| - c \leq |x| - |y| \leq |x - y|$, il suffit, seulement, de montrer que $f(y) \leq \alpha$ implique que $r(\alpha, x) \leq |x - y|$.

Ceci peut être fait en utilisant la formule de Taylor

$$|f(y) - f(x) - f'(x)(y - x)| \leq \frac{M}{2} |y - x|^2.$$

D'où on aura

$$|f(y) - f(x)| \leq \frac{M}{2} |y - x|^2 + |f'(x)| |y - x|,$$

et donc

$$\frac{M}{2} |y - x|^2 + |f'(x)| |y - x| \geq f(x) - f(y). \quad (15)$$

Maintenant, soit $f(y) \leq \alpha$. Alors d'après (15) on a

$$\frac{M}{2} |y - x|^2 + |f'(x)| |y - x| \geq f(x) - \alpha.$$

Mais, $\frac{M}{2}t^2 + |f'(x)|t$ est croissante pour $t > 0$ (pour x fixé). Par conséquent, et d'après la définition de $\rho(\alpha, x)$ on a $r(\alpha, x) \leq |x - y|$.

Notons que par les mêmes arguments, on peut faire l'extension de l'Exemple 2 au cas où f est une fonction définie sur \mathbb{R}^n , avec le Hessien H borné, tels que $f'(x) = \nabla f(x)$ et $f''(x) = \nabla^2 f(x) = H(x)$.

Exemple 3 :

Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et soit D un compact de \mathbb{R}^n tel que

$$D = \{x \in \mathbb{R}^n / g_i(x) \leq 0, i = 1, 2, \dots, m\},$$

où $g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ pour $i = 1, 2, \dots, m$.

Si toutes les fonctions f, g_1, g_2, \dots, g_m sont lipschitziennes avec les constantes de Lipschitz, respectivement, L, L_1, L_2, \dots, L_m , alors un séparateur pour f sur l'ensemble D , est donné par :

$$r(\alpha, x) = \max \left\{ 0, \frac{f(x) - \alpha}{L}, \frac{g_i(x)}{L_i}, i = 1, \dots, m \right\}.$$

1.4.2 Un critère d'optimalité globale

Supposons que $r(\alpha, x)$ est un séparateur pour la fonction f sur l'ensemble D , et pour tout $\alpha \in \overline{\mathbb{R}}$ on définit la fonction

$$h(\alpha, x) = \sup_{v \notin D_\alpha} \{2vx + r^2(\alpha, v) - \|v\|^2\}. \quad (16)$$

Il est clair que pour α fixé, la fonction $h(\alpha, x)$ est convexe.

Maintenant, on considère la fonction d. c. suivante :

$$\varphi_\alpha(x) = h(\alpha, x) - \|x\|^2 \quad (17)$$

Lemme 1.4.1

On a

$$\begin{cases} \varphi_\alpha(x) > 0 & , \text{ si } x \notin \overline{D}_\alpha & (18) \\ \varphi_\alpha(x) = -\inf \{ \|x - v\|^2 / v \notin D_\alpha \} & , \text{ si } x \in \overline{D}_\alpha & (19) \end{cases}$$

Démonstration

Si $x \notin \overline{D}_\alpha$, alors, d'après (ii) dans la Définition 1.4.2 on a $r(\alpha, x) > 0$, et ainsi en utilisant (16) on a

$$\begin{aligned}\varphi_\alpha(x) &= \sup_{v \notin D_\alpha} \{2vx + r^2(\alpha, v) - \|v\|^2\} - \|x\|^2 \\ &\geq 2xx + r^2(\alpha, x) - 2\|x\|^2 = r^2(\alpha, x) > 0\end{aligned}$$

Ce qui démontre (18).

Pour démontrer (19), on remarque que d'après (16) on a

$$\begin{aligned}\varphi_\alpha(x) &= \sup_{v \notin D_\alpha} \{2vx + r^2(\alpha, v) - \|v\|^2\} - \|x\|^2 \\ &= \sup_{v \notin D_\alpha} \{r^2(\alpha, v) - \|x - v\|^2\} \\ &\geq \sup_{v \notin D_\alpha} \{-\|x - v\|^2\} = -\inf_{v \notin D_\alpha} \{\|x - v\|^2\}.\end{aligned}\tag{20}$$

Maintenant, on considère un point arbitraire $x \in \overline{D}_\alpha$.

Si $f(x) = \alpha$, alors on a

$$\varphi_\alpha(x) \geq -\inf_{v \notin D_\alpha} \|x - v\|^2 \geq -\|x - x\|^2 = 0.\tag{21}$$

Ce qui implique que $-\inf_{v \notin D_\alpha} \|x - v\|^2 = 0$.

D'autre part, d'après (i) dans la Définition 1.4.2 on a

$$r^2(\alpha, v) \leq d_{D_\alpha}^2(v) \leq \|x - v\|^2, \quad \forall v \in \mathbb{R}^n.$$

Donc

$$\varphi_\alpha(x) = \sup_{v \notin D_\alpha} \{-\|x - v\|^2 + r^2(\alpha, v)\} \leq 0,$$

et ainsi, d'après (21) on a

$$\varphi_\alpha(x) = 0.$$

Alors, la relation (19) est satisfaite pour $x \in \overline{D}_\alpha$ vérifiant $f(x) = \alpha$.

Si $f(x) < \alpha$, alors à chaque point $v \notin D_\alpha$ on associe un point $z(v)$ de l'intersection du segment $[x, v]$ avec la frontière ∂D_α . Un tel point $z(v)$ existe parce que $x \in D_\alpha$ et $v \notin D_\alpha$.

Puisque $z(v) \in [x, v]$, alors $z(v) \in \overline{D}_\alpha$ et d'après (i) dans la Définition 1.4.2 on a

$$\|x - v\| = \|x - z(v)\| + \|v - z(v)\| \geq \|x - z(v)\| + r(\alpha, v).$$

Alors, on a

$$-\|x - z(v)\|^2 \geq r^2(\alpha, v) - \|x - v\|^2,$$

donc,

$$\begin{aligned} -\inf_{v \notin D_\alpha} \{\|x - v\|^2\} &= \sup_{v \notin D_\alpha} \{-\|x - v\|^2\} \\ &= \sup \{-\|x - v\|^2 : v \in \partial D_\alpha\} \\ &\geq \sup_{v \notin D_\alpha} \{r^2(\alpha, v) - \|x - v\|^2\} = \varphi_\alpha(x). \end{aligned} \quad (22)$$

Enfin, d'après (20) et (22) on a la relation (19) satisfaite pour $x \in D_\alpha$. ■

Corollaire 1.4.2

Pour tout $\alpha \in \overline{\mathbb{R}}$, vérifiant $\overline{D}_\alpha \neq \emptyset$ on a

$$\inf_{x \in \mathbb{R}^n} \varphi_\alpha(x) \leq 0.$$

De plus, pour tout $x \in D$ et $\alpha = f(x)$ on a

$$\varphi_\alpha(x) = 0.$$

Démonstration

Ce corollaire est une conséquence immédiate du Lemme 1.4.1. ■

Théorème 1.4.3

Soit $\tilde{x} \in D$ un point réalisable du problème (Pc) , et soit $\tilde{\alpha} = f(\tilde{x})$.

Considérons la fonction $\varphi_{\tilde{\alpha}}(x)$ définie dans (16), (17).

i) Si

$$\inf_{x \in \mathbb{R}^n} \varphi_{\tilde{\alpha}}(x) \leq 0 \quad (23)$$

alors, pour tout x vérifiant $\varphi_{\tilde{\alpha}}(x) < 0$ on a $x \in D$, $f(x) < \tilde{\alpha}$ (i.e., x est un point réalisable meilleur que \tilde{x})

ii) Si \tilde{x} est une solution optimale globale du problème (Pc) , alors

$$\min_{x \in \mathbb{R}^n} \varphi_{\tilde{\alpha}}(x) = 0 \quad (24)$$

iii) Si le problème (Pc) est régulier dans le sens que

$$\inf \{f(x) / x \in D\} = \inf \{f(x) / x \in \text{int}D\}$$

alors tout \tilde{x} vérifiant (24) est une solution optimale globale du problème (Pc) .

Démonstration

D'après le Lemme 1.4.1, tout $x \in \mathbb{R}^n$ vérifiant $\varphi_{\tilde{\alpha}}(x) < 0$ doit être dans $D_{\tilde{\alpha}}$. Cela démontre *i)* par la définition de $D_{\tilde{\alpha}}$.

En utilisant la supposition *i)* qu'on vient de prouver et le Corollaire 1.4.2, on voit que la condition (24) est nécessaire pour l'optimalité globale de \tilde{x} , i.e., on a *ii)*.

Dans le but de démontrer *iii)*, on suppose que \tilde{x} satisfait (24) mais il n'est pas solution optimale du problème (Pc) . Alors, en utilisant la condition de régularité (12) on voit qu'il existe un point $x' \in \text{int}D$ vérifiant $f(x') < \tilde{\alpha}$. Puisque la fonction f est continue, alors $x' \in \text{int}D_{\tilde{\alpha}}$. Ce qui implique, d'après le Lemme 1.4.1, que

$$\inf_{x \in \mathbb{R}^n} \varphi_{\tilde{\alpha}}(x) \leq \varphi_{\tilde{\alpha}}(x') = \inf_{v \notin D_{\tilde{\alpha}}} \|x' - v\| < 0$$

autrement dit, \tilde{x} ne satisfait pas (24), ce qui est contradictoire. ■

Une version légèrement modifiée du résultat précédent nous donne le corollaire suivant :

Corollaire 1.4.4

Si $\tilde{\alpha} = \min f(D)$ est la valeur optimale de la fonction objectif du problème (Pc), alors $\tilde{\alpha}$ satisfait (24), et toute solution optimale de (Pc) est un minimiseur global de $\varphi_{\tilde{\alpha}}(x)$ sur \mathbb{R}^n .

Réciproquement, si la condition de régularité (12) est vérifiée et si $\tilde{\alpha}$ satisfait (24), alors $\tilde{\alpha}$ est la valeur optimale de la fonction objectif du problème (Pc), et tout minimiseur global de $\varphi_{\tilde{\alpha}}(x)$ sur \mathbb{R}^n est une solution optimale de (Pc).

Pour la démonstration voir [13].

1.4.3 La méthode de l'indicateur de relief

Les propriétés de la fonction $\varphi_{\alpha}(x)$ présentée dans la section précédente suggèrent une interprétation de $\varphi(x)$ comme une sorte de gradient généralisé ou indicateur de relief. Selon Thach et Tuy (1990), la fonction $\varphi_{\alpha}(x)$ sera dite une *fonction indicateur de relief* pour f sur D .

On a vu que sous la condition de régularité (12), on a

$$\min_{x \in \mathbb{R}^n} \varphi_{\alpha}(x) = 0, \text{ si et seulement si, } \alpha = \min f(D)$$

Ce qui fait penser à remplacer le problème (Pc) par le problème d. c. de minimisation non contraint paramétrique suivant :

$$\text{Trouver } \alpha \text{ tel que } 0 = \inf_{x \in \mathbb{R}^n} \varphi_{\alpha}(x). \quad (25)$$

Supposons que $D \neq \emptyset$. Une méthode itérative simple pour résoudre le problème (25) est donnée comme suit :

Commencer par un point réalisable arbitraire x_1 .

A l'itération k , résoudre le problème auxiliaire suivant :

$$(P_k) \quad \text{minimiser } \varphi_{\alpha_k}(x) \text{ sur } \mathbb{R}^n, \quad (26)$$

où $\alpha_k = f(x_k)$.

Soit x_{k+1} la solution optimale de (P_k) .

Si $\varphi_{\alpha_k}(x_{k+1}) = 0$, alors arrêter, x_k est une solution optimale du problème (Pc) .

Sinon aller à l'itération $k + 1$ avec $\alpha_{k+1} = f(x_{k+1})$.

Remarque

Il est montré dans [13] que si le problème (Pc) est régulier dans le sens de (12), et si la procédure itérative ci-dessus est infinie, alors tout point d'accumulation \bar{x} de la suite $\{x_k\}$ est une solution optimale du problème (Pc) .

L'exécution de la méthode itérative, décrite ci-dessus, exige qu'on examine deux issues. La première concerne la solution du problème (P_k) . Ces problèmes, en général, ne peuvent pas être résolus par une procédure finie. Alors, on doit remplacer le problème (P_k) par une approximation convenable (Q_k) qu'on peut résoudre par un algorithme fini.

La seconde chose est le calcul du point réalisable initial x_1 . En plus, la définition implicite de la fonction $r(\alpha, x)$ donnée par (16), (17) nécessite des approximations convenables.

Une possibilité pour le traitement de ces issues est donnée comme suit :

On remplace le problème (P_k) par un problème de la forme

$$(Q_k) \quad \begin{cases} \min (h_k(x) - \|x\|^2) \\ x \in S \end{cases} \quad (27)$$

où S est un polytope convenable, et $h_k(x)$ une fonction convexe polyédrique convenable qui sous-estime $h(\alpha_k, x)$. Puisque $\min \{\varphi_{\alpha_k}(x) / x \in \mathbb{R}^n\}$ est atteint dans D , alors il suffit de choisir n'importe quel polytope S qui contient D . De plus, en s'inspirant de

la forme (16) de $h(\alpha, x)$, on considère

$$h_k(x) = \sup_{i=1,2,\dots,k} \{2x_i x + r^2(\alpha_i, x_i) - \|x\|^2\}, \quad (28)$$

où α_i est la plus petite valeur de $f(x)$ en tous les points réalisables évalués jusqu'à l'itération i , et x_{i+1} ($i \geq 1$) est une solution optimale du problème (Q_i) .

Par définition de α_i dans (28), on doit avoir $\alpha_i \leq f(x_i)$ quand $x_i \in D$. Alors, $x_i \notin D_{\alpha_k}$ pour $i = 1, \dots, k$. Puisque $r(\alpha_i, x) \leq r(\alpha_k, x)$ pour tout $x \in \mathbb{R}^n$, $i = 1, \dots, k$ (Définition 1.4.2, (iii)), on a d'après (16) et (28) pour tout x

$$h_k(x) \leq \sup_{v \notin D_{\alpha_k}} \{2vx + r^2(\alpha_k, v) - \|v\|^2\} = h(\alpha_k, x),$$

i.e., les fonctions $h_k(x)$ définies par (28) sous-estiment les fonctions $h(\alpha_k, x)$.

De plus, les fonctions $h_k(x)$ sont des fonctions convexes polyédriques puisque

$$h_k(x) = \max \{ \ell_i(x) \ / \ i = 1, \dots, k \},$$

avec

$$\ell_i(x) = 2x_i x + r^2(\alpha_i, x_i) - \|x_i\|^2. \quad (29)$$

Alors, le problème (Q_k) est équivalent à

$$(\bar{Q}_k) \quad \begin{cases} \min (t - \|x\|^2) \\ x \in S \\ \ell_i(x) \leq t, \ (i = 1, \dots, k) \end{cases} \quad (30)$$

En fait, dans ce cas, nous avons remplacé le problème d. c. non contraint (P_k) par le problème de minimisation concave (\bar{Q}_k) avec des contraintes linéaires et une fonction objectif quadratique.

Plusieurs algorithmes finis pour résoudre le problème (\bar{Q}_k) sont donnés dans [13].

Puisque $h_k(x) \leq h(\alpha_k, x)$ pour tout $x \in S$, on a, d'après le Lemme 1.4.1, la valeur

optimale de la fonction objectif de (\bar{Q}_k) n'est pas positive. De plus, si cette valeur est nulle, alors on doit avoir

$$0 = \min_{x \in \mathbb{R}^n} \varphi_{\alpha_k}(x)$$

et d'après le Théorème 1.4.3, on a $\alpha_k = \min f(D)$ et chaque point $\bar{x}_k \in D$ vérifiant $f(x_k) = \alpha_k$, résout le problème (P_c) .

Cependant, si $h_k(x_{k+1}) - \|x_{k+1}\|^2 < 0$, alors il n'est pas garanti que $f(x_{k+1}) < \alpha_k$.

Donc, on pose $\alpha_{k+1} = \alpha_k$ si $x_{k+1} \notin D$. Quand la solution x_{k+1} de (\bar{Q}_k) est réalisable, alors on peut faire appel à une procédure d'optimisation locale qui va prendre x_{k+1} comme point de départ et arrive à un point réalisable \bar{x}_{k+1} vérifiant $f(\bar{x}_{k+1}) \leq f(x_{k+1})$. Dans ce cas, on pose $\alpha_{k+1} = \min \{\alpha_k, f(\bar{x}_{k+1})\}$.

Maintenant, on donne un algorithme pour cette approximation de la méthode de l'indicateur de relief.

Algorithme de l'indicateur de relief

1) Initialisation :

Construire un polytope $S \supset D$ et choisir $x_1 \in S$.

Si aucun point réalisable n'est connu, poser $\alpha_0 = \infty$,

Si des points réalisables sont connus, poser α_0 égale à la valeur minimale de f en ces points connus.

2) Itération $k = 1, 2, \dots$:

k. 1. Si $x_k \in D$, alors, en utilisant une procédure d'optimisation locale qui débute par x_k , trouver un point $\bar{x}_k \in D$ vérifiant $f(\bar{x}_k) < f(x_k)$.

Poser $\alpha_k = \min \{\alpha_{k-1}, f(\bar{x}_k)\}$.

Si $x_k \notin D$, alors poser $\alpha_k = \alpha_{k-1}$.

Notons \tilde{x}_k le meilleur point réalisable connu, i.e., on a $f(\tilde{x}_k) = \alpha_k$.

k. 2. Poser

$$\ell_k(x) = 2x_k x + r^2(\alpha_k, x_k) - \|x_k\|^2,$$

et résoudre le problème

$$(\bar{Q}_k) \quad \begin{cases} \min (t - \|x\|^2) \\ x \in S \\ \ell_i(x) \leq t, \quad (i = 1, \dots, k) \end{cases}$$

Soit (x_{k+1}, t_{k+1}) une solution optimale de (\bar{Q}_k) .

Si

$$t_{k+1} - \|x_{k+1}\|^2 = 0,$$

alors, arrêter, \tilde{x}_k est une solution optimale de (Pc) , et $\alpha_k = f(\tilde{x}_k) = \min f(D)$.

Sinon ($t_{k+1} - \|x_{k+1}\|^2 < 0$), poser $k = k + 1$;

Aller à l'étape 2)

Proposition 1.4.5

Soit le problème (Pc) régulier dans le sens de (12), et soit l'ensemble réalisable D non vide. Supposons que $r(\alpha, x)$ est un séparateur pour la fonction f sur D et qu'il est semi-continu inférieurement dans $\mathbb{R} \times \mathbb{R}^n$. Alors, l'algorithme précédent termine, après un nombre fini d'itérations, avec une solution optimale \tilde{x}_k du problème (Pc) , ou bien, il génère une suite infinie $\{x_k\}$ et chaque point d'accumulation \bar{x} de cette suite est une solution optimale du problème (Pc) .

Pour la démonstration on peut voir [13].

1.5 Conclusion

Les problèmes d'optimisation globale de la forme (P) sont très difficiles à résoudre.

Le cas de l'optimisation lipschitzienne unidimensionnelle sans contraintes a été large-

ment étudié à travers plusieurs méthodes, parmi les méthodes les plus connues et les plus commentées dans la littérature on a celles qui sont basées sur la construction d'enveloppe en dents de scie tels que l'algorithme d'Evtushenko, l'algorithme de Piyavskii, celui de Shoen etc.

Si beaucoup de travail a été dévoué pour l'optimisation lipschitzienne multidimensionnelle sans contraintes, l'analyse des algorithmes reste beaucoup moins développée que dans le cas unidimensionnel. En effet, dans la théorie on peut définir un algorithme, le meilleur possible, mais trouver une suite de points d'évaluations d'un tel algorithme apparaît très difficile même dans le cas de deux variables.

Chapitre 2

LA METHODE ALIENOR COUPLEE AVEC L'ALGORITHME DE BRENT

2.1 Introduction

Considérons le problème d'optimisation globale multidimensionnel :

$$(P) \quad \min_{(x_1, x_2, \dots, x_n) \in X} f(x_1, x_2, \dots, x_n)$$

où f est une fonction continue sur le domaine $X = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbb{R}^n$

Le couplage de la méthode Aliénor avec certaines méthodes unidimensionnelles (méthode de Shubert) s'est avéré très efficace pour trouver les minima globaux des fonctions à plusieurs variables ayant un minimum global difficile à trouver par les méthodes classiques [37]. Le couplage consiste à approcher la fonction objectif f à plusieurs variables par une fonction à une seule variable f^* , puis chercher le minimum global de cette dernière. Dans ce qui suit, nous étudions le couplage de la méthode de la transformation réductrice Aliénor avec la méthode unidimensionnelle de Brent, qui s'applique aux fonc-

tions deux fois continûment différentiables.

2.2 La méthode Aliénor de base

La méthode de la transformation réductrice Aliénor a été élaborée par Y. Cherruault et A. Guillez (dans les années 80) [8]. Elle a été essentiellement développée pour l'optimisation globale, mais elle s'est révélée être une approche universelle, car elle peut être utilisée pour approximer des fonctions à plusieurs variables, pour résoudre des systèmes d'équations fonctionnelles et autres problèmes multidimensionnels [1], [6], [35].

L'idée de la méthode consiste à effectuer, à l'aide des courbes α -denses, une transformation qui permet de ramener le problème multidimensionnel à un problème unidimensionnel, pour pouvoir utiliser les méthodes unidimensionnelles connues pour leur efficacité et performance.

Définition 2.2.1

Soient un nombre $\alpha > 0$ et un compact $K \subset \mathbb{R}^n$. Une courbe ζ est dite α -dense dans K si $\zeta \subset K$ et $\forall x \in K, \exists a \in \zeta$ tel que $d(x, a) \leq \alpha$ (avec d la distance euclidienne dans \mathbb{R}^n).

Brièvement, la méthode Aliénor se résume ainsi :

Pour résoudre le problème (P) , on construit une courbe paramétrée :

$$h(\theta) = (h_1(\theta), h_2(\theta), \dots, h_n(\theta))$$

h est α -dense dans le pavé $\prod_{i=1}^n [a_i, b_i]$ pour $\theta \in [0, \theta_{\max}]$, où θ_{\max} est la borne supérieure du domaine de définition de h quand elle α -densifie le pavé.

Le problème de minimisation (P) est alors approximé par le problème unidimensionnel

$$(P') \quad \min_{\theta} f^*(\theta)$$

où

$$f^*(\theta) = f(h_1(\theta), h_2(\theta), \dots, h_n(\theta)) \text{ et } \theta \in [0, \theta_{\max}].$$

Dans la méthode de base, le problème unidimensionnel (P') est résolu en discrétisant l'intervalle $[0, \theta_{\max}]$ via un pas choisi $\Delta\theta$, puis on cherche le minimum de l'ensemble fini $\{f^*(\theta_k) / k = 0, 1, \dots, N\}$ où $\theta_0, \theta_1, \dots, \theta_N$ sont les points de la discrétisation [8]. Bien évidemment, le paramètre de densification α et le pas $\Delta\theta$ sont choisis de façon que le minimum global soit obtenu avec la précision désirée ε .

2.2.1 La première transformation réductrice

Dans les travaux de Y. Cherruault et A. Guillez [8] où la méthode a été décrite originellement, la fonction objectif $f(x_1, \dots, x_n)$ est supposée continue (non nécessairement lipschitzienne) sur \mathbb{R}^n et ayant un minimum global atteint à une distance finie.

Alors, pour $n = 2$, on a introduit dans le plan euclidien la spirale d'Archimède ; définie en coordonnées polaires par l'équation :

$$r = a\theta \quad \text{où } \theta \geq 0 \text{ et } a > 0 ;$$

ou par sa représentation paramétrique, après passage aux coordonnées cartésiennes :

$$h(\theta) = (h_1(\theta), h_2(\theta)) = (a\theta \cos \theta, a\theta \sin \theta), \text{ avec } \theta \geq 0.$$

Le paramètre a est destiné à tendre vers 0 pour permettre d'approcher tout point du plan euclidien par un point de la courbe. D'ailleurs, on peut montrer facilement que la courbe obtenue est α -dense dans \mathbb{R}^2 avec $\alpha = \pi a$.

Pour définir la transformation réductrice dans \mathbb{R}^n il suffit d'itérer le processus précédent.

Par exemple dans le cas de trois variables x_1, x_2, x_3 , on pose :

$$x_1 = a\theta_1 \cos \theta_1 \quad \text{et} \quad x_2 = a\theta_1 \sin \theta_1,$$

puis on lie θ_1 et x_3 par un nouveau paramètre θ en posant :

$$\theta_1 = a\theta \cos \theta \quad \text{et} \quad x_3 = a\theta \sin \theta ;$$

ainsi, on obtient la courbe paramétrée $h(\theta) = (h_1(\theta), h_2(\theta), h_3(\theta))$ définie par :

$$\begin{cases} h_1(\theta) = a^2\theta \cos \theta \cos(a\theta \cos \theta) \\ h_2(\theta) = a^2\theta \cos \theta \sin(a\theta \cos \theta) \\ h_3(\theta) = a\theta \sin \theta. \end{cases}$$

Dans le cas de quatre variables, on pose :

$$\begin{cases} x_1 = a\theta_1 \cos \theta_1 \\ x_2 = a\theta_1 \sin \theta_1 \end{cases} \quad \text{et} \quad \begin{cases} x_3 = a\theta_2 \cos \theta_2 \\ x_4 = a\theta_2 \sin \theta_2 \end{cases}$$

puis on lie θ_1 et θ_2 par un nouveau paramètre θ en posant :

$$\theta_1 = a\theta \cos \theta \quad \text{et} \quad \theta_2 = a\theta \sin \theta ;$$

on obtient la courbe paramétrée $(h_1(\theta), h_2(\theta), h_3(\theta), h_4(\theta))$ définie par :

$$\begin{cases} h_1(\theta) = a^2\theta \cos \theta \cos(a\theta \cos \theta) \\ h_2(\theta) = a^2\theta \cos \theta \sin(a\theta \cos \theta) \\ h_3(\theta) = a^2\theta \sin \theta \cos(a\theta \sin \theta) \\ h_4(\theta) = a^2\theta \sin \theta \sin(a\theta \sin \theta). \end{cases}$$

En considérant un borné X de \mathbb{R}^n , on peut prouver le résultat suivant :
quelque soit $\alpha > 0$, il existe un $a > 0$ pour lequel on a :

$$\forall x \in X, \exists \theta \geq 0 \text{ tel que } d(x, h(\theta)) \leq \alpha.$$

La fonction initiale $f(x_1, \dots, x_n)$ est alors approchée par la fonction d'une seule variable $f(h_1(\theta), \dots, h_n(\theta))$, où $x_i = h_i(\theta)$ pour $i = 1, \dots, n$, sont les fonctions définies itérativement par la spirale d'Archimède [8].

La difficulté avec cette méthode de **base** est que les fonctions composantes $h_i(\theta)$ ne s'obtiennent pas en une seule étape. La procédure de leur obtention est schématiquement une "structure d'arbre" à plusieurs paliers. A chaque palier on fait appel à la spirale d'Archimède pour exprimer les variables deux à deux en fonction d'une seule, jusqu'à l'obtention des expressions finales.

On peut voir que lorsque la dimension de l'espace \mathbb{R}^n augmente, les fonctions $h_i(\theta)$ prennent des expressions de plus en plus sophistiquées ; ce qui accroît le temps d'évaluation de la fonction $f^*(\theta)$ lors de la résolution du problème de minimisation unidimensionnelle. Notons que ce dernier problème est résolu en discrétisant l'intervalle $[0, \theta_{\max}]$ via un pas choisi $\Delta\theta$, puis on cherche le minimum de l'ensemble fini $\{f^*(\theta_k), k = 0, 1, \dots, N\}$, où $\theta_0, \theta_1, \dots, \theta_N$ sont les points de la discrétisation.

2.2.2 Complexité de la méthode Aliénor

Lorsque la fonction objectif f est continue et non nécessairement lipschitzienne, le paramètre de densification α et le pas $\Delta\theta$ sont choisis de façon heuristique; mais si f est lipschitzienne, α et $\Delta\theta$ peuvent être déterminés théoriquement tels que le minimum global soit obtenu avec la précision désirée $\varepsilon > 0$.

En effet, supposons que f soit une fonction lipschitzienne de constante l_1 ; désignons par l_2 la constante de Lipschitz de la fonction $h(\theta) = (h_1(\theta), \dots, h_n(\theta))$ qui représente la courbe α -dense dans $\prod_{i=1}^n [a_i, b_i]$. Il est évident que la fonction $f^*(\theta) = f(h(\theta))$ est lipschitzienne de constante $l_1 l_2$ sur l'intervalle $[0, \theta_{\max}]$.

D'autre part, désignons par m, m^* respectivement les minima absolus de f, f^* ; et par f_ε^* le minimum absolu du problème (P) obtenu par Aliénor. On doit donc avoir

$$f_\varepsilon^* - m \leq \varepsilon.$$

Puisque f est continue sur X , il existe un point $x_0 \in X$ tel que $m = f(x_0)$.

Par ailleurs, il existe $\theta_0 \in [0, \theta_{\max}]$ tel que $\|x_0 - h(\theta_0)\| \leq \alpha$, donc $\|f(x_0) - f(h(\theta_0))\| \leq l_1 \alpha$.

Il en résulte que

$$f(h(\theta_0)) - m \leq l_1 \alpha.$$

Et puisque $m \leq m^* \leq f(h(\theta_0))$, on en déduit

$$m^* - m \leq l_1 \alpha.$$

D'autre part on a

$$f_\varepsilon^* - m^* \leq l_1 l_2 \frac{\Delta\theta}{2},$$

ce qui entraîne

$$f_\varepsilon^* - m \leq l_1 \alpha + l_1 l_2 \frac{\Delta\theta}{2}.$$

Donc pour obtenir la précision désirée, on doit choisir

$$\alpha = \frac{\varepsilon}{2l_1}.$$

En posant

$$\varepsilon = l_1\alpha + l_1l_2\frac{\Delta\theta}{2},$$

on obtient

$$\Delta\theta = \frac{\varepsilon}{l_1l_2}.$$

Le temps de calcul pour la recherche des minima globaux dans cette méthode de base est donné par

$$T = \frac{\theta_{\max}}{\Delta\theta}T_0;$$

où T_0 est le temps moyen pour calculer $f^*(\theta)$ à θ fixé. En remplaçant $\Delta\theta$ par sa valeur, on obtient

$$T = \frac{l_1l_2}{\varepsilon}\theta_{\max}T_0.$$

2.2.3 Quelques variantes de la méthode Aliénor

Les premières courbes permettant de réduire le nombre de variables en une seule étape ont été proposées dans [1]; contrairement à la technique Aliénor classique qui nécessite plusieurs étapes intermédiaires pour exprimer n variables en fonction d'une seule. Pour un problème d'optimisation dans le pavé $\prod_{i=1}^n [a_i, b_i]$, la courbe considérée possède la représentation paramétrique suivante :

$$\begin{aligned} x_1(\theta) &= \frac{\alpha\alpha_2}{2\pi} \cos(\alpha_2) \\ x_2(\theta) &= \frac{\alpha\alpha_2}{2\pi} \sin(\alpha_2) \\ x_i(\theta) &= a_i + \frac{\alpha_i}{\theta_i}(b_i - a_i), \quad \text{pour } i = 3, \dots, n \end{aligned}$$

avec :

$$\theta_2 = \frac{\pi}{\alpha}((b_1 - a_1)^2 + (b_2 - a_2)^2)^{\frac{1}{2}}$$

$$\theta_i = \left(\frac{b_i - a_i}{\alpha}\right) \theta_{i-1}, \text{ pour } i = 3, \dots, n$$

$$\beta_i(\theta) = \left\lceil \frac{\theta}{\theta_i} \right\rceil, \text{ pour } i = 2, \dots, n$$

$$\alpha_i = (-1)^{\beta_i} \left[\theta - \left(\beta_i + \frac{1}{2}(-1)^{\beta_i+1} + 1 \right) \theta_i \right], \text{ pour } i = 2, \dots, n$$

On peut montrer que cette courbe est $\alpha\sqrt{n-1}$ -dense dans $\prod_{i=1}^n [a_i, b_i]$ [1].

La deuxième famille de courbes a été suggérée dans le cadre de l'élaboration d'une nouvelle variante d'Aliénor appelée Divanu (Diminution variables number) [3]. Leur représentation paramétrique est définie par la famille de fonctions :

$$x_i = \varphi(\theta)h_i(\theta), \quad i = 1, \dots, n$$

où φ est une fonction bornée et dérivable et $h(\theta) = (h_1(\theta), h_2(\theta), \dots, h_n(\theta))$ une fonction périodique.

Un choix approprié de φ et h permet d'assurer la précision de l'approximation (l' α -densité).

La troisième approche a été développée au départ par G. Mora et Y. Cherruault [19], mais A. Ziadi et Y. Cherruault [33] l'ont améliorée par l'obtention de nouveaux résultats qui permettent de construire de grandes classes de courbes α -denses. Cette dernière voie s'est révélée être la plus intéressante car les courbes obtenues possèdent des représentations paramétriques plus simples. Une étude complète sur la théorie des courbes α -denses est donnée dans [9]

2.3 La méthode de Brent

L'algorithme proposé par Brent s'applique aux fonctions deux fois continûment différentiables dont la dérivée seconde est bornée [5]. C'est une méthode unidimensionnelle qui consiste à construire une suite croissante de fonctions $(F_i)_{1 \leq i \leq k}$ paraboliques par morceaux, convergeant vers f .

L'idée de la méthode repose sur le résultat suivant :

S'il existe $M > 0$ tel que $|f''(x)| \leq 2M$, pour tout $x \in [a, b]$, alors $\forall x_1, x_2 \in [a, b]$ vérifiant $x_1 \leq x_2$, la parabole $\varphi(x)$ définie par $\frac{d^2\varphi(x)}{d(x^2)} = M$ pour tout $x \in [a, b]$, et vérifiant :

$$\begin{cases} \varphi(x_1) = f(x_1) \\ \varphi(x_2) = f(x_2) \end{cases}$$

satisfait l'inégalité suivante : $\varphi(x) \leq f(x)$; $\forall x \in [x_1, x_2]$.

2.3.1 Algorithme de Brent

1) Initialisation

Poser $k = 1$;

$$x_1 = \frac{f(b)-f(a)}{2M(b-a)} + \frac{a+b}{2};$$

$$x_\varepsilon = \arg \min (f(a), f(x_1), f(b));$$

$$f_\varepsilon = f(x_\varepsilon);$$

$$F_\varepsilon = \varphi_{[a,b]}(x_1);$$

$$F_1(x) = [\varphi_{[a,x_1]}, \varphi_{[x_1,b]}]; \quad (\text{voir remarque ci-dessous})$$

2) Etape $k = 2, 3, \dots$

Si $f_\varepsilon - F_\varepsilon \leq \varepsilon$, alors arrêter.

Sinon, déterminer $x_{k+1} \in \arg \min F_k([a, b])$;

Si $f(x_{k+1}) \leq f_\varepsilon$, alors poser $f_\varepsilon = f(x_{k+1})$, $x_\varepsilon = x_{k+1}$;

Poser $F_{k+1}(x) = [\varphi_{[a,x_1]}, \varphi_{[x_1,x_2]}, \dots, \varphi_{[x_k,x_{k+1}]}, \varphi_{[x_{k+1},b]}]$;

et $F_\varepsilon = \min F_{k+1}([a, b])$;

Poser $k = k + 1$;

Aller à l'étape 2).

Remarque

Pour toute subdivision $x_0 = a, x_1, \dots, x_{k-1}, x_k = b$ de $[a, b]$, on peut toujours construire une fonction F_k continue et parabolique par morceaux sur l'intervalle $[a, b]$, telle que $F_k(x) \leq f(x)$, $\forall x \in [a, b]$.

En effet, on construit à partir de chaque couple (x_j, x_{j+1}) formé de deux points successifs une parabole $\varphi_{[x_j, x_{j+1}]}$ définie sur l'intervalle $[x_j, x_{j+1}]$, et ceci pour $j = 1, \dots, k - 1$.

Ensuite, on définit la fonction F_k par :

$$F_k : [a, b] \longmapsto IR$$

$$x \longmapsto \varphi_{[x_j, x_{j+1}]}(x), \text{ si } x \in [x_j, x_{j+1}]$$

F_k est une fonction parabolique par morceaux, elle est notée :

$$F_k(x) = \left[\varphi_{[a, x_1]}, \varphi_{[x_1, x_2]}, \dots, \varphi_{[x_{k-1}, b]} \right]$$

Il est indiqué dans la littérature que la méthode de Brent est plus rapide que la méthode d'Evtushenko et celle de Piyavskii-Shubert [13], [29].

2.4 La méthode Aliénor couplée avec l'algorithme de Brent

Soit à résoudre le problème multidimensionnel :

$$(P) \quad \min_{(x_1, x_2, \dots, x_n) \in X} f(x_1, x_2, \dots, x_n),$$

où f est une fonction deux fois continûment différentiable ($f \in C^2(X)$) et $X = \prod_{i=1}^n [a_i, b_i]$.

Soit $h(\theta) = (h_1(\theta), h_2(\theta), \dots, h_n(\theta))$ une fonction deux fois continûment différentiable, représentant une courbe α -dense dans X , pour $\theta \in [0, \theta_{\max}]$.

Généralement, les fonctions h_i choisies sont de classe C^∞ .

Le problème (P) est alors approximé par le problème unidimensionnel (P')

$$(P') \quad \min_{\theta \in [0, \theta_{\max}]} f^*(\theta).$$

Puisque $f^* = f \circ h$ est deux fois continûment différentiable, on peut appliquer l'algorithme de Brent pour f^* sur l'intervalle $[0, \theta_{\max}]$.

Notons l_1 et l_2 , respectivement, les constantes de Lipschitz de f et h .

Soient M_1 et M_2 , respectivement, des bornes supérieures des dérivées secondes de f et h .

On peut en déduire que la dérivée seconde de f^* dans ce cas est bornée, c'est à dire $\exists M > 0$ tel que :

$$\left| f^{*''}(\theta) \right| \leq M, \forall \theta \in [0, \theta_{\max}]$$

En effet, on a :

$$h'(\theta) = (h'_1(\theta), \dots, h'_n(\theta))$$

et

$$h''(\theta) = (h''_1(\theta), \dots, h''_n(\theta))$$

Soient H_f le hessien de f , et $\|\cdot\|$ la norme euclidienne.

$$\begin{aligned} \text{Alors, } |f^{*''}(\theta)| &= |(f \circ h)''(\theta)| \\ &= |\langle H_f(h(\theta)).h'(\theta), h'(\theta) \rangle + \langle \nabla f(h(\theta)), h''(\theta) \rangle| \\ &\leq |\langle H_f(h(\theta)).h'(\theta), h'(\theta) \rangle| + |\langle \nabla f(h(\theta)), h''(\theta) \rangle| \\ &\leq \|H_f(h(\theta)).h'(\theta)\| \cdot \|h'(\theta)\| + \|\nabla f(h(\theta))\| \cdot \|h''(\theta)\| \\ &\leq \|H_f(h(\theta))\| \cdot \|h'(\theta)\| \cdot \|h'(\theta)\| + \|\nabla f(h(\theta))\| \cdot \|h''(\theta)\| \\ &= \|H_f(h(\theta))\| \cdot \|h'(\theta)\|^2 + \|\nabla f(h(\theta))\| \cdot \|h''(\theta)\| \\ &\leq M_1 l_2^2 + M_2 l_1 \end{aligned}$$

Il suffit de prendre $M = M_1 l_2^2 + M_2 l_1$

L'algorithme mixte Aliénor-Brent peut être donné comme suit :

2.4.1 Algorithme Aliénor-Brent

i) **Première étape :**

On définit l'application

$$h = (h_1, h_2, \dots, h_n) : [0, \theta_{\max}] \rightarrow \prod_{i=1}^n [a_i, b_i],$$

telle que la courbe paramétrée par $h(\theta)$ soit α -dense dans le pavé $\prod_{i=1}^n [a_i, b_i]$.

Considérons l'application h définie de $\left[0, \frac{\pi}{2\alpha_n}\right]$ dans $\prod_{i=1}^n [a_i, b_i]$ par :

$$h_1(\theta) = (b_1 - a_1) \sin(\alpha_1 \theta) + a_1$$

$$h_2(\theta) = (b_2 - a_2) \sin(\alpha_2 \theta) + a_2$$

.

.

.

$$h_n(\theta) = (b_n - a_n) \sin(\alpha_n \theta) + a_n$$

où $\alpha_1, \alpha_2, \dots, \alpha_n$ sont des paramètres donnés par :

$$\alpha_1 = (b_2 - a_2) \dots (b_n - a_n)$$

$$\alpha_2 = \frac{\alpha}{\pi} (b_3 - a_3) \dots (b_n - a_n)$$

$$\alpha_3 = \left(\frac{\alpha}{\pi}\right)^2 (b_4 - a_4) \dots (b_n - a_n)$$

.

.

.

$$\alpha_n = \left(\frac{\alpha}{\pi}\right)^{n-1}$$

avec

$$\alpha = \frac{1}{\sqrt{n-1}} \frac{\varepsilon}{2l_1},$$

où l_1 est la constante de Lipschitz de f , et ε la précision avec laquelle le minimum global est obtenu.

La courbe $h(\theta)$ est $(\frac{\varepsilon}{l_1}$ -dense) dans $\prod_{i=1}^n [a_i, b_i]$.

Pour la preuve on peut appliquer directement le Théorème 1 de [33].

Par ailleurs, il est facile de voir que la fonction h est lipschitzienne de constante l_2 telle que :

$$l_2 = \frac{1}{2} \left(\sum_{i=1}^n (b_i - a_i)^2 \alpha_i^2 \right)^{\frac{1}{2}} .$$

ii) **Deuxième étape :**

On applique maintenant l'algorithme de Brent pour la fonction

$$f^*(\theta) = f(h(\theta)), \text{ avec } \theta \in \left[0, \frac{\pi}{\alpha_n}\right] .$$

La fonction f^* est deux fois continûment différentiable et sa dérivée seconde est bornée par M .

1) **Initialisation**

Poser $k = 1$,

$$\theta_1 = \alpha_n \frac{f^*\left(\frac{\pi}{\alpha_n}\right) - f^*(0)}{\pi M} + \frac{\pi}{2\alpha_n} ,$$

$$\theta_\varepsilon = \arg \min \left(f^*(0), f^*(\theta_1), f^*\left(\frac{\pi}{\alpha_n}\right) \right),$$

$$f_\varepsilon^* = \min \left(f^*(0), f^*(\theta_1), f^*\left(\frac{\pi}{\alpha_n}\right) \right),$$

$$F_\varepsilon = \varphi_{\left[0, \frac{\pi}{\alpha_n}\right]}(\theta_1),$$

$$F_1 = \left[\varphi_{[0, \theta_1]}, \varphi_{\left[\theta_1, \frac{\pi}{\alpha_n}\right]} \right],$$

2) **Etape $k = 2, 3, \dots$**

Si $f_\varepsilon^* - F_\varepsilon \leq \frac{\varepsilon}{2}$, alors arrêter.

Sinon, déterminer $\theta_{k+1} \in \arg \min F_k \left(\left[0, \frac{\pi}{\alpha_n}\right] \right)$.

Si $f^*(\theta_{k+1}) \leq f_\varepsilon^*$, alors poser $f_\varepsilon^* = f^*(\theta_{k+1})$, $\theta_\varepsilon = \theta_{k+1}$.

Poser $F_{k+1}(\theta) = \left[\varphi_{[0, \theta_1]}, \varphi_{[\theta_1, \theta_2]}, \dots, \varphi_{[\theta_k, \theta_{k+1}]}, \varphi_{\left[\theta_{k+1}, \frac{\pi}{\alpha_n}\right]} \right]$

et $F_\varepsilon = \min F_{k+1} \left(\left[0, \frac{\pi}{\alpha_n}\right] \right)$.

Poser $k = k + 1$.

Aller à l'étape 2).

Remarque.

Pour une meilleure performance de la méthode, les fonctions composantes h_1, h_2, \dots, h_n peuvent être construites de telle façon que leurs variations décroissent avec la longueur des intervalles $[a_i, b_i]$, pour $i = 1, \dots, n$.

En effet, on considère l'application Ψ définie sur les indices des intervalles $[a_i, b_i]$ par : $\forall j = 1, \dots, n$, $\Psi(j)$ est l'ordre de classement de l'intervalle $[a_j, b_j]$ en ordonnant d'une manière décroissante par rapport à leur longueur la famille des intervalles $([a_i, b_i])_{1 \leq i \leq n}$, autrement dit $\Psi(j)$ est le nombre qui représente la position de l'intervalle $[a_j, b_j]$ selon l'ordre décroissant de la famille des longueurs $(b_i - a_i)_{1 \leq i \leq n}$.

Alors, dans ce cas la courbe paramétrée $h(\theta) = (h_1(\theta), h_2(\theta), \dots, h_n(\theta))$ est définie comme suit :

$$h_1(\theta) = (b_{\Psi(1)} - a_{\Psi(1)}) \sin(\alpha_1 \theta) + a_{\Psi(1)}$$

$$h_2(\theta) = (b_{\Psi(2)} - a_{\Psi(2)}) \sin(\alpha_2 \theta) + a_{\Psi(2)}$$

.

.

.

$$h_n(\theta) = (b_{\Psi(n)} - a_{\Psi(n)}) \sin(\alpha_n \theta) + a_{\Psi(n)}$$

et par conséquent :

$$x_{\Psi(i)} = h_i(\theta), \forall i = 1, \dots, n$$

$$\iff x_i = h_{\Psi^{-1}(i)}(\theta), \forall i = 1, \dots, n.$$

Donc

$$f^*(\theta) = f(h_{\Psi^{-1}(1)}(\theta), h_{\Psi^{-1}(2)}(\theta), \dots, h_{\Psi^{-1}(n)}(\theta)).$$

Ainsi, la courbe paramétrée par $h(\theta)$ sera de longueur minimale parmi les courbes appartenant à cette classe, i.e., en prenant toutes les permutations possibles de

$$(h_1(\theta), h_2(\theta), \dots, h_n(\theta)).$$

2.4.2 Convergence de la méthode

Théorème 2.4.1

La méthode mixte Aliénor couplée avec Brent appliquée au problème (P) converge vers le minimum global avec une précision inférieure ou égale à ε .

Démonstration

Désignons par m et m' , respectivement, les minima globaux de f et f^* , et désignons par f_ε^* le minimum global du problème (P) obtenu par la méthode mixte Aliénor couplée avec Brent.

Montrons que

$$f_\varepsilon^* - m \leq \varepsilon.$$

a) Puisque f est continue sur X , il existe un point $y \in X$ tel que : $m = f(y)$.

Par ailleurs, il existe $\theta_0 \in \left[0, \frac{\pi}{\alpha_n}\right]$ tel que $\|y - h(\theta_0)\| \leq \frac{\varepsilon}{2l_1}$,

donc

$$\|f(y) - f(h(\theta_0))\| \leq \frac{\varepsilon}{2}.$$

Il en résulte que :

$$f(h(\theta_0)) - m \leq \frac{\varepsilon}{2},$$

et puisque $m \leq m' \leq f(h(\theta_0))$, on en déduit que :

$$m' - m \leq \frac{\varepsilon}{2}. \quad (\text{i})$$

b) D'autre part f_ε^* est le minimum global du problème (P), obtenu par l'algorithme Aliénor couplé avec Brent, donc on obtient :

$$f_\varepsilon^* - F_\varepsilon \leq \frac{\varepsilon}{2},$$

où $F_\varepsilon = \min F_{k+1} \left(\left[0, \frac{\pi}{\alpha_n}\right] \right)$, et f_ε^* le minimum obtenu après k évaluations.

Pour tout $\theta \in \left[0, \frac{\pi}{\alpha_n}\right]$, on a $F_\varepsilon \leq f^*(\theta)$ donc $F_\varepsilon \leq m'$.

Il en résulte que

$$f_\varepsilon^* - m' \leq \frac{\varepsilon}{2}. \quad (\text{ii})$$

Ainsi, à partir de (i) et (ii) on en déduit que :

$$f_\varepsilon^* - m \leq \varepsilon. \quad \blacksquare$$

Remarque.

Cette méthode mixte Aliénor couplée avec Brent peut être parfois appliquée même avec f non différentiable sur tout l'ensemble X . Ceci, dans le cas où on peut construire une courbe $h(\theta)$ α -dense dans X qui évite les parties où f n'est pas différentiable.

2.4.3 Tests numériques

Pour compléter l'étude théorique de la méthode mixte Aliénor-Brent, nous avons effectué une série d'applications numériques sur des fonctions tests.

Nous étudierons les fonctions suivantes :

1- Rosenbrok : $100(x_2 - x_1^2)^2 + (1 - x_1)^2$
 $[-5; 5] \times [-5; 5]$

2- Goldstein : $[1 + (1 + x_2 + x_1)^2 f(x_1, x_2)] [30 + (2x_1 - 3x_2)^2 g(x_1, x_2)]$
 $[-2; 2] \times [-2; 2]$

avec $f(x_1, x_2) = 19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2$
 $g(x_1, x_2) = 18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2$

3- Branin : $4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
 $[-5; 5] \times [-5; 5]$

4- Wood : $100(x_2 - x_1^2)^2 + (1 - x_1^2)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 +$
 $10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$

$$[-2; 2] \times [-2; 2] \times [-2; 2] \times [-2; 2]$$

$$5- F(x_1, x_2) = \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right)$$
$$[-1; 1] \times [-1; 1]$$

$$6- G_2(x_1, x_2) = x_1^2 + x_2^2 - \frac{1}{10}(\cos(5\pi x_1) + \cos(5\pi x_2))$$
$$[-1; 1] \times [-1; 1]$$

$$7- G_6(x_1, \dots, x_6) = \sum_{i=1}^6 \left(x_i^2 - \frac{1}{10} \cos(5\pi x_i)\right)$$
$$[-1; 1]^6$$

Dans le tableau, on a utilisé les notations suivantes :

min : le minimum global

Nbre : le nombre d'itérations de l'algorithme

A-B : méthode mixte Aliénor-Brent

Nous avons présenté les résultats pour $\varepsilon = 0.01$.

Fonctions	La solution exacte	$A-B$	$Nbre$	
Rosenbrok	$x_1 = 1$ $x_2 = 1$ min = 0	$x_1 = 1.012$ $x_2 = 1.023$ min = 0.0002	22811	
Goldstein	$x_1 = 0$ $x_2 = -1$ min = 3	$x_1 = 0.0005$ $x_2 = -0.9975$ min = 3.0138	1897	
Branin	$x_1 = 0.08983$ $x_2 = -0.7126$ min = -1.0316	$x_1 = 0.08961$ $x_2 = -0.71321$ min = -1.031	13036	
Wood	$x_1 = 1$ $x_2 = 1$ $x_3 = 1$ $x_4 = 1$ min = 0	$x_1 = 1.005$ $x_2 = 1.003$ $x_3 = 0.997$ $x_4 = 0.989$ min = 0.00207	69542	
$F(x_1, x_2)$	$x_1 = \pm 1$ $x_2 = \pm 1$ min = 0.3678	$x_1 = -1$ $x_2 = -1$ min = 0.367	7721	
$G_2(x_1, x_2)$	$x_1 = 0$ $x_2 = 0$ min = -0.2	$x_1 = -0.001$ $x_2 = -0.002$ min = -0.19	24186	
$G_6(x_1, \dots, x_6)$	$x_1 = 0$ $x_2 = 0$ $x_3 = 0$ $x_4 = 0$ $x_5 = 0$ $x_6 = 0$ min = -0.6	$x_1 = 0.015$ $x_2 = 0.021$ $x_3 = -0.073$ $x_4 = -0.024$ $x_5 = -0.022$ $x_6 = 0.031$ min = -0.592	379958	

2.5 Conclusion

Les méthodes numériques qui permettent de chercher les solutions globales des problèmes à plusieurs variables, malgré leur importance, n'avaient pas été jusqu'alors très efficaces étant donné leur complexité et la longueur du temps de calcul. Dans la littérature, plusieurs travaux ont été consacrés aux problèmes d'optimisation globale unidimensionnels. La simplicité, relative, de ces problèmes a permis le développement des méthodes d'optimisation globale pour les problèmes multidimensionnels basés sur l'utilisation des techniques unidimensionnelles. La méthode de la transformation réductrice Aliénor est l'une des plus prometteuses de ces méthodes. Elle est basée sur l'idée de ramener l'optimisation d'une fonction multivariable à celle d'une fonction d'une seule variable ; pour ainsi disposer de toutes les méthodes performantes et les techniques connues faisant intervenir une seule variable.

La plupart des algorithmes mis en oeuvre à partir de cette méthode ont prouvé leur efficacité dans de différentes situations. C'est aussi le cas de la méthode mixte Aliénor-Brent donnée dans ce chapitre, comme le montre l'étude théorique et l'application numérique effectuée sur plusieurs fonctions tests.

Chapitre 3

UNE NOUVELLE FONCTION FILLED

3.1 Introduction

Soit à résoudre le problème suivant :

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

où f est une fonction définie sur l'ensemble $X \subseteq \mathbb{R}^n$.

Parmi les méthodes d'optimisation globale pour résoudre les problèmes non-linéaires et non-convexes on a la méthode de Tunneling et la méthode de fonction Filled, qui appartiennent à la même classe des méthodes de pénalité dont l'idée essentielle est de construire une fonction auxiliaire appelée fonction de pénalité.

La méthode de la fonction Filled est une approche pour résoudre le problème (P) et trouver le minimum global de la fonction f , sous les conditions suivantes :

1. f est continûment différentiable.
2. f possède un nombre fini de minimiseurs.
3. $f(x) \rightarrow +\infty$ si $\|x\| \rightarrow +\infty$ (f est coerssive).

Notons que la troisième condition (connue sous le nom de la coerssivité) implique l'existence d'un domaine fermé et borné $\Omega \subset \mathbb{R}^n$, tel que Ω contienne tous les minimiseurs de f , et la valeur de $f(x)$ quand x est sur la frontière de Ω est plus grande que la valeur de $f(x)$ quand x est à l'intérieur de Ω .

3.2 Méthode de la fonction Filled

Donnons d'abord quelques définitions, pour introduire l'essentiel de la méthode de la fonction Filled.

Définition 3.2.1

Un bassin (une vallée) de f à un minimiseur x_1 , est un domaine connexe B_1 qui contient x_1 , et dans lequel, partant d'un point quelconque la trajectoire de la descente de f converge vers x_1 ; mais en commençant à l'extérieur de B_1 la trajectoire de la descente de f ne converge pas vers x_1 .

Définition 3.2.2

Soit x_1 un maximiseur de f , on appelle colline (côte) de f au point x_1 , le bassin de la fonction $-f$ à son minimiseur x_1 .

Définition 3.2.3

On dit qu'un minimiseur local x_2 est supérieur au minimiseur local x_1 , si $f(x_2) > f(x_1)$, dans ce cas, on dit que le bassin B_2 est supérieur au bassin B_1 . L'ensemble des bassins supérieurs (resp. inférieurs) à un bassin courant B_1 de f est noté B_h (resp. B_l).

Définition 3.2.4 (La fonction Filled)

Une fonction H est appelée une fonction Filled de f au point x_1 (x_1 minimiseur de f) si :

- i) x_1 est un maximiseur de H dont le bassin B_1 de f est une partie de la coline de H .
- ii) H ne possède pas de point stationnaire dans les B_h .

iii) Il existe un point x' dans un B_l (si un tel bassin existe) qui minimise H sur la ligne qui lie x et x' .

La méthode de la fonction Filled se compose de deux phases, minimisation locale et remplissage (filling).

Phase 1 :

Dans cette phase, un minimiseur local x_1 de f est trouvé, on peut utiliser n'importe quelle méthode locale (par exemple : gradient, métrique variable,...).

Phase 2 :

Dans cette phase, une fonction dite fonction Filled (pleine) est construite. Cette fonction est donnée en expression de $f(x)$, elle a un maximiseur au point x_1 . De plus elle n'a pas de points stationnaires dans les bassins B_h , et possède un point stationnaire x_s dans un B_l . La phase 2 se termine quand un tel point x_s est trouvé.

Alors la méthode de la fonction Filled réitère la phase 1 avec x_s un point de départ pour trouver un nouveau minimiseur local x_2 de f (s'il existe), ainsi de suite.

Ce processus est répété jusqu'à obtention du minimum global.

Dans la littérature, plusieurs fonctions Filled ont été proposées. Parmi ces fonctions on a :

$$P(x, r, \rho) = \frac{1}{r+f(x_1)} \exp\left(-\frac{\|x-x_1\|}{\rho^2}\right)$$

$$0. G(x, r, \rho) = -[\rho^2 \ln(r + f(x))] + \|x - x_1\|^p$$

$$Q(x, a) = -[(f(x) - f(x_1))] \exp(a \|x - x_1\|^p)$$

tels que $p = 1$ ou 2

r et ρ sont deux paramètres ajustés.

a est un facteur poids positif ajusté.

Les deux fonctions P et G dépendent de deux paramètres (r et ρ) qui ont besoin d'être convenablement itérés et coordonnés entre eux ; donc la réalisation de l'algorithme est assez compliquée. Pour cette raison, il est convenu que la fonction Q est meilleure que les deux autres, puisqu'elle a besoin d'un seul paramètre. Cependant la fonction Q contient une exponentielle dont l'argument est le produit du facteur poids et la norme.

Plus le facteur a devient grand, plus la propriété de remplissage (filling) a besoin d'être préservée. L'augmentation rapide de la valeur du terme exponentiel va faire défaut dans les calculs, même si la taille de la région réalisable est modérée. Dans la pratique ce genre de problème mal conditionné arrive souvent.

Pour utiliser la fonction Q en pratique, plusieurs précautions supplémentaires doivent être incorporées dans l'algorithme.

Il est clair que le terme exponentiel dans la fonction Q limite sérieusement son application aux problèmes pratiques d'optimisation globale.

Une nouvelle fonction Filled H a été proposée par X. Liu [16] :

$$H(x) = \frac{1}{\ln(1 + f(x) - f(x_1))} - a \|x - x_1\|^2$$

où a est un réel positif, utilisé comme facteur poids et $\|\cdot\|$ est la norme euclidienne.

Notons que dans la phase 2 de la méthode on a $f(x) > f(x_1)$, car x_1 est un minimiseur local de f . Par conséquent $f(x) > f(x_1) - 1$, ce qui assure l'existence de la fonction H localement. Durant cette phase, l'algorithme vérifie la valeur de la fonction au point généré par l'itération courante. Si en un certain point x_s on obtient $f(x_s) < f(x_1)$, alors x_s est dans un bassin inférieur à celui de x_1 , dans ce cas, la phase 2 se termine au point x_s et l'algorithme de la fonction Filled réitère la phase 1, en prenant x_s comme point de départ.

Les propriétés de remplissage (filling) de la fonction H sont données par les théorèmes suivants :

Théorème 3.2.1

Soient $d \in \mathbb{R}^n$ et $f(x) > f(x_1)$. Si

$$d^T \nabla f(x) \geq 0 \text{ et } d^T (x - x_1) > 0$$

ou

$$d^T \nabla f(x) > 0 \text{ et } d^T (x - x_1) \geq 0$$

alors, d est une direction de descente pour la fonction H au point x .

Théorème 3.2.2

Soient $f(x) > f(x_1)$, $d^T \nabla f(x) < 0$ et $d^T(x - x_1) > 0$. Si

$$a > -\frac{d^T \nabla f(x)}{2d^T(x - x_1) [\ln(1 + f(x) - f(x_1))]^2 (1 + f(x) - f(x_1))} = a_l(x) \quad (1)$$

alors, d est une direction de descente pour la fonction H au point x .

Pour les démonstrations de ces théorèmes on peut voir [16].

3.2.1 Analyse du facteur poids

Le facteur poids a joue un rôle crucial dans la fonction Filled. Théoriquement la valeur de a doit être suffisamment grande pour préserver la capacité de remplissage (filling) désirée.

Cependant, sur le plan numérique la valeur de a devrait être petite pour exécuter convenablement les procédures de calcul. De plus, la fonction Filled est plus efficace (plus robuste), lorsque a prend des valeurs relativement petites.

Dans ce qui suit, on fera une comparaison entre la fonction H et la fonction Q en termes de borne inférieure de a .

Soit $d \in \mathbb{R}^n$, alors d'après la formule de la fonction Q on a :

$$d^T \nabla Q(x) = \exp(a \|x - x_1\|^2) [d^T \nabla f(x) + 2a(f - f_1)d^T(x - x_1)]$$

Par conséquent, avec la même condition donnée dans le Théorème 3.2.2, si :

$$a > -\frac{d^T \nabla f(x)}{2d^T(x - x_1)(f(x) - f(x_1))} = a_q(x) \quad (2)$$

Alors, d est une direction de descente de Q au point x (puisque un tel a vérifie $d^T \nabla Q(x) < 0$).

Ensuite, considérons le rapport :

$$\frac{a_q}{a_l} = \frac{[\ln(1 + f(x) - f(x_1))]^2 (1 + f(x) - f(x_1))}{f(x) - f(x_1)}$$

On note que $\frac{a_q}{a_l}$ augmente d'une façon monotone avec l'argument $f(x) - f(x_1)$, par conséquent si $f(x) - f(x_1) > 1.1$, alors $a_l(x)$ est toujours plus petit que $a_q(x)$. Cela implique que même avec un petit facteur poids a la fonction H préserve la propriété de remplissage (filling) désirée.

3.2.2 L'algorithme de la fonction Filled

On a vu que le paramètre poids a joue un rôle crucial dans la fonction Filled. Généralement ce paramètre est estimé par suite de quelques essais de l'algorithme, cependant il est possible de développer une formule pour adapter a si $x \in \mathbb{R}$, dans ce cas la formule (1) peut être remplacée par :

$$a = - \frac{\xi |f'(x)|}{2 |x_s - x_1| [\ln(1 + f(x) - f(x_1))]^2 (1 + f(x) - f(x_1))} \quad (3)$$

où $\xi > 1$ et x_s le point trouvé à la fin de la phase 2 .

Le processus de remplissage procède comme suit :

Au début, on pose $a = a_0 > 0$. Supposons que la valeur de a n'est pas assez grande, alors le point x_s est dans un bassin supérieur à B_1 , autrement dit, à la fin de la phase 1 du second cycle, un nouveau minimiseur x_2 de f est trouvé, tel que $f(x_2) > f(x_1)$. Dans ce cas, on recommence la phase 2 en utilisant la formule précédente (3) pour a et on répète la même procédure pour minimiser $H(x)$, comme dans la première itération (le premier cycle).

Le temps où l'itération arrive à x_s , la formule (3) va faire en sorte que d soit une direction de descente de la fonction H .

Dans ce qui suit on présente un algorithme pour l'optimisation globale d'une fonction f à une seule variable avec la notation suivante :

$[a_1, b_1]$: l'intervalle qui contient un minimiseur global de f .

a : le facteur poids dans la formulation de la fonction Filled.

x_1 : le minimiseur local trouvé de f .

$f_1 := f(x_1)$

x_0 : point de départ pour la phase1

x_c : point de départ pour la phase2

δ : un nombre réel positif (sensé être petit) utilisé pour construire x_c .

Alors, l'algorithme est donné comme suit :

Algorithme de la fonction Filled

Initialisation :

Donner un x_0 et un $a > 0$.

Etape 1 :

Commencer la phase 1 de la méthode de la fonction Filled.

Activer la procédure de minimisation locale de la fonction f , partant de x_0 .

Trouver un minimum x_{min} .

Si $f(x_{min}) < f_1$, alors on pose $x_1 = x_{min}$ et $f_1 = f(x_{min})$;

Sinon on utilise la formule (3) pour adapter a et réitérer l'**Etape 1**.

Etape 2 :

Commencer la phase 2 de la méthode de la fonction Filled.

Poser $x_c = x_1 + \delta$.

Etape 3 :

Construire la fonction Filled H en utilisant x_1 et le paramètre a .

Etape 4 :

Procéder à la minimisation locale de H à partir du point de départ x_c , on obtient un point x .

Etape 5 :

1) Si $x = b_1$, alors poser $x_c = x_1 - \delta$ et aller à l'**Etape3**.

- 2) Si $x = a_1$, alors prendre le plus petit minimum comme minimum global et arrêter.
- 3) Si $f(x) < f(x_1)$, alors prendre $x_0 = x$ et aller à l'**Etape 1**.
- 4) Si x est un minimiseur local de H , alors poser $x_0 = x$ et aller à l'**Etape 1**.

Soulignons que cet algorithme ne diminue pas les qualités déterministes de la méthode de la fonction Filled.

En effet dans le cas où $x \in \mathbb{R}$, il y a seulement deux directions de recherches possibles, les directions positive et négative de l'axe de x .

Pour le cas multidimensionnel, même pour $x \in \mathbb{R}^2$, on a besoin d'utiliser des approches probabilistes pour choisir les directions de descentes pour la fonction Filled.

3.3 Une nouvelle fonction Filled

Dans cette partie, on propose une nouvelle fonction Filled K , qu'on a construite à partir d'une fonction lipschitzienne f .

La fonction K est définie comme suit :

$$K(x) = \frac{1}{\ln\left(1 + \frac{f(x)-f(x_1)}{L}\right)} - a \|x - x_1\|^2,$$

où a est un réel positif, utilisé comme facteur poids et $\|\cdot\|$ la norme euclidienne.

La fonction f est lipschitzienne de constante L , définie sur l'ensemble $X \subseteq \mathbb{R}^n$, et x_1 le minimum local de f obtenue après application de la phase 1 d'où $f(x) > f(x_1)$, ce qui assure l'existence de la fonction K .

Les propriétés de remplissage de la fonctions K sont données par les théorèmes suivants.

Théorème 3.3.1

Soient $d \in \mathbb{R}^n$ et $f(x) > f(x_1)$. Si

$$d^T \nabla f(x) \geq 0 \text{ et } d^T(x - x_1) > 0,$$

ou

$$d^T \nabla f(x) > 0 \text{ et } d^T(x - x_1) \geq 0,$$

alors, d est une direction de descente pour la fonction K au point x .

Démonstration :

On montre que $d^T \nabla K(x) < 0$.

Comme la fonction K est donnée par

$$K(x) = \frac{1}{\ln\left(1 + \frac{f(x)-f(x_1)}{L}\right)} - a \|x - x_1\|^2,$$

alors,

$$d^T \nabla K(x) = -\frac{d^T \nabla f(x)}{L \left[\ln\left(1 + \frac{f(x)-f(x_1)}{L}\right) \right]^2 \left(1 + \frac{f(x)-f(x_1)}{L}\right)} - 2ad^T(x - x_1).$$

On a :

$$f(x) > f(x_1), d^T \nabla f(x) \geq 0, d^T(x - x_1) > 0, L > 0 \text{ et } a > 0,$$

alors, on trouve évidemment que $d^T \nabla K(x) < 0$, d'où d est une direction de descente de la fonction K au point x .

On trouve le même résultat, si on prend la deuxième condition :

$$d^T \nabla f(x) > 0 \text{ et } d^T(x - x_1) \geq 0. \quad \blacksquare$$

Théorème 3.3.2

Soient $f(x) > f(x_1)$, $d^T \nabla f(x) < 0$ et $d^T(x - x_1) > 0$. Si

$$a > -\frac{d^T \nabla f(x)}{2Ld^T(x - x_1) \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} = a_m(x),$$

alors, d est une direction de descente pour la fonction K au point x .

Démonstration :

On a :

$$f(x) > f(x_1), \quad d^T \nabla f(x) < 0, \quad d^T(x - x_1) > 0, \quad \text{et } a > a_m(x).$$

Rappelons que $d^T \nabla K(x)$ est donnée par :

$$d^T \nabla K(x) = -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} - 2ad^T(x - x_1).$$

On a :

$$\left\{ \begin{array}{l} i) -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} > 0 \\ ii) 2ad^T(x - x_1) > -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} \end{array} \right.$$

Donc $d^T \nabla K(x) < 0$, d'où d est une direction de descente de K au point x . ■

Théorème 3.3.3

Soient $f(x) > f(x_1)$, $d^T \nabla f(x) < 0$ et $d^T(x - x_1) > 0$. Si

$$a < a_m(x),$$

alors, d est une direction ascendante pour la fonction K au point x .

Démonstration :

Supposons que $f(x) > f(x_1)$, $d^T \nabla f(x) < 0$, $d^T(x - x_1) > 0$ et $a < a_m(x)$

On a

$$\left\{ \begin{array}{l} i) \quad d^T \nabla K(x) = -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} - 2ad^T(x - x_1) \\ ii) \quad -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} > 0 \\ iii) \quad -2ad^T(x - x_1) > \frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)} \end{array} \right.$$

Alors,

$$2ad^T(x - x_1) < -\frac{d^T \nabla f(x)}{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)},$$

et donc $d^T \nabla K(x) > 0$, d'où d est une direction ascendante pour K au point x . ■

Remarques

i) La propriété de remplissage de la fonction Filled K est caractérisée par les théorèmes précédents.

- Théorème 3.3.1 montre que dans une région ascendante du bassin en court (B_1) ou d'un bassin supérieur à B_1 , d est toujours une direction de descente de la fonction K .

- Théorème 3.3.2 montre que dans la région de descente d'un bassin supérieur à B_1 , d reste une direction de descente de la fonction K , pourvu que le facteur poids a soit suffisamment grand.

- Dans le Théorème 3.3.3, il est indiqué que dans un bassin inférieur à B_1 , d peut devenir une direction ascendante de la fonction Filled K .

ii) Dans le cas unidimensionnel, on peut donner un algorithme utilisant la nouvelle fonction Filled K , semblable à celui donné avec la fonction Filled H .

3.4 Etude comparative entre les fonctions Filled

Le facteur poids a joue un rôle crucial dans la fonction Filled. Théoriquement la valeur de a doit être suffisamment grande pour préserver la capacité de remplissage (filling) désirée. Cependant, sur le plan numérique la valeur de a devrait être petite pour exécuter convenablement les procédures de calcul. De plus, la fonction Filled est plus efficace, lorsque a prend des valeurs relativement petites.

Dans ce qui suit, on fera une comparaison entre la fonction K et les fonctions H et Q en termes de borne inférieure de a .

D'après ce qui précède on a constaté que la fonction Filled H est meilleure que la fonction Q , puisque il est prouvé que $a_l(x)$ est toujours plus petit que $a_q(x)$ si $f(x) - f(x_1) > 1.1$

3.4.1 Comparaisons entre la fonction K et la fonction Q

Rappelons que,

$$Q(x) = - [(f(x) - f(x_1))] \exp(a \|x - x_1\|^p),$$

$$K(x) = \frac{1}{\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right)} - a \|x - x_1\|^2,$$

$$a_q(x) = - \frac{d^T \nabla f(x)}{2d^T(x - x_1)(f(x) - f(x_1))},$$

$$a_m(x) = - \frac{d^T \nabla f(x)}{2Ld^T(x - x_1) \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)}.$$

Avec les mêmes conditions données dans le Théorème 3.3.2, si :

$$a > a_q(x),$$

alors, d est une direction de descente de Q au point x (puisque un tel a vérifie $d^T \nabla Q(x) < 0$).

Et si :

$$a > a_m(x),$$

alors, d est une direction de descente de K au point x (puisque un tel a vérifie $d^T \nabla K(x) < 0$).

En suite, considérons le rapport :

$$\frac{a_q}{a_m} = \frac{L \left[\ln \left(1 + \frac{f(x)-f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x)-f(x_1)}{L} \right)}{f(x) - f(x_1)}.$$

On note que $\frac{a_q}{a_m}$ augmente d'une façon monotone avec l'argument $f(x) - f(x_1)$, par conséquent si $f(x) - f(x_1) > L1.1$, alors $a_m(x)$ est toujours plus petit que $a_q(x)$. Cela implique que la fonction K est meilleure que la fonction Q quelque soit $L > 0$.

3.4.2 Comparaisons entre la fonction K et la fonction H

Rappelons que,

$$H(x) = \frac{1}{\ln(1 + f(x) - f(x_1))} - a \|x - x_1\|^2,$$

$$K(x) = \frac{1}{\ln \left(1 + \frac{f(x)-f(x_1)}{L} \right)} - a \|x - x_1\|^2,$$

$$a_l(x) = - \frac{d^T \nabla f(x)}{2d^T(x - x_1) [\ln(1 + f(x) - f(x_1))]^2 (1 + f(x) - f(x_1))},$$

$$a_m(x) = - \frac{d^T \nabla f(x)}{2Ld^T(x - x_1) \left[\ln \left(1 + \frac{f(x)-f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x)-f(x_1)}{L} \right)}.$$

Avec les mêmes conditions données dans le Théorème 3.3.2, si :

$$a > a_l(x),$$

alors, d est une direction de descente de H au point x (puisque un tel a vérifie $d^T \nabla H(x) < 0$).

Et si :

$$a > a_m(x),$$

alors, d est une direction de descente de K au point x (puisque un tel a vérifie $d^T \nabla K(x) < 0$).

Ensuite, considérons le rapport :

$$\frac{a_l}{a_m} = \frac{L \left[\ln \left(1 + \frac{f(x) - f(x_1)}{L} \right) \right]^2 \left(1 + \frac{f(x) - f(x_1)}{L} \right)}{[\ln(1 + f(x) - f(x_1))]^2 (1 + f(x) - f(x_1))}.$$

Pour $L = 1$, on a $a_m = a_l$ et $K = H$.

Pour $0 < L < 1$, on a $\frac{a_l}{a_m} > 1$ si $f(x) - f(x_1) > L1.1$, d'où $a_m < a_l$, par conséquent la fonction K est meilleure que la fonction H .

Pour $L > 1$, on a $\frac{a_l}{a_m} < 1$ si $f(x) - f(x_1) > L1.1$, d'où $a_m > a_l$. Dans ce cas, la fonction H est meilleure que la fonction K .

3.5 Conclusion

Les méthodes de fonctions Filled sont utilisées pour résoudre les problèmes d'optimisation globale multidimensionnelle. Plusieurs fonctions Filled ont été reportées dans la littérature. Cependant ces fonctions traditionnelles nécessitent beaucoup de calculs ; cela est dû au terme exponentiel ou bien au paramètres multiples dans leurs formulations. En plus, ces fonctions ont besoin d'un facteur poids assez grand pour préserver la propriété de remplissage, toutes ces caractéristiques conduisent à des complications dans les calculs quand on passe à l'application.

On a proposé une nouvelle fonction Filled K dans le but de remédier aux inconvénients

précédents comme c'est le but aussi de la fonction H qui a été proposée par Liu. Dans la formule de notre fonction K on ne trouve pas de terme exponentiel, ni de paramètres multiples. De plus, la borne inférieure du facteur poids a est plus petite que celles des fonctions Filled traditionnelles. De ce fait, il semble que la fonction K est mieux applicable en pratique en diminuant le temps de calcul.

CONCLUSION GENERALE

L'optimisation est l'un des domaines les plus importants dans les mathématiques appliquées, qu'elle soit locale ou globale. Nombreux sont les problèmes rencontrés en mathématiques appliquées et en industrie qui nécessitent pour leur résolution l'utilisation des techniques numériques d'optimisation. Ces problèmes sont faciles à résoudre si la fonction à minimiser dépend d'une seule variable ou si on cherche un minimum local. Les difficultés deviennent beaucoup plus considérables lorsqu'on a besoin du minimum global d'une fonction multivariable. Malgré leur importance et la contribution de beaucoup de chercheurs, les algorithmes d'optimisation globale ne sont pas encore en mesure de satisfaire les exigences des utilisateurs. Dans l'optimisation locale, les classes de fonctions tests, les critères d'efficacité, les critères d'arrêt et les conditions des expériences numériques sont standardisés. Ici, bien que certains résultats soient valables aussi pour l'optimisation globale, une standardisation analogue pour cette dernière n'existe pas actuellement. Les raisons sont diverses : (i) les classes de problèmes d'optimisation pour lesquelles un algorithme donné est efficace sont à peine ou presque pas définies ; (ii) la plupart des algorithmes dépend des paramètres dont le choix est vaguement justifié ou difficile à formuler. C'est la raison pour laquelle l'inventeur d'un algorithme peut généralement l'exploiter beaucoup plus efficacement qu'un utilisateur ordinaire ; (iii) la pauvreté de précision et l'insuffisance des résultats numériques posent des difficultés quant à l'estimation de l'influence des caractéristiques d'un problème d'optimisation tels que la dimension, le nombre de minimiseurs locaux, la régularité de l'ensemble faisable, etc. En effet, l'efficacité d'un algorithme en dépendrait beaucoup. Une difficulté supplémentaire, qui n'est pas une caractéristique particulière de l'optimisation globale, est due à l'existence de divers critères d'efficacité utilisés dans la comparaison numérique. Le temps de calcul exprimé en unité standard est le critère d'efficacité le plus commun. Parfois le nombre d'évaluations est utilisé au lieu du temps. Le deuxième critère en importance

est la fiabilité qui est très difficile à estimer dans les cas compliqués. La simplicité de la mise en oeuvre de l'algorithme et l'espace mémoire requis par celui-ci sont aussi des critères de qualité qui jouent un rôle prédominant pour certains problèmes. Ils agissent non seulement dans l'aptitude de résolution de tels problèmes par un algorithme donné, mais présentent aussi un facteur financier non négligeable concernant le coût d'exécution sur les gros ordinateurs. Ainsi, comme il a été signalé par d'autres auteurs, nous pensons que le développement des méthodes globales nécessite une nouvelle vision quant aux méthodes numériques. Non seulement ces techniques doivent conduire à la solution globale mais il faut aussi qu'elles soient fondées sur des bases mathématiques solides. Cette dernière exigence est cruciale, car dans la plupart des cas une étude théorique rigoureuse donne plus d'informations sur l'efficacité que les expériences numériques.

Dans ce qui précède, nous nous sommes intéressés à la méthode de la transformation réductrice Aliénor, dont l'idée de base consiste à "approcher" l'espace \mathbb{R}^n par une courbe paramétrée. Ce qui fait que nous avons toujours la possibilité de ramener n variables ($n \geq 2$) à une variable unique, ce qui permet d'utiliser les résultats très riches concernant l'analyse fonctionnelle à une seule variable. Nous avons voulu exploiter cette propriété en couplant la méthode multidimensionnelle Aliénor avec la méthode unidimensionnelle de Brent, ainsi nous avons donné naissance à une nouvelle variante de la méthode Aliénor. L'algorithme obtenu à partir de ce couplage est simple et efficace comme le montre l'étude faite et les résultats numériques obtenus.

Nous avons, aussi, tenté de construire une nouvelle fonction Filled qu'on a nommée K , qui semble être meilleure que les fonctions Filled traditionnelles telles que P , G et Q . Nous avons étudié les propriétés de remplissage de cette fonction en donnant quelques théorèmes et nous l'avons comparée aux fonctions Filled Q et H . Il reste cependant à faire une application numérique pour que l'étude soit complète.

Bibliographie

- [1] H. AMMAR and Y. CHERRUAULT, *Approximation of a Several Variables function by a one Variable Function and Application to Global Optimization*, Math. Comput. Modelling, Vol. 18, No. 2, pp. 17-21, 1993.
- [2] W. BARITOMPA and A. CUTLER, *Accelerations for Global Optimisation Covering Methods Using Second Derivatives*, Journal of Global Optimisation 4, pp 329-341, 1994.
- [3] O. BENDIAB and Y. CHERRUAULT, *Divanu : A New Method for Global Optimization in Dimension n*, International Journal of Bio-Medical Computing, No. 38, 1995.
- [4] F. H. BRANIN, *A widely convergent method for finding multiple solutions of simultaneous non-linear equations*, IBM J. Res. Develop., 16, 504-522, 1972.
- [5] R. P. BRENT, *Algorithms for minimization without derivation*, Englewood Cliffs, N. J. Prentice Hall, 1973.
- [6] Y. CHERRUAULT, *Modèles et Méthodes mathématiques pour les sciences du vivant*, Presses Universitaire de France, 1998.
- [7] Y. CHERRUAULT, *Optimisation, Méthodes locales et globales*, Presses Universitaire de France, 1999.
- [8] Y. CHERRUAULT and A. GUILLEZ, *Une méthode pour la recherche du minimum global d'une fonctionnelle*, C.R.A.S, Paris, T.296,Série I, pp.175-178,1983.

- [9] Y. CHERRUAULT et G. MORA, *Optimisation globale, Théorie des courbes α -denses*, Ed Economica, 2005.
- [10] G. CHOQUET, *Cours d'analyse, tome II, Topologie*, Masson et Cie, Paris, 1973.
- [11] Y. G. EVTUSHENKO, *Numerical Optimization Techniques*, Springer, Berlin, 1985.
- [12] R. HORST, *A General class of branch-and-bound methods in global optimization with some new approach for concave minimization*, J. Optimiz. Theory and Applic., 51, No. 2, pp. 271-291, 1986.
- [13] R. HORST and H. TUY, *Global Optimization, Deterministic Approach*, Springer-Verlag, Berlin, 1993.
- [14] R. HORST and P.M. PARDALOS, *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1995
- [15] R. HORST, P.M. PARDALOS and N. V. THOAI, *Introduction to Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1995.
- [16] X. LIU, *Finding Global Minima with a Computable Filled Function*, Journal of Global Optimisation 19, pp 329-341, 2001.
- [17] R. H. MLADINEO, *An Algorithm for Finding the Global Maximum of a Multimodal, Multivariate Function*, Mathematical Programming, 34, pp. 188-200, 1986.
- [18] G. MORA and Y. CHERRUAULT, *Characterization and Generation of α -Dense Curves*, Comp. and math. with Applic. Vol. 33, No. 9, p.p. 83-91, 1997.
- [19] G. MORA and Y. CHERRUAULT, *The theoretic calculation time associated with α -Dense Curves*, Kybernetes, Vol. 27, No. 8, 9, pp. 919-939, 1998.
- [20] G. MORA, Y. CHERRUAULT and A. ZIADI, *Functional Equations Generating Space-Densifying Curves*, Comp. and math. with Applic. 39, pp. 45-55, 2000.
- [21] J. PINTER, *Extended univariate algorithms for dimensional global optimization*, Computing, 36, No.1, pp. 91-103, 1986a.
- [22] J. PINTER, *Globally convergent methods for dimensional multiextremal optimization*, Optimization, 17, No.2, pp. 187-202, 1986b.

- [23] J. PINTER, *Branch-and-bound methods for solving global optimization problems with Lipschitzian structure*, Optimization, 19, No. 1, pp. 101-110, 1988.
- [24] S. A. PIYAVSKII, *An algorithm for finding the absolute minimum for a function*. Theory of Optimal Solutions, No.2, Kiev, IK AN USSR, pp. 13-24. 1967.
- [25] S. A. PIYAVSKY, *An algorithm for finding the absolute extremum of a function*, USSR Comp. Mathem. and Mathem. phys., 12, No. 4, pp. 888-896, 1972.
- [26] H. SAGAN, *Space-filling Curves*, Springer-Verlag, New York, 1994.
- [27] L. SCHWARTZ, *Analyse I, Théorie des Ensembles et Topologie*, Hermann, Paris, 1991.
- [28] B. O. SHUBERT, *A sequential method seeking the global maximum of a function*, SIAM, Journal of Numerical Analysis, Vol. 9, No. 3, pp. 379-388, 1972.
- [29] A. TORN and A. SILINSKAS, *Global Optimization*, Springer-Verlag, New York, 1988.
- [30] G. R. WOOD, *Multidimensional bisection applied to global optimisation*, Comp. and math. with Applic. 21, pp. 161-172, 1991.
- [31] A. A. ZHIGLJAVSKY, *Theory of Global Random Search*, Kluwer Academic Publishers, Dordrecht, 1991.
- [32] A. ZIADI and Y. CHERRUAULT, *Generation of α -dense Curves in a cube of \mathbb{R}^n* , Kybernetes Vol. 27 No.4, pp. 416-425, 1998.
- [33] A. ZIADI and Y. CHERRUAULT, *Generation of α -dense curves and application to global optimization*, Kybernetes, Vol. 29 No. 1, pp. 71-82, 2000.
- [34] A. ZIADI, Y. CHERRUAULT et G. MORA, *The existence of α -dense curves with minimal length in a metric space*, Kybernetes, Vol. 29 No. 2, pp. 219-230, 2000.
- [35] A. ZIADI, Y. CHERRUAULT and G. MORA, *Global Optimization, a New Variant of the Alienor Method*, Comp. and math. with Applic. 41, pp. 63-71, 2001.
- [36] A. ZIADI, S. KHELLADI and Y. CHERRUAULT, *The Alienor Method Coupled to the Brent algorithm*, Kybernetes, Vol. 34 No. 7/8, pp. 1059-1069, 2005.

- [37] A. ZIADI, D. GUETTAL and Y. CHERRUAULT, *Global Optimization : the Alienor mixed method with Piyavskii-Shubert technique*, Kybernetes, Vol. 34 No. 7/8, pp. 1049-1058, 2005.