

1. Introduction

Le problème d'optimisation sans contraintes consiste à chercher une solution admissible $\mathbf{x} \in \mathcal{R}^n$ minimisant une fonction objective $f(\mathbf{x})$.

Pour résoudre ce problème, plusieurs méthodes de recherches de l'optimum existent, on peut citer parmi ces méthodes quatre types de méthodes courantes :

- Méthodes analytiques.
- Méthodes de premier ordre.
- Méthodes du second ordre.
- Méthodes d'ordre zéro (directes)

1.1. Méthode analytique

Cette approche suppose que la fonction objectif $f(\mathbf{x})$ est dérivable aux moins deux fois. Puisque elle est basée sur le calcul du gradient $\nabla f(\mathbf{x})$ à partir duquel on calcul les points critiques (racines du gradient), et le calcul de la Hessienne $\nabla^2 f(\mathbf{x})$ à partir de laquelle on déduit la nature de ces points critiques. Cette méthode n'est pas pratique pour les systèmes complexes.

1.2. Méthodes de premier ordre

Ce sont des méthodes itératives qui nécessitent le calcul de la fonction $f(\mathbf{x})$ et son gradient $\nabla f(\mathbf{x})$ pour chaque itération. Ces méthodes consistent à construire une suite approximante $\mathbf{x}^{(k)}$ par récurrence, c'est-à-dire : on se donne un point initial $\mathbf{x}^{(0)} \in \mathcal{R}^n$ et on construit $\mathbf{x}^{(k+1)}$ en fonction de $\mathbf{x}^{(k)}$, telle que : $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$. Comme exemple de méthodes de premier ordre, on cite les méthodes de la relaxation, méthodes du gradient, méthodes du gradient conjugué,...

1.3. Méthodes de second ordre

Ce sont des méthodes itératives qui demandent l'évaluation de des fonctions $f(\mathbf{x})$, $\nabla f(\mathbf{x})$ et $\nabla^2 f(\mathbf{x})$ pour chaque itération, et résoudre implicitement les équations implicites obtenues à partir des conditions initiales. Parmi ces méthodes on peut citer la méthode de Newton, Levenberg-Marquardt,...

1.4. Méthodes d'ordre zéro

Ces méthodes ne nécessitent que le calcul de $f(\mathbf{x})$ pour chaque itération, elles sont généralement lentes et ne sont pas intéressantes que si $\nabla f(\mathbf{x})$ est indéfini ou il est difficile à évaluer. On peut citer comme exemple, la méthode de Nelder-Mead, la méthode de recherche de la section dorée, la méthode des directions conjuguées de Powell,...

2. Méthodes itératives de recherche de l'optimum

En général, une solution explicite ne peut pas être obtenue à partir des conditions théoriques. On va donc utiliser des méthodes itératives, qui consistent à partir d'une valeur initiale $\mathbf{x}^{(0)}$ permettront de calculer la suite des valeurs : $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(i)}, \dots, \mathbf{x}^{(k)}$. de manière à obtenir la relation : $f(\mathbf{x}^{(k)}) < \dots < f(\mathbf{x}^{(i)}) < \dots < f(\mathbf{x}^{(1)}) < f(\mathbf{x}^{(0)})$.

Comme cette suite numérique est bornée inférieurement par $f(x^*)$, alors elle convergera vers le minimum global x^* , si ce minimum global x^* est unique, et s'il n'y a pas d'autres minimums locaux. Dans le cas contraire, seul l'un des minimums locaux est atteint.

2.1. Méthode de gradient

Dans cette méthode on passe d'un estimé $x^{(k)}$ de la solution x^* à l'estimé suivante $x^{(k+1)}$, par l'algorithme suivant :

Pour $k=0,1,2,\dots$ jusqu'à ce que $|x^{(k+1)}-x^{(k)}| < \varepsilon$ (ε : petit nombre positif).

$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \nabla_x f(x^{(k)})$ (où : $-\nabla_x f(x^{(k)})$ est la direction de descente ou direction de recherche)

Deux techniques existent pour déterminer le scalaire $\alpha^{(k)}$ appelé aussi pas de calcul (pas de descente).

- 1) Méthode de pas de descente : on choisit le pas tel que : $f(x^{(k+1)}) < f(x^{(k)})$.
- 2) Méthode de pas de descente optimale : on détermine le pas $\alpha^{(k)}$ qui minimise $f(x^{(k)})$.

2.2. Méthode de gradient conjugué

Sachant que le choix de la direction de descente $P(k) = -\nabla_x f(x(k))$ n'est pas en général le meilleur en pratique. Alors une autre méthode qui est le gradient conjugué propose de générer les directions $P(k)$ selon l'algorithme suivant :

$$x(0) = x_0$$

Pour $k=1,2,\dots$ jusqu'à ce que $|x^{(k+1)}-x^{(k)}| < \varepsilon$

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

$$\text{avec : } P(k) = -\nabla_x f(x(k)) + \beta(k) P(k-1)$$

$$\text{où : } \beta(0) = 0$$

$$\text{et } \beta(k) = \frac{\|\nabla_x f(x(k))\|^2}{\|\nabla_x f(x(k-1))\|^2}$$

2.3. Méthode de Newton

La méthode de Newton repose sur le principe qui consiste à minimiser localement la fonction définie par le développement limité au second ordre de la fonction objectif $f(x)$.

L'algorithme de Newton est le suivant :

$$x(0) = x_0$$

Pour $k=1,2,\dots$ jusqu' à ce que $|x(k+1)-x(k)| < \varepsilon$

$$x(k+1) = x(k) - [\nabla_x^2 f(x(k))]^{-1} \nabla_x f(x(k))$$

2.4.Méthode de Newton modifiée

Lorsque les matrices $\nabla_x^2 f(x(k)), \forall k$, sont définies positives on peut considérer la méthode de Newton comme une méthode de descente avec un pas 1. On peut cependant envisager dans ce cas des pas de descente $\alpha(k) > 0$ différent de 1 pour améliorer la décroissance de f . (dans ce cas l'algorithme de Newton est modifié est on l'appelle méthode de Newton modifiée). L'algorithme de cette nouvelle méthode est le suivant :

$$x(0) = x_0$$

Pour $k=1,2,\dots$ jusqu' à ce que $|x(k+1)-x(k)| < \varepsilon$

$$x(k+1) = x(k) - \alpha(k) [\nabla_x^2 f(x(k))]^{-1} \nabla_x f(x(k))$$

$\alpha(k)$ est choisit tel que $f(x(k+1)) < f(x(k))$.

Exemple : minimiser la fonction $x_1^2 + x_2^2$ avec $x(0) = [1 \ 1]$, en utilisant l'algorithme du gradient.

$$\text{Le gradient est : } \nabla_x f(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow x_1 = 0 \text{ et } x_2 = 0$$

$$\text{La matrice Hessienne est : } \nabla_x^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad |2| > 0 \text{ et } \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0 \Rightarrow \text{la matrice}$$

Hessienne est définie positive. Alors le point critique $(0,0)$ est un minimum.

- Calcul du pas de descente

$$f(x(k+1)) < f(x(k)) \Rightarrow [x_1(k+1)]^2 + [x_2(k+1)]^2 < [x_1(k)]^2 + [x_2(k)]^2 \quad (1)$$

En utilisant l'algorithme du gradient on va avoir :

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} - \alpha(k) \begin{bmatrix} 2x_1(k) \\ 2x_2(k) \end{bmatrix} \Rightarrow \begin{cases} x_1(k+1) = x_1(k) - 2\alpha(k)x_1(k) \\ x_2(k+1) = x_2(k) - 2\alpha(k)x_2(k) \end{cases} \quad (2)$$

En remplaçant la relation (2) dans la relation (1), on obtient :

$$[x_1(k) - 2\alpha(k)x_1(k)]^2 + [x_2(k) - 2\alpha(k)x_2(k)]^2 < [x_1(k)]^2 + [x_2(k)]^2 \quad (3)$$

$$\Rightarrow 2\alpha(k)(\alpha(k) - 1)([x_1(k)]^2 + [x_2(k)]^2) < 0 \quad (4)$$

Comme résultat final, on obtient $0 < \alpha(k) < 1$ (c'est le pas de descente)

Si on choisit aléatoirement le pas fixe $\alpha(k) = 0.25$

$$\text{Première itération : } \begin{cases} x_1(1) = x_1(0) - 2\alpha(0)x_1(0) = 1 - 2 * 0.25 * 1 = 0.5 \\ x_2(1) = x_2(0) - 2\alpha(0)x_2(0) = 1 - 2 * 0.25 * 1 = 0.5 \end{cases}$$

$$\text{Deuxième itération : } \begin{cases} x_1(2) = x_1(1) - 2\alpha(1)x_1(1) = 0.5 - 2 * 0.25 * 0.5 = 0.25 \\ x_2(2) = x_2(1) - 2\alpha(1)x_2(1) = 0.5 - 2 * 0.25 * 0.5 = 0.25 \end{cases}$$

$$\text{Troisième itération : } \begin{cases} x_1(3) = x_1(2) - 2\alpha(2)x_1(2) = 0.25 - 2 * 0.25 * 0.25 = 0.125 \\ x_2(3) = x_2(2) - 2\alpha(2)x_2(2) = 0.25 - 2 * 0.25 * 0.25 = 0.125 \end{cases}$$

$$\text{Quatrième itération : } \begin{cases} x_1(4) = x_1(3) - 2\alpha(3)x_1(3) = 0.125 - 2 * 0.25 * 0.125 = 0.0625 \\ x_2(4) = x_2(3) - 2\alpha(3)x_2(3) = 0.125 - 2 * 0.25 * 0.125 = 0.0625 \end{cases}$$

On remarque qu'après quatre itérations l'algorithme n'a pas convergé vers la solution analytique.

Calcul du pas de descente optimal : **minimiser** ($f(x(k+1))$)

$$\alpha(k)$$

Cela revient à **minimiser** $[x_1(k+1)]^2 + [x_2(k+1)]^2$

$$\alpha(k)$$

Minimiser ($[x_1(k) - 2\alpha(k)x_1(k)]^2 + [x_2(k) - 2\alpha(k)x_2(k)]^2$)

$$\alpha(k)$$

Le gradient $\nabla_{\alpha} f(x(k+1)) = 4(2\alpha(k)-1) ([x_1(k)]^2 + [x_2(k)]^2)$

$$\nabla_{\alpha} f(x(k+1)) = 0 \Rightarrow \alpha(k) = 0.5$$

Si on utilise le pas optimal pour rechercher le minimum avec l'algorithme du gradient, on va avoir :

$$\text{Pour la Première itération : } \begin{cases} x_1(1) = x_1(0) - 2\alpha(0)x_1(0) = 1 - 2 * 0.5 * 1 = 0 \\ x_2(1) = x_2(0) - 2\alpha(0)x_2(0) = 1 - 2 * 0.5 * 1 = 0 \end{cases}$$

$$\text{Deuxième itération : } \begin{cases} x_1(2) = x_1(1) - 2\alpha(1)x_1(1) = 0 - 2 * 0.5 * 0.5 = 0 \\ x_2(2) = x_2(1) - 2\alpha(1)x_2(1) = 0 - 2 * 0.5 * 0.5 = 0 \end{cases}$$

Comme la solution se stabilise à (0,0) pour la première et deuxième itération.

Ceci signifie que l'algorithme à converger rapidement vers la solution analytique, en utilisant le pas optimal.

En conclusion on peut dire que le choix du pas de descente influe sur le nombre des itérations à effectuer pour atteindre l'optimum.