

III.1. Introduction

L'optimisation et particulièrement l'optimisation numérique a connu un départ important ces dernières années avec l'apparition de l'ordinateur. Elle est souvent la dernière étape de l'analyse numérique où, après avoir étudié un phénomène physique, l'avoir mis en équation, avoir étudié ces équations et avoir montré que l'on pouvait calculer les solutions avec un ordinateur, on commence à optimiser le système en changeant certains paramètres pour changer la solution dans un sens désiré.

III.2. Définition et formulation

On distingue trois étapes pour la modélisation mathématique de problème d'optimisation :

- Identification des variables de décisions : ce sont les paramètres sur lesquels l'utilisateur peut agir pour faire évoluer le système considéré.
- Définition d'une fonction coût ou fonction objectif : permettant d'évaluer l'état du système (ex : rendement, performance,...).
- Description des contraintes imposées aux variables de décision.

Le problème d'optimisation consiste alors à déterminer les variables de décision conduisant aux meilleures conditions de fonctionnement du système (ce qui revient à minimiser ou maximiser la fonction coût), tout en respectant les contraintes d'utilisation définies à l'étape 3.

III.2.1. Notions de minimum, maximum, infimum, supremum

On classera les notions de minimum et de maximum des notions d'infimum et de supremum.

Ces notions sont des prérequis pour les démonstrations des résultats d'existence et d'unicité d'extrema d'une fonction donnée.

- **Minorant / Majorant** : Soit E un sous-ensemble de $\mathbb{R}$, $m \in \mathbb{R} \cup \{-\infty, +\infty\}$ est un minorant de E si m est inférieur ou égal à tous les éléments de E tandis que $M \in \mathbb{R} \cup \{-\infty, +\infty\}$ est un majorant de E si il est supérieur ou égal à tous les éléments de E . Ainsi :

(m minorant $\Leftrightarrow \forall x \in E, m \leq x$) et (M majorant $\Leftrightarrow \forall x \in E, M \geq x$):

Si E admet un minorant (respectivement. majorant) fini alors il est dit minoré (respectivement. majoré).

- **Infimum / Supremum** : Soit $E \subset \mathbb{R}$. L'infimum $\inf(E) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de E est le plus grand des minorants. Le supremum $\sup(E) \in \mathbb{R} \cup \{-\infty, +\infty\}$ de E est le plus petit des majorants.

On les note respectivement :

$$\inf(E) = \inf_{x \in E}(x) \quad \text{et} \quad \sup(E) = \sup_{x \in E}(x)$$

Nous avons parlé d'infimum et de supremum, nous les relierons maintenant aux définitions classiques de minimum et de maximum.

- **Minimum / maximum :** Soit $E \subset \mathbb{R}$. L'infimum de E est appelé minimum si $\inf(E) \in E$.

Le supremum de E est appelé maximum si $\sup(E) \in E$. Dans ce cas, on les note respectivement $\min(E)$ et $\max(E)$.

Exemple :

Si on se donne $E =]0, 1]$, alors l'ensemble des minorants de E est $[-\infty, 0]$ et l'ensemble des majorants est $[1, +\infty]$. On en déduit que le l'infimum vaut 0 et le supremum vaut 1. Comme 1 appartient à E alors c'est aussi le maximum de E . Cependant 0 n'appartient pas à E et donc E n'a pas de minimum. On dit souvent que l'infimum de E n'est pas atteint.

La formulation et l'analyse d'un problème d'optimisation consiste à chercher la minimisation (ou maximisation) d'une fonction objectif (de coût).

Cette fonction comporte des paramètres et est généralement soumise à des contraintes en fonction des caractéristiques des problèmes (paramètres fortement variables, domaines discrets, types de contraintes, etc.), le problème peut être difficile à résoudre et nécessite des outils avancés pour la modélisation et la résolution comme :

- Variables (paramètres)
- Fonction objectif
- Contraintes

III.2.2. Description d'un problème d'optimisation

Comme nous l'avons vu en introduction, tous les problèmes d'optimisation que nous considérerons peuvent être exprimés de la façon suivante :

$$\min_{x \in \mathbb{R}^n} f(x)$$

Sous la contrainte : $x \in X$:

Où X est un sous-ensemble de $\mathbb{R}^n$. On pourra écrire aussi

$$\min_{x \in X} f(x)$$

Les variables $x = (x_1, \dots, x_n)$ sont appelées "variables d'optimisation" ou variables de décision (sont appelés également points admissibles).

La fonction $f : X \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée fonction coût, objectif ou critère et l'ensemble X est appelé ensemble des contraintes ou domaine des contraintes.

Chercher une solution du problème avec des contraintes, revient à chercher un point de minimum local de f dans l'ensemble des points admissibles, au sens de la définition suivante :

➤ $x \in \mathbb{R}^n$ est un point de minimum local de f sur $X \subset \mathbb{R}^n$ si : $x \in X$

Et

$$\exists r > 0 \mid \forall y \in X \cap B(x, r), \quad f(x) \leq f(y)$$

→ On dit alors que $f(x)$ est un minimum local de f sur X .

➤ $x \in \mathbb{R}^n$ est un point de minimum global de f sur X si : $x \in X$ et

$$\forall y \in X, \quad f(x) \leq f(y)$$

→ On dit alors que $f(x)$ est un minimum global de f sur X .

Les notions de maximum local et global sont définies de façon tout à fait analogue. En fait, on peut facilement démontrer que les problèmes (avec ou sans contraintes) :

$$\min_{x \in X} f(x) \quad \text{et} \quad \max_{x \in X} -f(x)$$

Sont équivalents dans le sens où ils ont même ensemble de solutions et :

$$\min_{x \in X} f(x) = - \max_{x \in X} -f(x)$$

Ou encore :

$$\max_{x \in X} f(x) = - \min_{x \in X} -f(x)$$

Ainsi la recherche d'un maximum pouvant se ramener à la recherche d'un minimum, nous supporterons une attention plus particulière à la recherche du minimum.

III.2.3. Types d'optimisation

Les problèmes d'optimisation peuvent être classés en plusieurs grandes familles :

- Optimisation numérique : $X \subset \mathbb{R}^n$.
- Optimisation discrète (ou combinatoire) : X fini ou mesurable.
- Commande optimale : X est un ensemble de fonctions.
- Optimisation stochastique : données aléatoires (à ne pas confondre avec les méthodes stochastiques d'optimisation).
- Optimisation multicritères : plusieurs fonctions objectives.

III.2.4. Algorithme d'optimisation

Un algorithme d'optimisation est aussi jugé sur les propriétés (théoriques) de $(x_k)_k$, en particulier propriété de convergence :

- Convergence globale : pour tout point initial $x_0 \in \mathbb{R}^n$.
- Convergence locale.

Nous nous intéressons dans cette partie à la conception de méthodes numériques pour la recherche des points $x \in \mathbb{R}^n$ qui réalisent le minimum d'une fonction f : où f est une fonction définie sur $\mathbb{R}^n$ à valeurs réelles supposée différentiable, voire même deux fois différentiable. Les conditions nécessaires d'optimalité du premier et du second ordre expriment le fait qu'il n'est pas possible de "descendre" à partir d'un point de minimum (local ou global). Cette observation va servir de point de départ à l'élaboration des méthodes dites de descente rappelées dans cette partie.

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ définie par :

$x_{k+1} = x_k + s_k d_k$ telle que :

$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k).$$

Un tel algorithme est ainsi déterminé par deux éléments : le choix de la direction d_k appelée direction de descente, et le choix de la taille du pas s_k à faire dans la direction d_k . Cette étape est appelée recherche linéaire.

Un vecteur $d \in \mathbb{R}^n$ est une direction de descente pour f à partir d'un point $x \in \mathbb{R}^n$ si :

$t \rightarrow f(x + t_d)$ est décroissante en $t = 0$, c'est-à-dire s'il existe $\eta > 0$ tel que :

$\forall t \in]0, \eta], f(x + td) < f(x)$.

Il est donc important d'analyser le comportement de la fonction f dans certaines directions.

Lorsqu'elle existe, la dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d , en déduisant donc :

- Le vecteur $d \in \mathbb{R}^n$ est une direction de descente de f au point $x \in \mathbb{R}^n$ si $f'(x; d) < 0$.
- Soit $f: \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable. Soit $x \in \mathbb{R}^n$. Alors pour toute direction d de norme constante égale à :

$$\|d\| = \|\nabla f(x)\|$$

On a :

$$(-\nabla f(x))^T \nabla f(x) \leq d^T \nabla f(x)$$

Ainsi, la direction d^* est appelée "direction de plus forte descente".

$$d^* = -\nabla f(x)$$

D'après la caractérisation de la descente, il s'agit donc à chaque itération k , de trouver un point x_{k+1} dans une direction d vérifiant :

$$\nabla f(x_k)^T d < 0.$$

- Algorithme de descente Modèle

Données: $f: \mathbb{R}^n \rightarrow \mathbb{R}$ différentiable, x_0 point initial arbitraire.

Sortie: une approximation de la solution du problème : $\min_{x \in \mathbb{R}^n} f(x)$.

1. $k := 0$

2. Tant que "test d'arrêt" non satisfait,

(a) Trouver une direction de descente d_k telle que : $\nabla f(x_k)^T d_k < 0$.

(b) *Recherche linéaire* : Choisir un pas $s_k > 0$ à faire dans cette direction et tel que :

$$f(x_k + s_k d_k) < f(x_k).$$

(c) Mise à jour : $x_{k+1} = x_k + s_k d_k$; $k := k + 1$;

3. Retourner x_k .

- Structure d'un algorithme d'optimisation

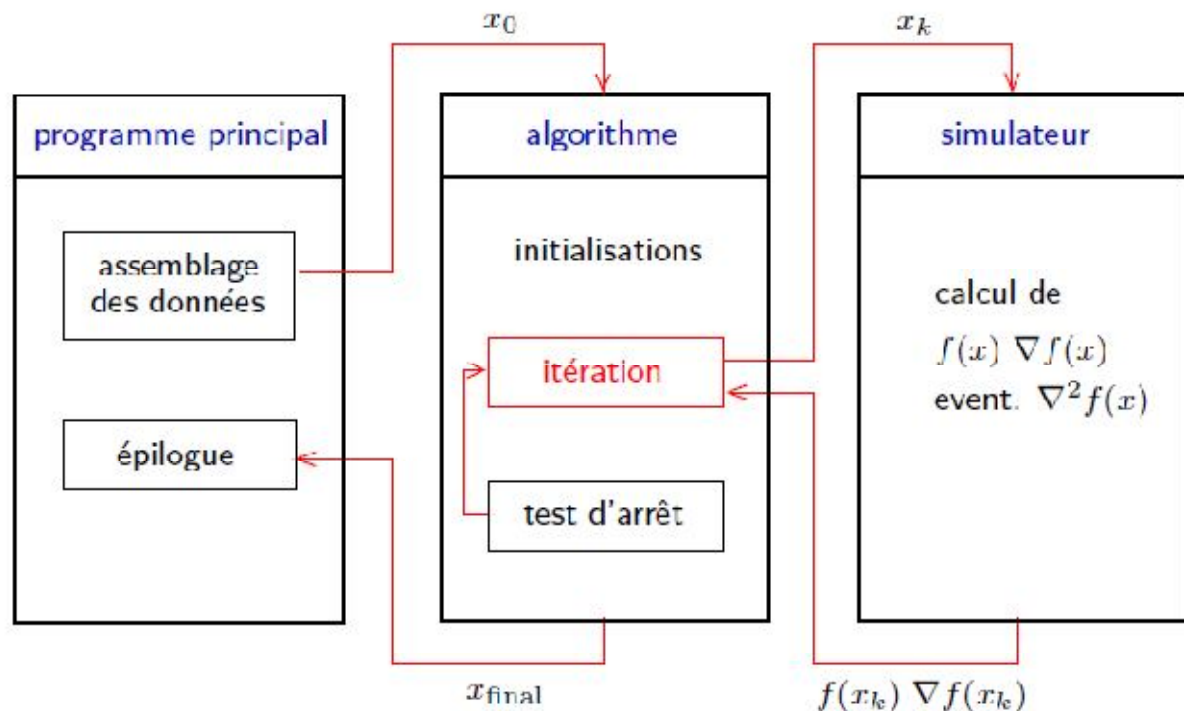


Fig. 1. Structure d'un algorithme d'optimisation

III.3. Optimisation sans contraintes

On s'intéresse ici aux algorithmes de descente qui sont déterminés par les stratégies de choix des directions de descente successives, puis par le pas qui sera effectué dans la direction choisie. Concentrons-nous également, dans cette partie sur le choix de la direction de descente : l'idée est de remplacer f par un modèle local plus simple, dont la minimisation nous donnera une direction de descente de f . On distingue donc :

➤ Algorithmes de type gradient

Données: f, x_0 première approximation de la solution cherchée, $\varepsilon > 0$ précision demandée

Sortie: une approximation x^* de la solution de : $\nabla f(x) = 0$

1. $k := 0$;
2. Tant que critère d'arrêt non satisfait,
 - (a) *Direction de descente* : $d_k = -\nabla f(x_k)$.
 - (b) *Recherche linéaire* : trouver un pas s_k tel que : $f(x_k + s_k d_k) < f(x_k)$.
 - (c) $x_{k+1} = x_k - s_k \nabla f(x_k)$; $k := k + 1$;
3. Retourner x_k .

➤ Algorithmes de type Newton locale

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$ de classe C^2 , x_0 première approximation de la solution cherchée, $\varepsilon > 0$ précision demandée

Sortie: une approximation x^* de la solution

1. $k := 0$;
2. Tant que $\|\nabla f(x_k)\| > \varepsilon$,
 - (a) Calculer d_k solution du système : $H[f](x_k)d_k = -\nabla f(x_k)$;
 - (b) $x_{k+1} = x_k + d_k$;
 - (c) $k := k + 1$;
3. Retourner x_k ;

➤ Algorithmes de type Gauss-Newton

Données: F fonction différentiable, x_0 point initial, $\varepsilon > 0$ précision demandée

Sortie: une approximation de la solution du problème de moindres carrés :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} F(x)^\top F(x).$$

1. $k := 0$;
2. Tant que ,
 - (a) *Calcul d'une direction de recherche* : calculer d_{k+1} solution de :

$$J_F(x_k)^\top J_F(x_k) d = -J_F(x_k)^\top F(x_k).$$

- (b) $x_{k+1} = x_k + d_{k+1}$;
 - (c) $k := k + 1$;
3. Retourner s_k .

➤ Algorithmes de type Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt peut être vu comme une régularisation de l'algorithme de Gauss-Newton, en particulier lorsque la jacobienne de f n'est pas de rang plein.

D'où :

Données: F fonction différentiable, x_0 point initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation de la solution du problème de moindres carrés :

$$\min_{x \in \mathbb{R}^n} r(x) = \frac{1}{2} F(x)^T F(x).$$

1. $k := 0$;
2. Tant que *critère d'arrêt à définir*,
 - (a) *Calcul d'une direction de recherche* : calculer d_k solution de :

$$(J_F(x_k)^T J_F(x_k) + \lambda I) d = -J_F(x_k)^T F(x_k).$$
 - (b) $x_{k+1} = x_k + d_k$;
 - (c) Mise à jour du paramètre λ .
 - (d) $k := k + 1$;
3. Retourner x_k .

III.4. Optimisation sous contraintes

Cette partie est une courte introduction à l'optimisation sous contraintes. On s'intéresse à la résolution de problèmes d'optimisation de la forme :

$$\min f(x)$$

Où X est un sous-ensemble non vide de $\mathbb{R}^n$ défini par des contraintes d'égalité ou d'inégalités de fonctions :

$$X = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$$

Où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$ sont continues. Ici les écritures $h(x) = 0$ et $g(x) \leq 0$ signifient :

$$\begin{cases} \forall i = 1, \dots, p, h_i(x) = 0 \\ \forall j = 1, \dots, q, g_j(x) \leq 0 \end{cases}$$

L'ensemble X est appelé ensemble ou domaine des contraintes. Tout point $x \in \mathbb{R}^n$ vérifiant :
 $x \in X$, est appelé point admissible du problème d'optimisation.

D'après les résultats précédents, rappelons que si g et h sont continues alors X est un ensemble fermé (mais non nécessairement borné). Dans la suite nous travaillerons toujours avec un problème sous "forme standard" c'est-à-dire un problème écrit sous la forme suivante :

$$(P) \left| \begin{array}{l} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.c.} \quad h_i(x) = 0, \quad i = 1, \dots, p \\ \quad \quad g_j(x) \leq 0, \quad j = 1, \dots, q. \end{array} \right.$$

Par exemple, la forme standard du problème :

$$\max_{(x,y) \in \mathbb{R}^2} f(x,y) \text{ s.t. : } x^2 + y^2 \geq 1 \text{ et } x + y = 5,$$

Est :

$$\min_{(x,y) \in \mathbb{R}^2} -f(x,y) \text{ s.t. : } 1 - x^2 - y^2 \leq 0 \text{ et } x + y - 5 = 0.$$

Le problème (P), écrit sous forme standard, est dit convexe si h est affine, g convexe et si la fonction objectif f est convexe sur X

Le but de cette section est d'élaborer un algorithme de résolution du problème :

Minimiser $f(x)$, $x \in \mathbb{R}^n$, sous la contrainte : $x \in X$,

On distingue donc :

➤ Algorithmes du gradient projeté

Données: f , p_X l'opérateur de projection sur X , x_0 un point initial et $\varepsilon > 0$ la précision demandée

Sortie: une approximation x^* de la solution

1. $k := 0$;
2. Tant que critère d'arrêt non satisfait,
 - (a) $y_k = x_k - s \nabla f(x_k)$ où s est le pas calculé par la méthode de gradient choisie ;
 - (b) Projection sur X : $x_{k+1} = p_X(y_k)$; $k := k + 1$;
3. Retourner x_k .

Il est important de remarquer que le calcul à l'étape 2(b) du projeté sur X , peut parfois être aussi difficile que le problème initial. En effet y_k est obtenu en résolvant le problème :

$$\min_{y \in \mathbb{R}^n} \frac{1}{2} \|x_k - s \nabla f(x_k) - y\|_2^2$$

Il s'agit donc de résoudre un problème d'optimisation sur un convexe, avec une fonction objectif convexe. Lorsque le domaine X des contraintes est simple (contraintes de bornes en particulier), c'est faisable. Dès que les contraintes ne sont pas des contraintes de bornes, le calcul de la projection devient beaucoup plus délicat.

Vérifions que la direction $d_k = x_{k+1} - x_k$, si elle est non nulle, est bien une direction de descente de f en x_k .

➤ Algorithmes newtoniens – Méthode SQP avec contraintes d'égalité.

Comme son nom l'indique, la méthode SQP (Sequential Quadratic Programming) consiste à remplacer le problème initial par une suite de problèmes quadratiques sous contraintes linéaires plus faciles à résoudre. L'algorithme est le suivant :

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ différentiables, x_0 point initial, $\lambda_0 \in \mathbb{R}^p$ multiplicateur initial, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
2. Tant que $\|\nabla L(x^k; \lambda^k)\| > \varepsilon$,
 - (a) Résoudre le sous-problème quadratique :

$$(QP_k) \quad \begin{cases} \min_{d \in \mathbb{R}^n} & \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d \\ \text{s.t.} & h_i(x^k) + \nabla h_i(x^k)^\top d = 0. \end{cases}$$

et obtenir la solution primale d_k et le multiplicateur λ' associé à la contrainte d'égalité.

- (b) $x^{k+1} = x_k + d_k$; $\lambda^{k+1} = \lambda'$; $k = k + 1$;
3. Retourner x_k .

➤ Algorithmes newtoniens – Méthode SQP avec contraintes d'égalité et d'inégalité.

Données: $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^q$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ différentiables, x_0 point initial, $\lambda_0 \in \mathbb{R}_+^q$ et $\mu_0 \in \mathbb{R}^p$ multiplicateurs initiaux, $\varepsilon > 0$ précision demandée.

Sortie: une approximation x^* de la solution.

1. $k := 0$;
2. Tant que $\|\nabla L(x^k; \lambda^k)\| > \varepsilon$,
 - (a) Résoudre le sous-problème quadratique :

$$(QP_k) \quad \begin{cases} \min_{d \in \mathbb{R}^n} & \nabla f(x_k)^\top d + \frac{1}{2} d^\top H_k d \\ \text{s.t.} & g_j(x^k) + \nabla g_j(x^k)^\top d = 0, \quad j = 1, \dots, q, \\ & h_i(x^k) + \nabla h_i(x^k)^\top d = 0, \quad i = 1, \dots, p. \end{cases}$$

et obtenir la solution primale d_k et les multiplicateurs λ' et μ' associés aux contraintes d'inégalité et d'égalité respectivement.

- (b) $x^{k+1} = x_k + d_k$; $\lambda^{k+1} = \lambda'$; $\mu^{k+1} = \mu'$; $k = k + 1$;
3. Retourner x_k .

III.5. Conclusion

L'optimisation différentiable sans contraintes nous apprend que :

- Les algorithmes utilisent une information locale - mais on les globalise
- Les méthodes efficaces exigent d'utiliser de l'information du 2nd ordre

Mêmes notions et les mêmes techniques de base en :

- Optimisation avec contraintes
- Optimisation non-différentiable

Choix d'une méthode :

- dépend de la structure de la fonction – mais souvent plusieurs méthodes sont disponibles
- Critères : taille des problèmes, contraintes pratiques et surtout temps de calcul de $f(x)$, $\Delta f(x)$ et $\Delta^2 f(x)$.