Chapitre 5

OPTIMISATION NON DIFFERENTIABLE

5.1 Introduction

On rencontre de nombreuses situations en Recherche Opérationnelle dans lesquelles on doit optimiser une fonction non partout différentiable. En général, il s'agit de minimiser une fonction convexe de type 'max' ou de maximiser une fonction concave de type 'min'. Les techniques de Programmation Non Linéaire pour les problèmes d'optimisation différentiable sont alors inopérantes et des méthodes adaptées doivent être mises au point pour ces situations. Toutefois, l'exemple des fonctions 'max' nous montre que, d'une certaine manière, minimiser une fonction non différentiable peut se ramener à un probème différentiable, mais sous contraintes.

Soit $f(x) = \max\{\langle a_i, x\rangle + b_i, i = 1, \ldots, p\}$ une fonction convexe, affine par morceaux. Cette fonction est clairement non différentiable aux points x tels que $I(x) = \{i \in \{1, \ldots, p\} \mid f(x) = \langle a_i, x\rangle + b_i\}$ ne se réduit pas à un singleton. Mais il est clair également que minimiser cette fonction dans $I\!R^n$ est équivalent à résoudre le problème suivant dans $I\!R^{n+1}$:

$$\begin{array}{ll} \text{Minimiser} & z \\ \text{sous} & z - \langle a_i, x\rangle - b_i \geq 0,\ i = 1, \ldots, p \end{array}$$

Symétriquement, certains problèmes de satisfaction de contraintes peuvent se ramener à des problèmes d'optimisation non différentiable :

Considérons par exemple la recherche d'un point satisfaisant les contraintes $f_i(x) \leq 0, i = 1, \ldots, p$, où les f_i sont des fonctions convexes différentiables. Ce problème est équivalent à minimiser la fonction convexe non différentiable $f(x) = \max\{0, f_i(x), i = 1, \ldots, p\}$.

Observons enfin qu'une discontinuité du gradient d'une fonction Lipschitz continue peut s'assimiler à une situation limite où une fonction différentiable présente des variations très brutales de certaines dérivées directionnelles. Donc, d'une part, toute fonction Lipschitz continue peut être approchée par une fonction différentiable et, d'autre part, les méthodes pour l'optimisation non différentiable étudiées dans ce chapitre pourront bénéficier aux fonctions différentiables dont le gradient est mal conditionné.

5.2 Méthodes de sous-gradient

Les méthodes de sous-gradients sont proposées pour maximiser des fonctions concaves (ou minimiser des fonctions convexes) non nécessairement différentiables pour lesquelles il est relativement aisé de déterminer un sous-gradient en un point. C'est le cas typiquement des fonctions duales rencontrées au chapitre précédent. La méthode générique initialement étudiée par l'école de Kiev dans les années 60 consiste à effectuer des petits pas dans la direction du sous-gradient :

Algorithme (SG)

Soit h une fonction concave, scs. On définit la suite de points $\{u^k\}$ suivante :

$$u^{k+1} = u^k + \lambda_k \frac{g^k}{\|g^k\|}$$

où $g^k \in \partial h(u^k)$ et $\lambda_k > 0$.

Dans un premier temps, on remarque que la direction g^k n'est pas toujours une direction de montée pour la fonction concave h. On veut dire par là que, contrairement au cas différentiable, la dérivée directionnelle $h'(u^k; g^k)$ peut tre négative en certains points où h n'est pas différentiable.

Exemple : Dans $I\!R^2$, $h(u) = -|u_1| - 4|u_2|$

h peut s'écrire $h(u) = \min\{u_1 + 4u_2, u_1 - 4u_2, -u_1 + 4u_2, -u_1 - 4u_2\}$ ou $h(u) = \min\{h_i;\ i = 1, \ldots, 4\}$. On constate sur la figure 5.1 où la courbe de niveau $h(u) = -4$ est représentée que le sous-gradient $g^4 = \nabla h_4(u) = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$ au point $u = \begin{pmatrix} -4 \\ 0 \end{pmatrix}$ n'est pas une direction de montée.

Remarques :

1. Il est numériquement improbable que l'algorithme (SG) génère des points de non différentiabilité (h est différentiable presque partout). Toutefois, les points de la suite $\{u^k\}$ tendent asymptotiquement vers les surfaces de non différentiabilité et on a, si g^k est le gradient de h en u^k, $g^k \in \partial h(u)$ où u est un point de non différentiabilité de h arbitrairement proche de u^k quand $k \to +\infty$.
2. Dans la plupart des situations, le point u^* qui maximise h, quand il existe, est un point de non différentiabilité. Dans ces conditions, la limite du sous-gradient effectivement calculé g^k n'est pas nulle et on ne peut tester la condition d'optimalité. La suite des u^k présente un comportement en zig-zags et peut devenir arbitrairement lente quand l'angle α associé aux courbes de niveau tend vers zéro (figure 5.2).

On va démontrer maintenant que g^k est une direction de descente pour la fonction distance à l'optimum. Comme g^k est un sous-gradient de h en u^k, on a :

$$h(u) \leq h(u^k) + \langle g^k, u - u^k \rangle$$

Supposons qu'il existe u^* qui maximise hh. Alors :

$$0 < h(u^*) - h(u^k) \leq \langle g^k, u^* - u^k \rangle$$

Figure 5.1 –

Figure 5.2 –

FIGURE 5.3 – Pas optimal

Cette dernière relation peut s'écrire, si $d(u) = 1/2\|u^* - u\|^2$, $d'(u^k; g^k) < 0$.

Un premier résultat de convergence pour la suite générée par (SG) a été obtenu par Polyak :

Théorème 5.1 *Supposons que l'ensemble des maxima de h, H^*, est borné. Alors, si la suite des pas $\{\lambda_k\}$ associée à l'algorithme (SG) satisfait :*

$$\lambda_k \to 0 \text{ et } \sum_k \lambda_k \to +\infty$$

la suite $\{u^k\}$ converge vers un point de H^*.

Démonstration : cf. Shor , Minimization methods for non-differentiable functions, Springer V. 1985, p. 25.

Il suffit donc d'imposer a priori à la suite $\{\lambda_k\}$ la condition ci-dessus, dite de la série divergente, pour garantir la convergence (par exemple $\lambda_k = 1/k$). L'inconvénient principal est que cette condition implique une convergence sous-linéaire des itérés vers une solution. En pratique, on implémente l'algorithme (SG) avec une série $\{\lambda_k\}$ convergente qui permet d'obtenir une convergence linéaire. Cette convergence est d'autant meilleure que l'angle α , entre le sous-gradient et la direction optimale, est plus petit.

On appelle *condition* de la fonction h le nombre ξ égal au cosinus du plus grand angle α (donc le pire des cas), quand u parcourt l'espace, entre la direction du sous-gradient $g(u)$ et la direction de l'optimum $u^* - u$:

$$\xi = \inf_u \frac{\langle u^* - u, g(u)\rangle}{\|u^* - u\|.\|g(u)\|}$$

Plus cette condition est proche de 0, plus la fonction est mal conditionnée et la convergence lente.

Les choix les plus courants pour le pas λ_k sont décrits ci-dessous :

a) Série convergente de Shor

Soit d_k la distance à l'optimum en u^k. Le pas optimal correspond à projeter u^* sur la direction du sous-gradient g^k. On a alors $d_{k+1} = d_k sin\alpha_k$, où α_k est l'angle entre g^k et la direction optimale (fig. 5.3). Si on dispose d'une estimation ζ de la condition de la fonction h, on peut alors choisir :

$$\lambda_k = \lambda_0 \rho^k \qquad \textbf{(SH)}$$

avec $\lambda_0 = d_0\zeta$ où d_0 est une estimation de la distance initiale à l'optimum et $\rho = (1 - \zeta^2)^{1/2}$.

b) Relaxation de Held et Karp (Mathematical Programming, vol. 1, 1971)

Dans cet article, Held et Karp ont proposé une relaxation lagrangienne du problème du voyageur de commerce avec une méthode de sous-gradient où le pas correspond à :

$$\lambda_k = \omega_k \frac{\bar{h} - h(u^k)}{\|g^k\|} \qquad \textbf{(HK)}$$

où $0 < \omega_k < 2$ et $\bar{h}$ est une estimation par défaut de la valeur optimale $h(u^*)$

FIGURE 5.4 – Projections successives

Pour illustrer ce choix, reprenons le problème de la recherche d'une solution d'un système d'inégalités linéaires vu dans l'introduction :

Trouver $x \in \Omega = \{x \in \mathbb{R}^n \mid \langle_a i, x\rangle \leq b_i, i = 1, \ldots, m\}$

Soit la fonction de $\mathbb{R}^n$: $f(x) = \max\{0, \langle a_i, x\rangle - b_i, i = 1, \ldots, m\}$. f est convexe, s.c.i., telle que $f(x) \geq 0$ et on a : $\Omega \neq \emptyset \Longleftrightarrow \min f(x) = 0$.

On pose $a_0 \in \mathbb{R}^n, a_0 = 0$ et $b_0 = 0$. Soit $I(x) = \{i \in \{0,_l dots, m\} \mid f(x) = \langle a_i, x\rangle - bi\}$. Il est clair que f est différentiable si $I(x)$ est un singleton et on a (cf. Chap. II-3) :

$$\partial f(x) = \mathrm{Conv}\{a_i, i \in I(x)\}$$

L'algorithme, dit des projections successives, est en fait une méthode de sous-gradient pour minimiser f. Il est défini par :

$$x^{k+1} = x^k - \lambda_k \frac{a_{i_k}}{\|a_{i_k}\|}, \text{ où } a_{i_k} \in \partial f(x^k)$$

avec $\lambda_k = \omega_k \frac{\langle a_{i_k}, x^k\rangle - b_i}{\|a_{i_k}\|}$, ω_k est un paramètre de relaxation compris strictement entre 0 et 2.

Le choix de i_k correspond à la contrainte la plus violée en x^k sur laquelle on projète x^k pour obtenir x^{k+1} (ω_k garantit que la distance à Ω diminue à chaque itération).
Observation : Quand la valeur optimale de la fonction n'est pas connue, on peut vérifier que la mise à jour du pas donnée par (HK) fournit (dans le cas d'une maximisation) une borne inférieure du pas optimal associé à la projection de u^* sur la direction de g^k, car la concavité de h implique :

$$h(u^*) \leq h(u^k) + \langle g^k, u^* - u^k\rangle$$

En pratique, la condition est souvent proche de 1 et la convergence reste assez lente. Elle peut tre accélérée par un changement de métrique comme dans les méthodes de type Newton (cf. chap. I-3). On parle de dilatation de l'espace dans la littérature russe. La méthode la plus connue pour ses implications remarquables sur la complexité de la programmation linéaire est la méthode de l'ellipsode de Khachian (1979). C'est en effet la première méthode de complexité polynomiale proposée pour résoudre un programme linéaire.