

Programmation stochastique

Régularisation par région de confiance

Fabian Bastin

IFT-6512 – Hiver 2011

Régularisation au moyen de régions de confiance

Motivation : un petit détour par la programmation non-linéaire.

Considérons le problème générique, sans contraintes,

$$\min_x f(x),$$

où $f(x) \in C^2$.

Principe : à l'itération k , minimiser approximativement un modèle m_k de l'objectif dans une région \mathcal{B}_k .

Un classique : **modèle quadratique** (souvenir de Taylor ?)

$$m_k(x_k + s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T H_k s.$$

H_k : approximation de $\nabla^2 f(x_k)$.

Régions de confiance (2)

Nous devons donc résoudre le sous-problème :

$$\min_s m_k(\mathbf{x}_k + \mathbf{s}), \text{ tel que } \mathbf{x}_k + \mathbf{s} \in \mathcal{B}_k.$$

La solution représente l'**itéré candidat**.

Calculons le rapport suivant :

$$\rho_k = \frac{\text{réduction réelle}}{\text{réduction prédite}}$$

Soit des constantes η_1 et η_2 telles que $0 < \eta_1 \leq \eta_2 < 1$ (par exemple, $\eta_1 = 0.01$ et $\eta_2 = 0.75$).

- Si $\rho_k \geq \eta_1$, accepter le candidat.
- Si $\rho_k \geq \eta_2$, élargir sinon réduire ou garder identique \mathcal{B}_k .
- Si $\rho_k < \eta_1$, rejeter le candidat et réduire \mathcal{B}_k .

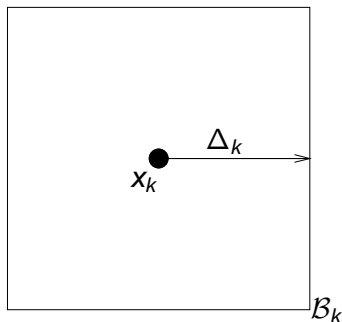
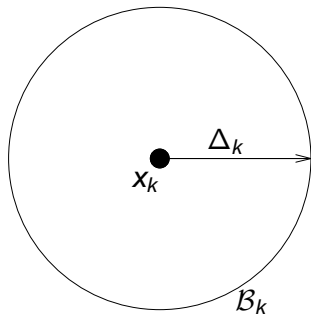
Arrêt lorsque certains critères sont rencontrés (p.e. norme du gradient relatif suffisamment petite).

Régions de confiance : normes

Le voisinage où nous considérons le modèle valide est défini mathématiquement par une boule centrée en l'itéré courant x_k , et de rayon Δ_k :

$$\mathcal{B}_k = \{x \mid \|x - x_k\|_k \leq \Delta_k\}.$$

Le choix de la norme peut être dépendant de l'itération k . Souvent, on choisit la norme 2 ou ∞ .



Soit x un vecteur de \mathbb{R}^n .

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_{(i)}^2}, \quad \|x\|_\infty = \max_{i=1, \dots, n} \{|x_{(i)}|\}, \quad \|x\|_1 = \sum_{i=1}^n |x_{(i)}|.$$

Dans le cadre de la régularisation par région de confiance, nous utiliserons la norme ℓ_∞ :

$$-\Delta_k \mathbf{e} \leq x - x_k \leq \Delta_k \mathbf{e},$$

où $\mathbf{e} = (1, 1, \dots, 1)^T$.

Pour la simplicité, nous supposerons que le recours est complet ; par conséquent, nous n'envisagerons pas les coupes de faisabilité.

Mise à jour de la région de confiance

A la fin de chaque itération k , on met à jour le rayon de la région de confiance suivant le schéma

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{si } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{si } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{si } \rho_k < \eta_1, \end{cases}$$

avec $0 < \eta_1 \leq \eta_2 < 1$ et $0 < \gamma_1 \leq \gamma_2 < 1$.

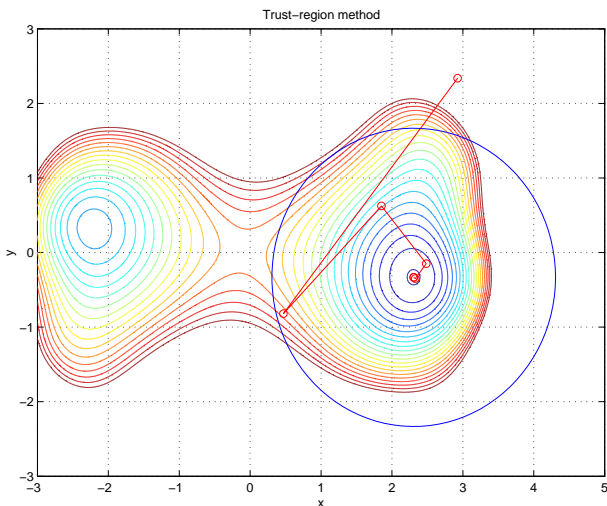
Remarque : des variantes existent, à la fois pour des soucis théoriques (convergence au second ordre), ou numériques, comme récemment l'approche rétrospective :

http://www.optimization-online.org/DB_HTML/2007/10/1821.html

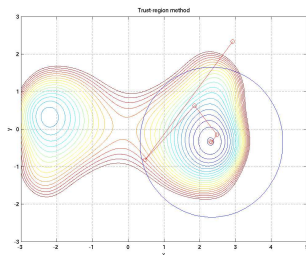
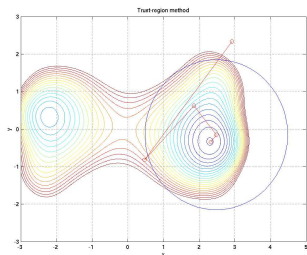
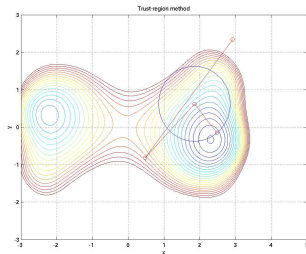
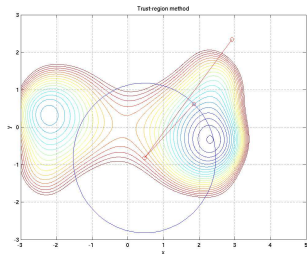
Le choix de Δ_0 reste délicat, et plusieurs heuristiques existent. Réf. : A. Sartenaer, *Automatic Determination Of An Initial Trust Region In Nonlinear Programming*, SIAM Journal of Scientific Computing 18(6), pp. 1788-1803, 1997.

Régions de confiance : exemple

$$\min_{x,y} -10x^2 + 10y^2 + 4 \sin(xy) - 2x + x^4$$



Régions de confiance : exemple (2)



La littérature sur les méthodes de régions de confiance est très riche. L'idée est en effet applicable à un grand nombre de programmes mathématiques, contraints et non contraints, moyennant quelques ajustements des détails internes.

Deux références majeures :

- 1 A. R. Conn, N. I. M. Gould, and Ph. L. Toint, Trust-Region Methods, SIAM, Philadelphia, USA, 2000.
- 2 J. Nocedal and S. J. Wright, Numerical Optimization, Springer, New York, USA, 1999.

Point de départ pour mieux comprendre : Chapitre 6 de Conn, Gould et Toint.

Simplification d'écriture

Jusqu'à présent, nous considérons le problème sous la forme

$$\min_{x \geq 0} c^T x + Q(x), \text{ t.q. } Ax = b,$$

où

$$Q(x) = E_{\xi}[Q(x, \xi)].$$

Or il est évident que

$$c^T x + E_{\xi}[Q(x, \xi)] = E_{\xi}[c^T x + Q(x, \xi)],$$

aussi il est fréquent d'exprimer les problèmes sous la forme

$$\min_{x \geq 0} Q(x), \text{ t.q. } Ax = b,$$

où

$$Q(x) = c^T x + E_{\xi}[Q(x, \xi)] = c^T x + \sum_{s=1}^S p_s Q_s(x),$$

$$Q_s(x) = \min_{y_s} q_s^T y_s, \text{ t.q. } W y_s = h_s - T_s x, y_s \geq 0.$$

Motivation : retour sur l'algorithme L-shaped

Nous considérons ici l'approche multi-coupe avec clusters : $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_C$, et notons

$$Q_{[j]}(x) = \sum_{i \in \mathcal{S}_j} p_i Q_i(x).$$

L'algorithme L-shaped construit un **modèle** $m_{[j]}^k$: borne inférieure, linéaire par morceaux, sur $Q_{[j]}$:

$$m_{[j]}^k(x) = \inf\{\theta_j \mid \theta_j \mathbf{e} \geq F_{[j]}^k x + f_{[j]}^k\}.$$

$F_{[j]}^{k,\ell}$ est une matrice dont les lignes sont des sous-gradients de $Q_{[j]}$ aux précédentes itérations. Rappel : sous-gradient de $Q_{[j]}$:

$$g_j = - \sum_{i \in [j]} p_i T_i^T \pi_i,$$

où les π_i sont les solutions duales des programmes de seconde étape.

Retour sur l'algorithme L -shaped (2)

A la k^e itération de la méthode L -shaped multi-coupes hybride, l'itéré x_{k+1} est obtenu en résolvant l'approximation

$$\min_x m_k(x), \text{ t.q. } Ax = b, x \geq 0,$$

où

$$m_k(x) \stackrel{\text{def}}{=} c^T x + \sum_{j=1}^C m_{[j]}^k(x),$$

soit

$$\begin{aligned} \min_{x, \theta_1, \dots, \theta_C} \quad & c^T x + \sum_{j=1}^C \theta_j \\ \text{t.q.} \quad & \theta_j e \geq F_{[j]}^{k, \ell} x + f_{[j]}^{k, \ell}, j = 1, 2, \dots, C, \\ & Ax = b, x \geq 0. \end{aligned}$$

Algorithm 1: L-shaped multi-coupes

- Etape 0** Choisir une tolérance ε_0 et un point initial x_0 . Définir un modèle initial m_0 , sous-estimé linéaire par morceaux de $Q(x)$, tel que $m_0(x_0) = Q(x_0)$, et m_0 est borné par le bas. $Q_{\min} \leftarrow Q(x_0)$, $k \leftarrow 0$.
- Etape 1** Obtenir x_{k+1} en minimisant $m_k(x)$. Si $Q_{\min} - m_k(x_{k+1}) \leq \varepsilon_{tol}(1 + |Q_{\min}|)$, arrêt.
- Etape 2** Evaluer la fonction et l'information de sous-gradient en x_{k+1} ; $Q_{\min} \leftarrow Q(x_{k+1})$. Obtenir m_{k+1} en ajoutant les coupes d'optimalité à m_k . $k \leftarrow k + 1$. Retour à l'étape 1.

Adéquation entre le modèle et la fonction objectif

Une hypothèse courante en régions de confiance est que pour l'itéré x_k , $m_k(x_k) = f(x_k)$, où $f(x_k)$ est la fonction à minimiser.

Dans notre cas, en ajoutant les coupes d'optimalité à m_k pour former m_{k+1} , on obtient $m_{k+1}(x_{k+1}) = Q(x_{k+1})$.

Rappel :

$$m_k(x) = c^T x + \sum_{j=1}^C m_{[j]}^k(x),$$
$$m_{[j]}^k(x) = \inf\{\theta_j \mid \theta_j e \geq F_{[j]}^k x + f_{[j]}^k\}.$$

Dès lors (en supposant qu'au moins une coupe existe),

$$m_{[j]}^k(x) = \max \left\{ F_{[j]}^k x + f_{[j]}^k \right\}.$$

Adéquation entre le modèle et la fonction objectif (2)

Si pour un certain j , aucune coupe n'est ajoutée, la contrainte est active : au point considéré, θ_j coïncide avec $Q_{[j]}$.

Sinon, nous avons une inégalité du type $\hat{\theta}_j < F_j \hat{x} + f_j$, où $(\hat{x}, \hat{\theta}_j)^T$ est une solution du programme maître (qui satisfait donc les précédentes coupes). La coupe à ajouter correspondant à l'inégalité du sous-gradient

$$\sum_{s \in \mathcal{S}_j} p_s Q(x, \xi_s) \geq \sum_{s \in \mathcal{S}_j} p_s \left(Q(\hat{x}, \xi_s) - \hat{\pi}_s T(\xi_s)^T (x - \hat{x}) \right), \quad \forall x \in K_2(\xi_s).$$

Aussi, introduire la nouvelle d'optimalité revient à exiger

$$\theta_j \geq \sum_{s \in \mathcal{S}_j} p_s \left(Q(\hat{x}, \xi_s) - \hat{\pi}_s T(\xi_s)^T (x - \hat{x}) \right), \quad \forall x \in K_2(\xi_s).$$

De par sa définition, $m_{[j]}^k(\hat{x}) = \sum_{s \in \mathcal{S}_j} p_s Q(\hat{x}, \xi_s) = Q_{[j]}(\hat{x})$.

Adéquation entre le modèle et la fonction objectif (3)

La description de l'algorithme nous donne que $\mathbf{x}_{k+1} := \hat{\mathbf{x}}$, et dès lors,

$$\begin{aligned} m_{k+1}(\mathbf{x}) &= \mathbf{c}^T \mathbf{x}_{k+1} + \sum_{j=1}^C m_{[j]}^k(\mathbf{x}_{k+1}) \\ &= \mathbf{c}^T \mathbf{x}_{k+1} + \sum_{j=1}^C Q_{[j]}(\mathbf{x}_{k+1}) = Q(\mathbf{x}_{k+1}). \end{aligned}$$

Région de confiance : n'accepter une solution produite par le modèle que quand on a un gain suffisant sur la fonction objectif.

On va reprendre cette idée en introduisant le concept d'itération majeure (on accepte le candidat $\hat{\mathbf{x}}$ comme nouvel itéré \mathbf{x}_{k+1}) et mineure (on rejette $\hat{\mathbf{x}}$ et on cherche un meilleur candidat).

Itérations majeures et mineures

Une manière de noter les itérés : on ne considère que les itérations réussies pour représenter x_k .

Au cours de la k^e itération de l'algorithme, nous pourrions en effet être amené à résoudre plusieurs problèmes de régions de confiance, avec différents modèles et différentes valeurs Δ , avant d'obtenir un nouvel itéré x_{k+1} , ce qui conduira à distinguer

- itérés majeurs : x_k, x_{k+1}, \dots
- itérés mineurs : $x_{k,\ell}, \ell = 0, 1, 2, \dots$: obtenus en minimisant le modèle courant à l'intérieur de la région de confiance correspondante.

Le nouvel itéré mineur est accepté comme nouvel itéré majeur seulement s'il apporte une réduction substantielle de la fonction objectif $Q(x)$.

Régions de confiance : modèle

- $m_{k,\ell}(x)$: modèle après ℓ itérations mineures, à l'itération majeure k ;
- $\Delta_{k,\ell}$: rayon de la région de confiance à cette étape.
- Le problème à résoudre pour obtenir l'itéré mineur $x_{k,\ell}$ est

$$\min_{x \geq 0} m_{k,\ell}(x),$$

$$\text{t.q. } Ax = b,$$

$$\|x - x_k\|_\infty \leq \Delta_{k,\ell}.$$

Le modèle initial $m_{k,0}$ à l'itération majeure k satisfait

- 1 $m_{k,0}(x_k) = Q(x_k)$;
- 2 $m_{k,0}$ est une sous-estimation convexe, linéaire par morceaux, de Q .

Régions de confiance : modèle (2)

Nous procéderons de nouveau par segmentation de l'ensemble des scénarios en C sous-ensembles ("clusters") $\mathcal{S}_1, \dots, \mathcal{S}_C$:

$$\begin{aligned} \min_{\mathbf{x}, \theta_1, \dots, \theta_C} \quad & \mathbf{c}^T \mathbf{x} + \sum_{j=1}^C \theta_j \\ \text{t.q.} \quad & \theta_j \mathbf{e} \geq F_{[j]}^{k,\ell} \mathbf{x} + f_{[j]}^{k,\ell}, \quad j = 1, 2, \dots, C, \\ & A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \\ & -\Delta_{k,\ell} \mathbf{e} \leq \mathbf{x} - \mathbf{x}_k \leq \Delta_{k,\ell} \mathbf{e}. \end{aligned}$$

Soit $\mathbf{x}_{k,\ell}$ la solution obtenue en minimisant $m_{k,\ell}$, et $\alpha \in (0, 1/2)$.
Nous n'acceptons ce point que s'il permet une décroissance suffisante de Q :

$$Q(\mathbf{x}_{k,\ell}) \leq Q(\mathbf{x}_k) - \alpha(Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell})).$$

Si le test (de réduction suffisante) échoue, comment construire $m_{k,\ell+1}$?

Ajout et éventuellement suppression de coupes.

- Affiner le modèle pour obtenir finalement une nouvelle itération majeure.
- Permettre d'économiser sur le stockage en permettant la suppression de sous-gradients qui ne semblent plus utiles.
- Ajout et suppression de lignes dans $F^{k,\ell}[j]$ et $f_{[j]}^{k,\ell}$.

Comment décider de conserver ou de supprimer une coupe ?

On garde une coupe si

- elle a été générée en x_k (soit pour produire l'itéré majeure) ;
- elle est active lors de la solution du programme maître, avec un multiplicateur de Lagrange positif ;
- elle a été générée lors d'une précédente itération mineure $\bar{\ell} = 0, 1, \dots, \ell - 1$, avec

$$Q(x_k) - m_{k,\ell}(x_{k,\ell}) > \eta \left[Q(x_k) - m_{k,\bar{\ell}}(x_{k,\bar{\ell}}) \right],$$

et $\eta \in (\alpha, 1)$,

- si elle a été active en une des β dernières solutions du programme maître (par exemple, $\beta = 100$).

Bref, une coupe est abandonnée si visiblement elle n'apporte plus rien lors de la résolution du programme maître.

Propriétés du modèle $m_{k,\ell}$

Comme nous conservons toutes les coupes générées en x_k au cours de l'itération majeure k , nous avons

$$m_{k,\ell}(x_k) = Q(x_k), \ell = 0, 1, 2, \dots$$

et $m_{k,\ell}$ est une sous-estimation convexe, linéaire par morceaux, de Q , pour $\ell = 0, 1, 2, \dots$. Comme dans le contexte classique des régions de confiance, entre itérations mineures, nous pouvons décroître le rayon :

$$\Delta_{k,\ell+1} < \Delta_{k,\ell}.$$

Pour décider de cette réduction, comme dans le schéma classique, nous calculons

$$\rho = \frac{Q(x_k) - Q(x_{k,\ell})}{Q(x_k) - m_{k,\ell}(x_{k,\ell})} = \frac{Q(x_k) - Q(x_{k,\ell})}{m_{k,\ell}(x_k) - m_{k,\ell}(x_{k,\ell})}.$$

Vu que nous minimisons $m_{k,\ell}(\cdot)$,

$$Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell}) > 0.$$

Comme on veut ultimement minimiser $Q(\mathbf{x})$, on espère

$$Q(\mathbf{x}_k) - Q(\mathbf{x}_{k,\ell}) > 0,$$

mais rien ne nous le garantit, et donc, on peut avoir

$$\rho = \frac{Q(\mathbf{x}_k) - Q(\mathbf{x}_{k,\ell})}{Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell})} < 0.$$

Contrairement aux méthodes de région de confiance traditionnelles, le modèle est ici une sous-estimation de l'objectif, non une estimation, aussi est-il plus difficile de capitaliser sur le cas où $\rho > 0$, mais nous espérons ultimement avoir $\rho \approx 1$. Le cas $\rho > 0$ est étudié lors de l'acceptation d'un nouvel itéré majeur.

Réduction de $\Delta_{k,\ell}$

Dans leur article, Linderoth et Wright proposent la stratégie ci-dessous.

Algorithm 2: Réduction de $\Delta_{k,\ell}$ au cours des itérations mineures.

Evaluer

$$\rho = \min(1, \Delta_{k,\ell}) \frac{Q(\mathbf{x}_k) - Q(\mathbf{x}_{k,\ell})}{Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell})}.$$

Si $\rho < 0$, `counter` \leftarrow `counter`+1. Si $\rho < -3$, ou (`counter` ≥ 3 et $\rho \in [-3, -1)$), poser

$$\Delta_{k,\ell_1} = \frac{1}{\max\{-\rho, 4\}} \Delta_{k,\ell},$$

et remettre `counter` à 0.

Si le pas vers un nouvel itéré majeur \mathbf{x}_{k+1} montre un accord particulièrement bon entre la véritable fonction Q et le modèle $m_{k,\ell}$ lors de la dernière itération mineure de l'itération majeure k , nous envisageons d'accroître le rayon de la région de confiance.

Plus précisément, si

$$Q(\mathbf{x}_{k,\ell}) \leq Q(\mathbf{x}_k) - 0.5(Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell})), \quad \|\mathbf{x}_k - \mathbf{x}_{k,\ell}\|_\infty = \Delta_{k,\ell},$$

on posera

$$\Delta_{k+1,0} = \min\{\Delta_{\max}, 2\Delta_{k,\ell}\},$$

où Δ_{\max} est une borne spécifiée a priori (e.g. 2^{30}).

Algorithme L -shaped avec région de confiance

Etape 0 Choisir $\alpha \in (0, 1/2)$, le paramètre de suppression de coupe $\eta \in (\alpha, 1)$, Δ_{\max} , ε_{tol} .
Choisir le point de départ x_0 , le modèle $m_{0,0}$ et $\Delta_{0,0} \in [1, \Delta_{\max}]$.

Etape 1 $\ell \rightarrow 0$, counter $\leftarrow 0$.

Etape 2 Résoudre le problème maître avec $m_{k,\ell}$ pour obtenir $x_{k,\ell}$. Si

$$Q(x_k) - m_{k,\ell} \leq \varepsilon_{\text{tol}}(1 + |Q(x_k)|),$$

arrêt avec la solution approchée x_k .

Etape 3 Evaluer la fonction et le sous-gradient en $\mathbf{x}_{k,\ell}$. Si

$$\mathcal{Q}(\mathbf{x}_{k,\ell}) \leq \mathcal{Q}(\mathbf{x}_k) - \alpha(\mathcal{Q}(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell})),$$

- $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_{k,\ell}$;
- obtenir $m_{k+1,0}$ en supprimant éventuellement des coupes de $m_{k,\ell}$, mais en conservant le fait que $m_{k+1,0}$ est une sous-estimation convexe, linéaire par morceaux, de \mathcal{Q} , avec $m_{k+1,0}(\mathbf{x}_{k+1}) = \mathcal{Q}(\mathbf{x}_{k+1})$;
- choisir $\Delta_{k+1,0} \in [\Delta_{k,\ell}, \Delta_{\max}]$;
- retour à l'Etape 1.

Etape 4 Obtenir $m_{k,\ell+1}$ à partir de $m_{k,\ell}$, et $\Delta_{k,\ell+1}$. Poser $\ell \leftarrow \ell + 1$. Retour à l'Etape 2.

Pourquoi avoir choisi $\|\cdot\|_\infty$ et non $\|\cdot\|_2$?

On peut montrer que la solution du sous-problème

$$\min_{x \geq 0} m_{k,\ell}(x) \text{ t.q. } Ax = b, \|x - x_k\|_2 \leq \Delta_k$$

est identique à la solution de

$$\min_{x \geq 0} m_{k,\ell}(x) + \lambda \|x - x_k\|_2^2 \text{ t.q. } Ax = b,$$

pour un certain $\lambda \geq 0$.

On retrouve la régularisation par pénalité ! Alors que l'utilisation d'une région de confiance et de la norme $\|\cdot\|_\infty$ permet de conserver un programme linéaire comme sous-problème.

D'autre part, il est connu que la norme $\| \cdot \|_2$ offre souvent pour les méthodes de régions de confiance une meilleure direction de recherche que la norme $\| \cdot \|_\infty$ quand l'objectif est lisse.

Une telle observation n'est pas claire pour la fonction Q , qui est linéaire par morceaux, avec un grand nombre de morceaux.

Linderoth et Wright proposent des versions parallélisées du *L-shaped* multi-coupe hybride (avec clusters), incluant ou non le mécanisme de région de confiance. Ils ont aussi prouvé la convergence de l'algorithme basés sur les régions de confiance.

Preuve de convergence : aperçu

Beaucoup des éléments ci-dessous sont communs aux preuves rencontrées en régions de confiance classiques.

Dans la suite, nous désignerons l'ensemble des solutions optimales du problème initial par \mathcal{X}^* . Nous souhaitons montrer que l'algorithme va produire un point de \mathcal{X}^* , ou à défaut, une séquence de points dont les points limites sont dans \mathcal{X}^* . Nous dénoterons par $P(\cdot)$ l'opérateur de projection sur \mathcal{X}^* .

Lemma (Augmentation du modèle lors des itérés mineurs)

Supposons que $\mathbf{x}_{k,\ell}$ ne satisfait pas le test pour être accepté comme nouvel itéré majeur (i.e. devenir \mathbf{x}_{k+1}). Alors,

$$m_{k,\ell}(\mathbf{x}_{k,\ell}) \leq m_{k,\ell_1}(\mathbf{x}_{k,\ell+1}).$$

Lemma (Réduction suffisante)

Pour tout $k = 0, 1, 2, \dots$ et $\ell = 0, 1, 2, \dots$, nous avons

$$\begin{aligned} m_{k,\ell}(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell}) &= Q(\mathbf{x}_k) - m_{k,\ell}(\mathbf{x}_{k,\ell}) \\ &\geq \min \left(\frac{\Delta_{k,\ell}}{\|\mathbf{x}_k - P(\mathbf{x}_k)\|_\infty}, 1 \right) [Q(\mathbf{x}_k) - Q^*]. \end{aligned}$$

Lemma (Borne inférieure sur $\Delta_{k,\ell}$)

Pour tout rayon $\Delta_{k,\ell}$ utilisé au cours de l'algorithme, nous avons

$$\Delta_{k,\ell} \geq \frac{1}{4} \min \left(\min_{\gamma=1,\dots,k} \|\mathbf{x}_\gamma - P(\mathbf{x}_\gamma)\|_\infty, \frac{1}{\beta} \min_{\gamma=0,1,\dots,k, \mathbf{x} \notin \mathcal{X}^*} \frac{Q(\mathbf{x}_\gamma) - Q^*}{\|\mathbf{x}_\gamma - P(\mathbf{x}_\gamma)\|_\infty} \right),$$

où, $\beta = \sup\{\|g\|_1 \mid g \in \partial Q(x), x \in \mathcal{L}(Q_0, \Delta_{\max})\}$ et $\mathcal{L}(Q_0) = \{x \mid Ax = b, x \geq 0, Q(x) \leq Q_0\}$.

Preuve de convergence : aperçu (4)

Lemma (Nombre fini d'itérations mineures - lemme)

Posons $\varepsilon_{tol} = 0$. Soit l_1 un indice quelconque tel que x_{k,l_1} échoue au test d'acceptation pour devenir un itéré majeur. Alors, soit la séquence d'itérations mineures porte finalement un point x_{l_2} satisfaisant le test d'acceptation, ou il existe un indice $l_2 > l_1$ tel que (plus faible décroissance)

$$(0 \leq) \mathcal{Q}(x_k) - m_{k,l_2}(x_{k,l_2}) \leq \eta[\mathcal{Q}(x_k) - m_{k,l_1}(x_{k,l_1})].$$

Théorème (Nombre fini d'itérations mineures)

Posons $\varepsilon_{tol} = 0$.

- (i) Si $x_k \notin \mathcal{X}^*$, il existe un $l \geq 0$ tel que $x_{k,l}$ satisfait le critère d'acceptation d'itéré majeur.*
- (ii) Si $x_k \in \mathcal{X}^*$, alors soit l'algorithme se termine, soit $\mathcal{Q}(x_k) - m_{k,l}(x_{k,l}) \downarrow 0$ (et donc on s'arrêtera pour $\varepsilon_{tol} > 0$).*

Théorème

Supposons que $\varepsilon_{tol} = 0$. La séquence d'itérés majeurs $\{x_k\}$ est soit finie, se terminant avec un point appartenant à \mathcal{X}^ , ou infinie, avec $\|x_k - P(x_k)\|_\infty \rightarrow 0$.*

Dans leur papier, Linderoth et Wright rapportent un exemple numérique dont l'équivalent déterministe a plus de 10^{10} inconnues, et qu'ils résolvent avec la version parallélisée de leur méthode.

On perd ici la garantie de trouver une solution exacte en un nombre fini d'itérations, mais sur le plan numérique, de telles approches sont souvent préférables, car plus efficaces.

Autres méthodes intéressantes : points-intérieurs.