

Gradient Perturbé dans un espace de Hilbert

Théorie et Applications

Kengy Barty² Jean-Sébastien Roy¹ Cyrille Strugarek¹

¹EDF R&D

²Ecole Nationale des Ponts et Chaussées

7 avril 2005

Plan de la présentation

- 1 Gradient stochastique
- 2 Gradient stochastique en boucle fermée
- 3 Approximation stochastique

Problème de moindres carrés stochastique

On cherche $u \in \mathbb{R}$ qui minimise en espérance le carré de l'écart avec une variable aléatoire ξ suivant une loi normale de moyenne x et de variance s^2 . Soit :

$$\min_u J(u) := \mathbb{E}(j(u, \xi))$$

où $j(u, \xi) = \frac{1}{2} (u - \xi)^2$ et $\xi \sim \mathcal{N}(x, s^2)$.

La solution de ce problème est $u = \mathbb{E}(\xi) = x$.

Résolution par Monte-Carlo

Comment calculer $\mathbb{E}(\xi)$? Une méthode est de réaliser des tirages de ξ et d'en faire la moyenne. C'est la méthode de Monte-Carlo. Précisément, soit :

- $(\xi_k)_{k \in \mathbb{N}}$ des tirages i.i.d.¹, selon la loi de ξ ,
- $u_n = \frac{1}{n} \sum_{k=1}^n \xi_k$

Alors $\lim_{n \rightarrow \infty} u_n = x$. Il suffit donc de faire suffisamment de tirages pour résoudre le problème avec la précision souhaitée.

¹Indépendants et identiquement distribués.

Présentation incrémentale

Le calcul de u_n peut être réalisé de manière incrémentale. En effet, en posant $u_0 = 0$:

$$u_n = \frac{1}{n} \sum_{k=1}^n \xi_k = u_{n-1} + \frac{1}{n} (\xi_n - u_{n-1})$$

On peut aussi utiliser un pas autre que $\frac{1}{n}$ dans cette formule : cela correspond à faire une moyenne pondérée. Il suffit que le pas p_n soit une σ -suite, c'est-à-dire :

- $\sum_n p_n = \infty$ (il faut pouvoir avancer aussi loin que nécessaire) ;
- et $\sum_n (p_n)^2 < \infty$ (il ne faut pas avancer trop vite et diverger).

Algorithme du gradient stochastique

On remarque que $u_{n-1} - \xi_n$ est le gradient de $j(u_{n-1}, \xi_n)$ par rapport à u . Ainsi notre algorithme s'écrit :

- $u_0 = 0$ (le point de départ n'a pas d'importance)
- $u_n = u_{n-1} - \rho_n \nabla_{uj}(u_{n-1}, \xi_n)$

Il s'agit de l'algorithme du gradient stochastique. Celui-ci fonctionne pour une large classe de fonctions j (pas seulement pour les problèmes de moindres carrés) du moment que u ne dépend pas de ξ (i.e. le problème est en boucle ouverte).

Lien avec la méthode du gradient

La méthode du gradient (non stochastique) correspondante serait :

$$u_n = u_{n-1} - \rho_n \nabla J(u_{n-1})$$

où $\nabla J(u_{n-1}) = \mathbb{E}(\nabla_{uj}(u_{n-1}, \xi))$. La méthode du gradient stochastique correspond à remplacer cette espérance par une réalisation de la variable aléatoire $\nabla_{uj}(u_{n-1}, \xi)$ en un tirage ξ_n des aléas. Cette réalisation $\nabla_{uj}(u_{n-1}, \xi_n)$ est une estimation sans biais de $\nabla J(u_{n-1})$:

$$\mathbb{E}(\nabla_{uj}(u_{n-1}, \xi_n)) = \nabla J(u_{n-1})$$

Gradient stochastique avec projection

La méthode du gradient stochastique s'adapte au cas où l'on restreint le domaine dans lequel on recherche u (typiquement pour imposer des limites à une commande).

Soit U l'espace dans lequel on autorise u à évoluer, et Π_U la projection sur U . La mise à jour s'effectue en projetant à chaque itération :

$$u_n = \Pi_U [u_{n-1} - \rho_n \nabla_{u_j} j(u_{n-1}, \xi_n)]$$

Notons que si U est un espace défini par des contraintes complexes, la projection peut s'avérer difficile à calculer. Dans ce cas, on pénalisera ces contraintes.

Exemple d'application

Définition du problème

On cherche à résoudre un problème de gestion hydraulique face à un prix x connu. Soit u la quantité turbinée, le gain est xu . Enfin, la valeur du stock résiduel est prise égale à $\sqrt{\eta + s + \xi - u}$ où s est le stock initial, η une constante strictement positive et ξ un apport aléatoire, suivant une loi uniforme sur $[0, a]$.

Ainsi, on cherche à résoudre :

$$\min_{u \in [0, s]} J(u) := \mathbb{E} \left(-xu - \sqrt{\eta + s + \xi - u} \right)$$

Exemple d'application

Résolution du problème

L'algorithme s'écrit :

$$u_n = \Pi_{[0,s]} [u_{n-1} - p_n \nabla_{uj} (u_{n-1}, \xi_n)]$$

où :

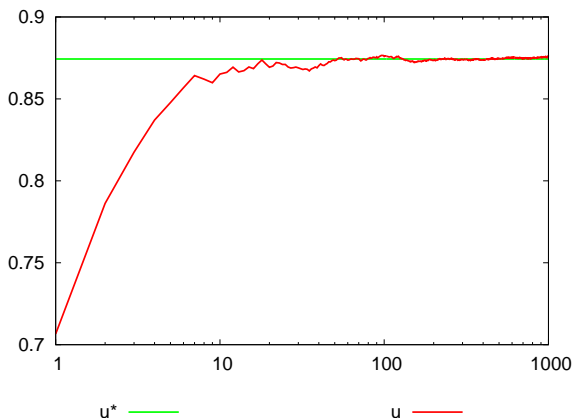
$$\nabla_{uj} (u_{n-1}, \xi_n) = -x + \frac{1}{2\sqrt{\eta + s + \xi_n - u_{n-1}}}$$

et $\Pi_{[0,s]}(u)$ correspond à tronquer la valeur de u pour qu'elle reste dans $[0, s]$.

En général, on prend p_n de la forme $\frac{a}{b+n}$.

Exemple d'application

Itérations



Problèmes en boucle ouverte

La méthode du gradient stochastique s'applique à la plupart des problèmes de commande optimale stochastique. Par exemple :

- Gestion de production sur un arbre d'aléas (type Soprano).

On s'en sert peu en pratique, la programmation dynamique étant souvent plus adaptée à nos problèmes.

Problèmes en boucle fermée

Définition

Malheureusement dans de nombreux cas on cherche non pas à obtenir une commande optimale dans un cas particulier, mais une stratégie de commande en fonction de la réalisation d'aléas.

Rechercher une telle stratégie correspond à rechercher u comme une fonction des aléas ξ : pour chaque réalisation de l'aléa ξ , $u(\xi)$ correspond à la commande optimale à réaliser. Le problème s'écrit :

$$\min_u J(u) := \mathbb{E}(j(u(\xi), \xi))$$

Cette fois u est une fonction et est donc dans un espace de dimension infinie.

Problèmes en boucle fermée

Exemple

On cherche à résoudre un problème de gestion hydraulique face à un prix ξ . Soit $u(\xi)$ la quantité turbinée, le gain est $\xi u(\xi)$. Enfin, la valeur du stock résiduel est prise égale à $\sqrt{\eta + s - u(\xi)}$ où s est le stock initial et η une constante strictement positive.

Ainsi, on cherche à résoudre :

$$\min_u J(u) := \mathbb{E} \left(-\xi u(\xi) - \sqrt{\eta + s - u(\xi)} \right)$$

où l'on impose à la fonction $u : \forall \xi, u(\xi) \in [0, s]$.

Calcul du gradient, inapplicabilité de la méthode

La méthode du gradient stochastique exposée précédemment n'est alors plus applicable. Le gradient² $\nabla J(u)(\cdot) = \nabla_{\omega} j(u(\cdot), \cdot)$ est alors une fonction et est en général impossible à représenter avec un nombre fini de réels.

Nous verrons par la suite un moyen d'attaquer ce problème directement.

²Notons qu'il n'y a plus d'espérance.

Règles de décision linéaires

Définition

Pour contourner ce problème, la solution usuelle est de paramétrer la stratégie recherchée. Plutôt que de chercher la stratégie optimale, on cherche un ensemble de paramètres permettant de définir une stratégie approchée la meilleure possible. En général, cette paramétrisation est linéaire, c'est à dire :

$$u(\xi) = \sum_{i=1}^m a_i f_i(\xi)$$

où les $(a_i)_i$ sont les paramètres recherchés et $(f_i)_i$ sont des fonctions prédéfinies des aléas. Il s'agit d'une règle de décision linéaire[Holt et al., 1955].

Règles de décision linéaire

Résolution

Soit $a = (a_i)_i$ le vecteur des paramètres, on est donc ramené au problème :

$$\min_a \mathbb{E} (h(a, \xi))$$

où $h(a, \xi) = j(\sum_{i=1}^m a_i f_i(\xi), \xi)$. Ce problème est alors soluble par la méthode du gradient stochastique présentée précédemment.

Règles de décision linéaire

Exemple

Dans l'exemple précédent de gestion de gestion hydraulique face à un prix, on peut par exemple imposer $u(\xi)$ à être affine, c'est-à-dire prendre $f_1(\xi) = 1$, $f_2(\xi) = \xi$.

Ainsi, $u(\xi) = a_1 + a_2\xi$, et l'on cherche à résoudre :

$$\min_{(a_1, a_2)} \mathbb{E} \left(-\xi (a_1 + a_2\xi) - \sqrt{\eta + s - a_1 - a_2\xi} \right)$$

où, si $\xi \in [\underline{\xi}, \bar{\xi}]$ l'on impose aux paramètres : $a_1 + a_2\underline{\xi} \in [0, s]$ et $a_1 + a_2\bar{\xi} \in [0, s]$.

Cas particulier des arbres d'aléas

Principe

Notons que la discrétisation des aléas, par exemple à travers un arbre, représente un cas particulier de règle de décision linéaire.

Dans ce cas, les fonctions $(f_i)_i$, où i est le nœud de l'arbre, sont prises nulles pour tout ξ sauf pour l'ensemble des valeurs de ξ amenant au nœud i , auquel cas la fonction vaut 1.

Les valeurs des paramètres correspondent donc aux commandes aux nœuds.

Cas particulier des arbres d'aléas

Exemple

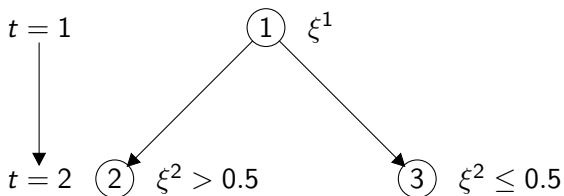
Par exemple, on considère un problème à deux pas de temps, où la valeur de l'aléa sur le premier pas de temps ξ^1 est connue au moment de la résolution, et l'aléa sur le deuxième pas de temps, ξ^2 , est compris entre 0 et 1.

ξ^1 étant connu, la stratégie s'écrit $u(\xi^2) = (u^1, u^2(\xi^2))$, i.e., seul ξ^2 est un aléa, la décision au premier pas de temps est indépendante de ξ^2 , et décision au deuxième pas de temps est fonction de ξ^2 .

Cas particulier des arbres d'aléas

Construction de l'arbre

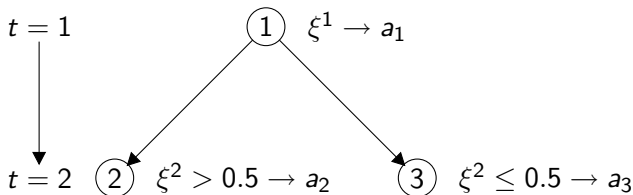
On peut alors construire un arbre à 3 nœuds : Un nœud 1, racine, lors duquel on prend la décision pour le premier pas de temps, puis deux nœuds correspondant au deuxième pas de temps, l'un (le numéro 2) pour les valeurs élevées de ξ^2 , par exemple $\xi^2 > 0.5$, l'autre (le numéro 3) pour les valeurs faibles $\xi^2 \leq 0.5$.



Cas particulier des arbres d'aléas

Règle de décision linéaire

La règle de décision linéaire $u(\xi) = \sum_{i=1}^m a_i f_i(\xi)$ correspond à prendre $f_1(\xi^2) = (1, 0)$, $f_2(\xi^2) = (0, \mathbf{1}_{\xi^2 > 0.5})$, $f_3(\xi^2) = (0, \mathbf{1}_{\xi^2 \leq 0.5})$. Ainsi, $u(\xi^2) = (a_1, a_2 \mathbf{1}_{\xi^2 > 0.5} + a_3 \mathbf{1}_{\xi^2 \leq 0.5})$. Les commandes $(a_i)_i$ correspondent aux décisions à chaque nœud i .



Problèmes en boucle fermée

Définition

Rappelons que les problèmes en boucle fermée consistent à rechercher non pas une commande mais une stratégie de commande fonction des aléas. Rechercher une telle stratégie correspond à rechercher u comme une fonction des aléas ξ : pour chaque réalisation de l'aléa ξ , $u(\xi)$ correspond à la commande optimale à réaliser. Le problème s'écrit :

$$\min_u J(u) := \mathbb{E}(j(u(\xi), \xi))$$

Cette fois u est une fonction et est donc dans un espace de dimension infinie.

Inapplicabilité de la méthode usuelle

Impossible de représenter le gradient

Les problèmes en boucle fermée sont de la forme :

$$\min_u J(u) := \mathbb{E}(j(u(\xi), \xi))$$

Nous avons vu que la méthode du gradient stochastique exposée précédemment n'est alors pas applicable. Le gradient³ $\nabla J(u)(\cdot) = \nabla_u j(u(\cdot), \cdot)$ est alors une fonction et est en général impossible à représenter avec un nombre fini de réels.

³Notons qu'il n'y a plus d'espérance.

Inapplicabilité de la méthode usuelle

Estimation du gradient sans biais inutilisable

Il est possible d'en calculer une estimation sans biais via une réalisation ξ des aléas par le produit de la valeur de $\nabla J(u)(\xi)$ avec un Dirac en ξ :

$$\nabla J(u)(\cdot) = \mathbb{E}(\nabla_{uj}(u(\xi), \xi) \delta_{\xi}(\cdot))$$

Malheureusement, ajouter des Dirac ne permettra jamais de converger vers la fonction $u(\cdot)$ optimale. Une somme dénombrable de Dirac est en général presque partout nulle.

Approximation d'un Dirac

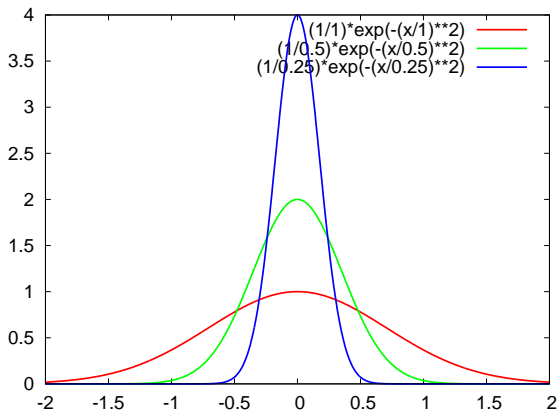
Le problème principal empêchant l'application du gradient stochastique à un problème en boucle fermée est l'apparition du Dirac pour l'estimation sans biais du gradient. Pour contourner ce problème, on se propose d'approximer le Dirac par une fonction en forme de bosse $\frac{1}{\epsilon_n} K_n(\xi, \cdot)$, type Gaussienne, centrée au même point que le Dirac $\delta_\xi(\cdot)$. Notre estimation de $\nabla J(u)(\cdot)$ est alors :

$$\nabla J(u)(\cdot) = \mathbb{E}(\nabla_{uj}(u(\xi), \xi) \delta_\xi(\cdot)) \simeq \mathbb{E}\left(\nabla_{uj}(u(\xi), \xi) \frac{1}{\epsilon_n} K_n(\xi, \cdot)\right)$$

où K_n est un noyau (fonction en forme de bosse), de hauteur 1 et de fenêtre (i.e. largeur) ϵ_n . On se rapproche de plus en plus du Dirac, en faisant tendre ϵ_n vers 0.

Approximation d'un Dirac

Avec des noyaux gaussiens de fenêtres 1, 0.5 et 0.25



Gradient stochastique en boucle fermée

Nous proposons donc l'algorithme suivant :

- $u_0(\cdot) = 0$
- $u_n(\cdot) = u_{n-1}(\cdot) - p_n \frac{1}{\epsilon_n} \nabla_{uj} (u_{n-1}(\xi_n), \xi_n) K_n(\xi_n, \cdot)$

Cet algorithme est implémentable en pratique : la fonction u_n est représentée par la liste des n triplets $(\xi_k, \epsilon_k, p_k \nabla_{uj} (u_{k-1}(\xi_k), \xi_k))_k$ et se calcule par :

$$u_n(\xi) = - \sum_{k=1}^n p_k \frac{1}{\epsilon_k} \nabla_{uj} (u_{k-1}(\xi_k), \xi_k) K_k(\xi_k, \xi)$$

Il reste à démontrer que cet algorithme converge.

Remarque sur le temps de calcul

L'algorithme proposé est essentiellement composé de deux étapes de calcul :

- 1 D'une part le calcul de $u_{n-1}(\xi_n)$;
- 2 D'autre part le calcul de $\nabla_{uj}(u_{n-1}(\xi_n), \xi_n)$.

Le calcul $u_{n-1}(\xi_n)$ est au pire de complexité linéaire en le nombre de tirage effectués et de plus chaque étape est élémentaire. Enfin, en l'absence de projection, avec les fenêtres usuelles l'utilisation d'un kd-tree permet de rendre ce calcul de complexité logarithmique (les noyaux sont localisés).

En pratique, le calcul du gradient (par exemple, simulation d'un système électrique sur un scénario d'aléas) sera l'étape prédominante. Cette étape étant inévitable, l'algorithme semble peu coûteux.

Algorithme de gradient perturbé

On cherche à résoudre des problèmes du type :

$$\min_{u \in U} J(u)$$

à l'aide d'un algorithme de gradient perturbé du type :

$$u_n = \Pi_U [u_{n-1} + p_n (s_n + w_n)]$$

où s_n est une direction de descente, w_n une perturbation et p_n un pas positif décroissant.

Théorème de convergence de Bertsekas

Soit \mathcal{F}_n une filtration par rapport à laquelle s_n et w_n sont mesurables.

Sans rentrer dans les détails, Bertsekas et Tsitsiklis [Bertsekas and Tsitsiklis, 2000] ont montré que ce type d'algorithme convergeait dans le cas où :

- $\exists c_1 > 0$ tq $c_1 |\nabla J(u_{n-1})|^2 \leq -\langle \nabla J(u_{n-1}), s_n \rangle$, i.e. s_n est une direction de descente,
- $\exists c_2 > 0$ tq $|s_n| \leq c_2 (1 + |\nabla J(u_{n-1})|)$, i.e. la norme de s_n est contrôlée,
- $\mathbb{E}(w_n | \mathcal{F}_n) = 0$, i.e. la perturbation est sans biais,
- $\exists b_2 > 0$, $\mathbb{E}(|w_n|^2 | \mathcal{F}_n) \leq b_2 (1 + |\nabla J(u_{n-1})|^2)$, i.e. la variance de la perturbation est contrôlée,
- $(p_n)_n$ est une σ -suite.

Cas fonctionnel, biais asymptotiquement nul

Malheureusement, l'estimation utilisée dans le cadre du gradient stochastique en boucle fermée (i.e. quand u est une fonction) n'est pas sans biais :

$$\mathbb{E} \left(\frac{1}{\epsilon_n} \nabla_{uj} (u(\xi), \xi) K_n(\xi, \cdot) \right) \neq \nabla J(u) (\cdot)$$

Mais juste asymptotiquement sans biais, si $\lim_{n \rightarrow \infty} \epsilon_n = 0$:

$$\lim_{n \rightarrow \infty} \mathbb{E} \left(\frac{1}{\epsilon_n} \nabla_{uj} (u(\xi), \xi) K_n(\xi, \cdot) \right) = \nabla J(u) (\cdot)$$

On ne peut donc utiliser le théorème précédent. Néanmoins, nous avons démontré que l'algorithme convergeait même si l'estimation du gradient n'était pas sans biais mais seulement asymptotiquement sans biais.

Autres théorèmes précédents

D'autres auteurs [Kushner and Clark, 1978, Kulkarni and Horn, 1996, Delyon, 1996, Chen and White, 1998] ont démontré des variantes du théorème précédent, malheureusement, ces approches ont toutes différents défauts qui rendent leur utilisation dans notre cas difficile :

- Soit l'approche est limitée à la dimension finie ;
- Soit il n'est pas possible de prendre en compte des contraintes sur u ;
- Soit l'approche est limitée au cas déterministe.

Enfin, de manière générale, ces approches sont assez difficiles d'accès.

Nouveau théorème de convergence

Soit u^* une solution du problème. Les hypothèses [Barty et al., 2005a] sont :

- $\exists c_1 > 0$ tq $\langle u_{n-1} - u^*, s_n \rangle \leq c_1 (J(u^*) - J(u_{n-1}))$, i.e. s_n est une direction de descente,
- $\exists c_2 > 0$ tq $|s_n| \leq c_2 (1 + |\nabla J(u_{n-1})|)$, i.e. la norme de s_n est contrôlée,
- $\exists b_1 \geq 0$, $|\mathbb{E}(w_n | \mathcal{F}_n)| \leq b_1 \epsilon_n (1 + |\nabla J(u_{n-1})|)$, i.e. la perturbation est asymptotiquement sans biais,
- $\exists b_2 > 0$, $\mathbb{E}(|w_n|^2 | \mathcal{F}_n) \leq b_2 \left(1 + \frac{1}{\epsilon_n} |\nabla J(u_{n-1})|^2\right)$, i.e. la variance de la perturbation est contrôlée (mais peut être non-bornée),
- $(p_n)_n$ et (ϵ_n) vérifient $\sum p_n = \infty$, $\sum \frac{(p_n)^2}{\epsilon_n} < \infty$ et $\sum b_1 p_n \epsilon_n < \infty$.

Hypothèses sur la fonction objectif

Cas général

Pour que l'on puisse appliquer la méthode, il faut tout de même certaines hypothèses de régularité et de convexité sur J .

Précisément, J doit être :

- fortement convexe ;
- semi-continue inférieurement ;
- et le gradient de J , $\nabla J(\cdot)$ doit être Lipschitzien.

De plus l'espace U doit être un convexe fermé.

Hormis la forte convexité, ces hypothèses sont assez naturellement vérifiées dans nos problèmes.

Une version du théorème existe aussi dans le cas où J est seulement convexe, et U un espace vectoriel (et quelques autres hypothèses).

Application au gradient stochastique en boucle fermée

L'application de ce théorème à l'algorithme de gradient stochastique en boucle fermée est alors relativement directe [Barty et al., 2005b] : il suffit d'imposer quelques contraintes à la fonction K_n .

Soit $r_n(\cdot) = \nabla j(u_n(\cdot), \cdot)$, on impose :

- $\exists b_1 \geq 0$ tq $\left| r_{n-1} - \mathbb{E} \left(r_{n-1}(\xi) \frac{1}{\epsilon_n} K_n(\xi, \cdot) \right) \right| \leq b_1 \epsilon_n (1 + |r_{n-1}|)$, i.e. le noyau est une approximation d'un Dirac,
- $\exists b_2 \geq 0$ tq $\mathbb{E} \left(|K_n(\xi, \cdot)|^2 \right) \leq b_2 \epsilon_n$, i.e. la variance du noyau est contrôlée.

Ces hypothèses sont très peu restrictives.

Hypothèses sur la fonction objectif

Cas des problèmes en boucle fermée

Les hypothèses sur la fonction objectif J présentées dans le cas général se traduisent en hypothèses sur la fonction j . Celle-ci doit être :

- fortement convexe par rapport à u uniformément en ξ ;
- semi-continue inférieurement par rapport à u ;
- conjointement mesurable sur U , convexe fermé de \mathcal{L}^2 ;
- $\nabla_u j(\cdot, \xi)$ lipschitzienne en u uniformément en ξ .

Similairement, une autre version du théorème ne requiert que la convexité, si U est un sous espace vectoriel (et quelques autres hypothèses).

Hypothèses sur le pas et la fenêtre

Les hypothèses sur le pas p_n et la fenêtre ϵ_n sont $\sum p_n = \infty$, $\sum \frac{(p_n)^2}{\epsilon_n} < \infty$ et $\sum b_1 p_n \epsilon_n < \infty$. Ces hypothèses s'expliquent assez bien (et sont classiques dans le cas de perturbations biaisées[Kiefer and Wolfowitz, 1952]). Il faut en effet :

- ne pas diverger à cause du biais ($\sum b_1 p_n \epsilon_n < \infty$) ;
- qu'en tout point, l'algorithme se comporte comme un algorithme de gradient stochastique usuel, i.e, comme le pas en chaque point est $\frac{p_n}{\epsilon_n}$, Il faut qu'en espérance, cela soit une σ -suite.

On note que ces hypothèses impliquent naturellement :

- de progresser partout, i.e. $\sum \epsilon_n = \infty$;
- de tendre vers un gradient sans biais, i.e. $\lim_{n \rightarrow \infty} \epsilon_n = 0$.

Hypothèses sur le pas et la fenêtre

Rôle de la dimension de l'espace

Notons que, même si cela n'apparaît pas explicitement dans les formules précédentes, la dimension de l'espace Ξ a un impact significatif sur le choix de ϵ_n .

En effet, pour que $\frac{1}{\epsilon_n} K_n(\xi, \cdot)$ soit l'approximation d'un Dirac, il faut notamment $\mathbb{E} \left(\frac{1}{\epsilon_n} K_n(\xi, \cdot) \right) = 1$.

Ainsi dans un espace vectoriel réel de dimension d , si le noyau choisi est le noyau gaussien produit de fenêtre η_n :

$$K_n(\xi, x) = \prod_{i=1}^d \left[\frac{1}{\sqrt{2\pi}} e^{-\left(\frac{\xi_i - x_i}{\eta_n}\right)^2} \right]$$

il faut alors que $\epsilon_n = \eta_n^d$. Ainsi $\epsilon_n \sim \frac{1}{n} \implies \eta_n \sim \frac{1}{\sqrt[d]{n}}$.

Choix du pas et de la fenêtre

Le choix des pas est une part importante du problème. On peut naturellement choisir $p_n = \frac{1}{n}$ et $\epsilon_n = \frac{1}{\sqrt{n}}$, mais certains choix peuvent être plus intéressants, notamment si la répartition des tirages de ξ n'est pas uniforme. On peut par exemple souhaiter que la décroissance des pas et des noyaux soit fonction du nombre de tirages réalisés dans le voisinage du point considéré.

Pour cela, il est possible de se servir de la somme des noyaux déjà utilisés, au point ξ_n tiré, soit $d_n = \sum_{i < n} K_i(\xi_i, \xi_n)$.

Par exemple, soit $(\eta_n)_n$ une suite adéquate de fenêtres, et j_n tel que $\sum_{i < j_n} \eta_i \sim d_n$, on peut prendre $\epsilon_n = \eta_{j_n}$. De même pour le pas avec $p_n = \eta_{d_n}$. Il peut être plus simple de contrôler $\gamma_n = \frac{p_n}{\epsilon_n}$ que p_n .

Cas des contraintes de mesurabilité

Définition

Dans de nombreux cas, la fonction u recherchée doit respecter des contraintes de non-anticipativité. Typiquement, pour un problème à 2 pas de temps, et donc deux bruits $\xi = (\xi^1, \xi^2)$, et deux commandes, i.e., $u(\xi) = (u^1(\xi), u^2(\xi))$, où u^1 est la commande au premier pas de temps et u^2 la commande au deuxième pas de temps, on désirera que u^1 ne dépende que de ξ^1 (car ξ^2 est inconnu lors du premier pas de temps) alors que u^2 peut dépendre de ξ^1 et ξ^2 .

Ces contraintes sont appelées contraintes de mesurabilité sur u .

Cas des contraintes de mesurabilité

Mise en œuvre

De telles contraintes sont simples à mettre en œuvre dans notre algorithme car la projection sur l'espace des contraintes de mesurabilité est aisée à réaliser.

Soit $\mathcal{K}_\epsilon(\xi, \cdot)$ le noyau gaussien de fenêtre ϵ , $x = (x^1, x^2)$ le point où l'on veut évaluer le noyau, et U l'espace des contraintes de mesurabilité précédemment décrit. On a :

$$\mathcal{K}_\epsilon(\xi, x) = \mathcal{K}_\epsilon(\xi^1, x^1) \mathcal{K}_\epsilon(\xi^2, x^2)$$

$$\Pi_U[\mathcal{K}_\epsilon(\xi, x)] = (\epsilon_n \mathcal{K}_{\epsilon_n}(\xi^1, x^1), \mathcal{K}_{\epsilon_n}(\xi^1, x^1) \mathcal{K}_{\epsilon_n}(\xi^2, x^2))$$

Il suffira donc de prendre un noyau déjà projeté sur les contraintes, i.e. $\mathcal{K}_n = \Pi_U[\mathcal{K}_{\epsilon_n}]$.

Problème de moindres carrés avec contrainte

Description du problème

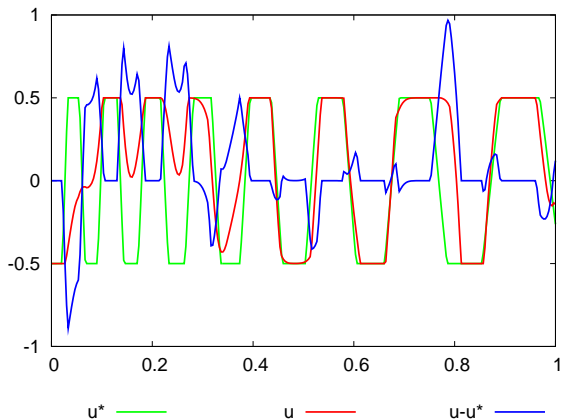
On cherche à résoudre :

$$\min_u J(u) := \mathbb{E} \left(\left(u(\xi) - \sin \left(\frac{100}{\xi + 1} \right) \right)^2 \right)$$

où l'on impose à la fonction $u : \forall \xi, u(\xi) \in \left[-\frac{1}{2}, \frac{1}{2}\right]$.

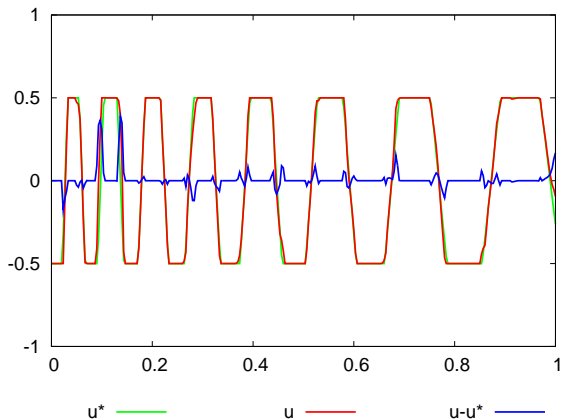
Problème de moindres carrés avec contrainte

50 itérations



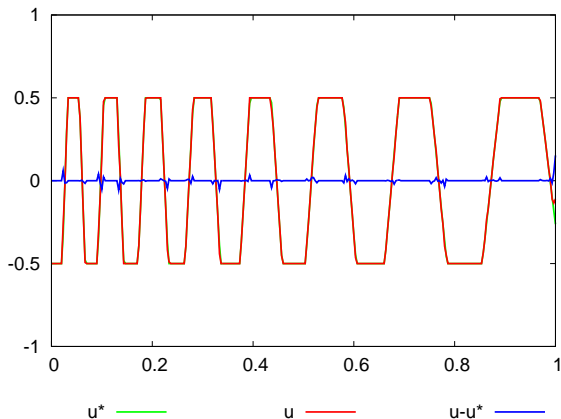
Problème de moindres carrés avec contrainte

200 itérations



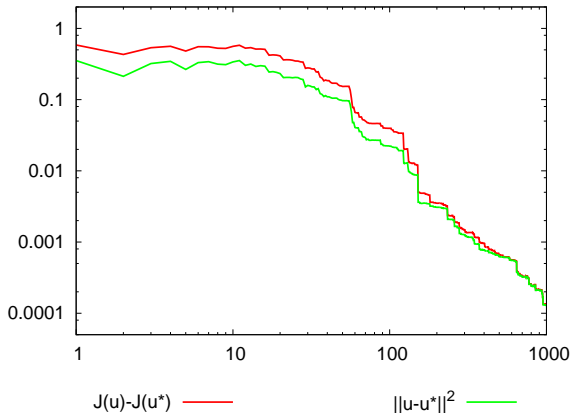
Problème de moindres carrés avec contrainte

1000 itérations



Problème de moindres carrés avec contrainte

Vitesse de convergence



Gestion d'une réserve hydraulique (1 pas de temps)

Description du problème

On cherche à résoudre un problème de gestion hydraulique face à un prix ξ . Soit $u(\xi)$ la quantité turbinée, le gain est $\xi u(\xi)$. Enfin, la valeur du stock résiduel est prise égale à $\sqrt{\eta + s - u(\xi)}$ où s est le stock initial et η une constante strictement positive.

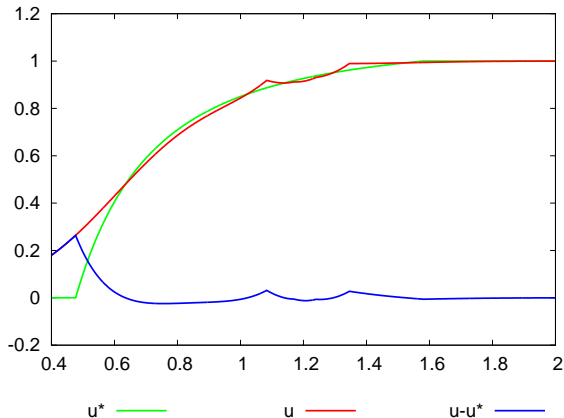
Ainsi, on cherche à résoudre :

$$\min_u J(u) := \mathbb{E} \left(-\xi u(\xi) - \sqrt{\eta + s - u(\xi)} \right)$$

où l'on impose à la fonction $u : \forall \xi, u(\xi) \in [0, s]$.

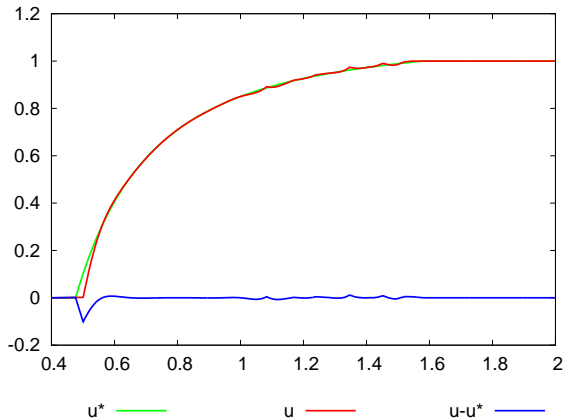
Gestion d'une réserve hydraulique (1 pas de temps)

500 itérations



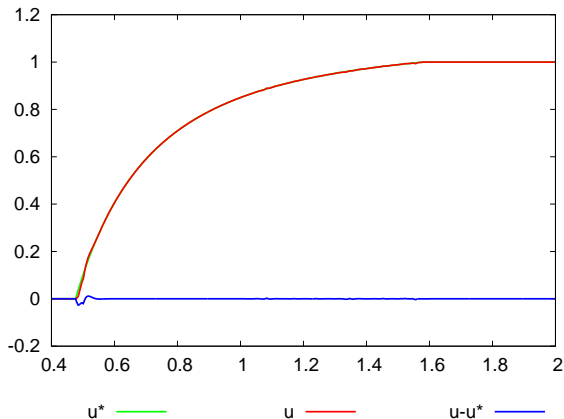
Gestion d'une réserve hydraulique (1 pas de temps)

3000 itérations



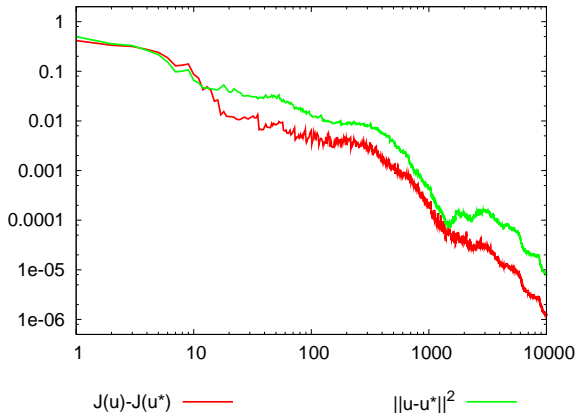
Gestion d'une réserve hydraulique (1 pas de temps)

10000 itérations



Gestion d'une réserve hydraulique (1 pas de temps)

Vitesse de convergence



Gestion d'une réserve hydraulique (2 pas de temps)

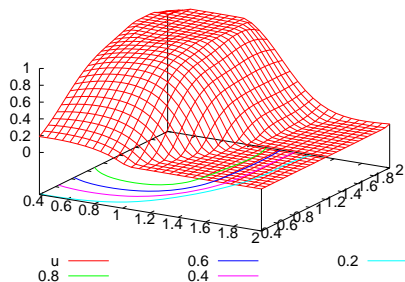
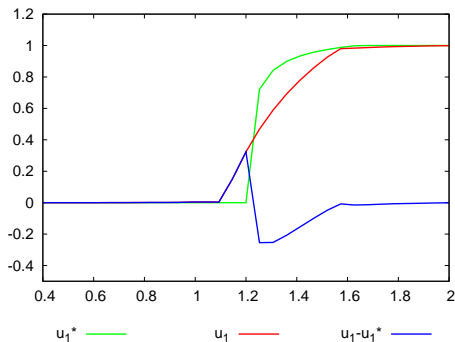
On cherche à résoudre le même problème que précédemment mais sur deux pas de temps. Il y a donc deux prix ξ^1 et ξ^2 , le deuxième n'étant observable qu'au deuxième pas de temps. Ainsi, la commande lors du premier pas de temps ne doit dépendre que de ξ^1 et $u(\xi^1, \xi^2) = (u^1(\xi^1), u^2(\xi^1, \xi^2))$. La fonction de coût est donc :

$$-\xi^1 u^1(\xi^1) - \xi^2 u^2(\xi^1, \xi^2) - \sqrt{\eta + s - u^1(\xi^1) - u^2(\xi^1, \xi^2)}$$

De plus on impose à la fonction $u : \forall \xi, u^1(\xi^1) \in [0, s]$ et $u^2(\xi^1, \xi^2) \in [0, s - u^1(\xi^1)]$. Cette dernière contrainte rend particulièrement difficile la projection et est donc pénalisée.

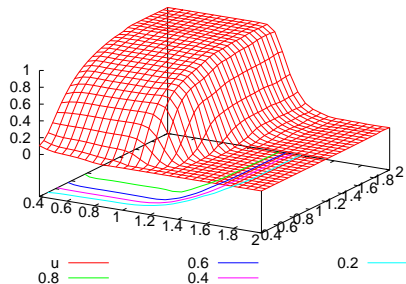
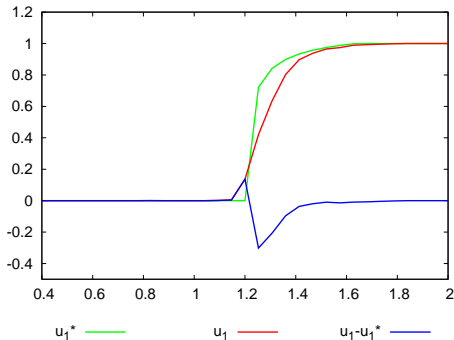
Gestion d'une réserve hydraulique (2 pas de temps)

1000 itérations



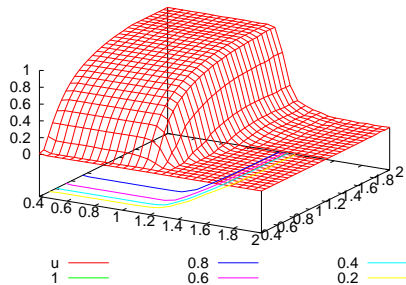
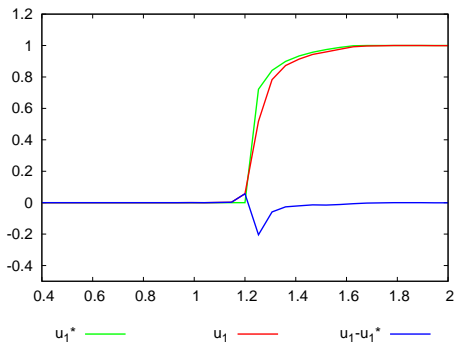
Gestion d'une réserve hydraulique (2 pas de temps)

10000 itérations



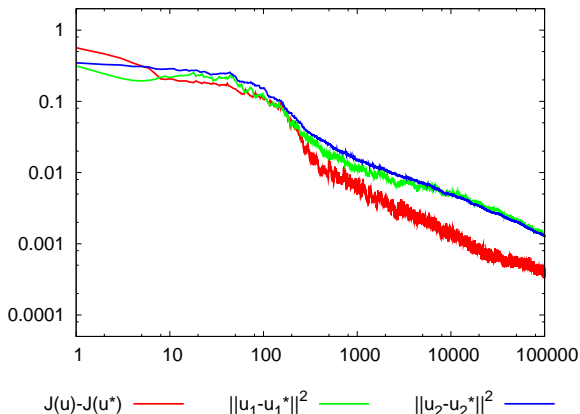
Gestion d'une réserve hydraulique (1 pas de temps)

100000 itérations



Gestion d'une réserve hydraulique (1 pas de temps)

Vitesse de convergence



Modélisation d'un système dynamique

Etat, Commande, Aléa

Typiquement, un système dynamique est caractérisé, pour chaque pas de temps t par son état $x_t \in X$. Cet état évolue en fonction de commandes $u_t \in U$, et d'aléas i.i.d. $w_t \in W$ non observés au moment de prendre la commande, selon une dynamique :

$$x_{t+1} = f(x_t, u_t, w_t)$$

Un coût est observé en fonction de l'état, la commande et l'aléa :

$$g(x_t, u_t, w_t)$$

Le choix des commandes est effectué en fonction de l'état selon une stratégie π :

$$u_t = \pi(x_t)$$

Modélisation d'un système dynamique

Optimisation du coût total en horizon infini

Le coût total du système pour une stratégie donnée est alors, compte tenu d'un taux d'actualisation $0 < \alpha < 1$ fixé :

$$J_{\pi}(x) = \mathbb{E}_w \left[\sum_{t=0}^{\infty} \alpha^t g(x_t, u_t, w_t) \mid x_0 = x, \pi \right]$$

L'objectif est alors de déterminer la meilleure stratégie pour minimiser le coût total du système :

$$J^*(x) := \min_{\pi \in U^X} J_{\pi}(x)$$

Exemple

Optimisation du coût total en horizon infini

Par exemple, il pourra s'agir de minimiser le coût de gestion d'un parc électrique (avec g , le coût de gestion) sur le futur, en agissant sur les production hydrauliques u , compte tenu d'une part d'aléas de prix et d'apports $(w_k)_k$, mais aussi des taux d'intérêts α , l'état du système (niveau des stocks, prix courant, ...) étant caractérisé par x .

Exemple

Gestion d'une réserve face à un marché

Prenons un exemple très simplifié, de gestion d'une réserve face à un marché. Notons $s_t \in [0, \bar{s}]$ le stock, p_t le prix de marché, $a_t \geq 0$ l'apport et $u_t \in [0, \bar{u}]$ le turbiné. Supposons que les prix et les apports sont i.i.d.

L'état du système s'écrit $x_t = (s_t, p_t)$. Les aléas s'écrivent $w_t = (q_t, a_t)$ où q_t suit la loi du prix. Ainsi :

$$f_t(x_t, u_t, w_t) = (\min(s_t - \min(u_t, s_t) + a_t, \bar{s}), q_t)$$

$$g(x_t, u_t) = -p_t \min(u_t, s_t)$$

Programmation dynamique

Compte tenu de l'horizon de temps infini, ce type de problème est difficilement abordable par la méthode du gradient stochastique, et se résoud en général à l'aide de la programmation dynamique. En effet :

$$J^*(x) = \min_{u \in U} \mathbb{E}_w (g(x, u, w) + \alpha J^*(f(x, u, w)))$$

On reconnaît ici l'équation de la programmation dynamique.

Soit T l'opérateur :

$$T(J)(\cdot) = \min_{u \in U} \mathbb{E}_w (g(\cdot, u, w) + \alpha J(f(\cdot, u, w)))$$

Le problème à résoudre est alors un problème de point fixe :

$$J^* = T(J^*)$$

Programmation dynamique

Obtention de la stratégie optimale

La stratégie π n'intervient plus dans le calcul du coût optimal précédent.

Notons que quand ce coût optimal est connu, la stratégie optimale peut être calculée ainsi :

$$\pi^*(x) = \arg \min_{u \in U} \mathbb{E}_w (g(x, u, w) + \alpha J^*(f(x, u, w)))$$

On ne s'intéresse donc qu'au coût optimal.

Itération de la valeur

Comme pour la plupart des problèmes de point fixe, la résolution s'effectue alors en général en itérant l'application de l'opérateur T partir d'un point de départ $J_0(\cdot)$ quelconque :

$$J_n = T(J_{n-1})$$

Soit :

$$\forall x, J_n(x) = \min_{u \in U} \mathbb{E}_w (g(x, u, w) + \alpha J_{n-1}(f(x, u, w)))$$

Cette méthode est appelée itération de la valeur.

Itération de la valeur

Cas fini

Dans le cas où l'espace d'état X et l'espace des aléas W sont finis, J est un vecteur indicé par l'état x , et le calcul s'effectue aisément (à supposer les cardinaux de X et W petits) :

$$J_n(x) = \min_{u \in U} \sum_{w \in W} P_w [g(x, u, w) + \alpha J_{n-1}(f(x, u, w))]$$

où P_w est la probabilité d'observer w . Et dans ce cas, l'algorithme converge :

$$\forall x \in X, \lim_{n \rightarrow \infty} J_n(x) = J^*(x)$$

Malheureusement ce calcul n'est pas implémentable dans le cas où X ou W ne sont pas finis... ou de cardinal très grand tels que ceux obtenus en discrétisant finement un espace continu de grande dimension.

Stratégie fixée

Intéressons-nous dorénavant au cas où la stratégie π est fixée. Notons alors $f_\pi(x, w) = f(x, \pi(x), w)$ et $g_\pi(x, w) = g(x, \pi(x), w)$. Dans ce cas, on cherche à calculer J_π tel que :

$$J_\pi(x) = \mathbb{E}_w (g_\pi(x, w) + \alpha J_\pi(f_\pi(x, w)))$$

qui est aussi un problème de point fixe, mais cette fois-ci, l'opérateur T ne fait plus intervenir de minimisation et devient linéaire, ce qui simplifie la résolution :

$$T(J)(\cdot) = \mathbb{E}_w (g_\pi(\cdot, w) + \alpha J(f_\pi(\cdot, w)))$$

Dynamique invariante

Notons que ce type de problème peut apparaître même si la stratégie n'est pas fixée, par exemple dans le cas où la dynamique f ne dépend pas de la commande prise u . C'est typiquement le cas des décisions prises sur les marchés financiers où les actions des agents individuels ne modifient pas les prix. Dans ce cas, si de plus le coût est indépendant de l'aléa, en notant $g_\pi(x) = \min_{u \in U} g(x, u)$, on a :

$$J_\pi(x) = \mathbb{E}_w (g_\pi(x) + \alpha J_\pi(f(x, w)))$$

qui est un problème similaire au précédent.

Problème de Robbins-Monro

Soit l'opérateur :

$$t(J, w)(\cdot) = g_{\pi}(\cdot, w) + \alpha J(f_{\pi}(\cdot, w))$$

Notre problème de point fixe s'écrit donc, trouver une fonction J telle que $J = T(J)$ où :

$$T(J) = \mathbb{E}_w(t(J, w))$$

C'est un problème de point fixe sur lequel on peut appliquer l'algorithme de Robbins-Monro.

Algorithme de Robbins-Monro

A stratégie fixée, tout comme pour l'algorithme du gradient stochastique, il est possible de réaliser une estimation de l'espérance $\mathbb{E}_w (g_\pi(x, w) + \alpha J_\pi(f_\pi(x, w)))$ par des tirages $(w_n)_n$, puis de moyenner les valeurs obtenues. Notons :

$$\Delta_{n-1, \pi}(x, w) = g_\pi(x, w) + \alpha J_{n-1}(f_\pi(x, w)) - J_{n-1}(x)$$

L'algorithme est alors (dans le cas où X est un espace fini) :

$$\forall x \in X, J_n(x) = J_{n-1}(x) + p_n \Delta_{n-1, \pi}(x, w_n)$$

Il s'agit de l'algorithme de Robbins-Monro [Robbins and Monro, 1951] :

$$J_n = J_{n-1} + p_n (t(J_{n-1}, w_n) - J_{n-1})$$

Différences temporelles

Alternativement, plutôt que de mettre à jour la fonction J pour tous les états x , il est possible de tirer aussi au hasard l'état $(x_n)_n$ à mettre à jour. Soit :

$$J_n(\cdot) = J_{n-1}(\cdot) + p_n \Delta_{n-1, \pi}(x_n, w_n) \delta_{x_n}(\cdot)$$

Cet algorithme est l'algorithme des différences temporelles [Sutton, 1988] dit TD(0). Il converge vers le coût de la stratégie π :

$$\forall x \in X, \lim_{n \rightarrow \infty} J_n(x) = J_\pi(x)$$

Algorithme TD(0) fonctionnel

Dans le cas où l'état est continu, il n'est plus possible d'implémenter l'algorithme précédent, pour lequel les mises à jours sont ponctuelles en x_n . Mais comme précédemment, il est possible d'approximer le Dirac $\delta_{x_n}(\cdot)$ par un noyau $\frac{1}{\epsilon_n} K_n(x_n, \cdot)$. On obtient alors l'algorithme de différences temporelles fonctionnel suivant :

$$J_n(\cdot) = J_{n-1}(\cdot) + \rho_n \frac{1}{\epsilon_n} \Delta_{n-1, \pi}(x_n, w_n) K_n(x_n, \cdot)$$

Cette généralisation permet de s'affranchir de la paramétrisation de la fonction de Bellman. Nous avons montré que cet algorithme converge [Barty et al., 2005c] à l'aide de la théorie sur le gradient perturbé déjà exposée. [Chen and White, 1998] propose une approche relativement similaire.

Programmation dynamique approximée

Règles linéaires pour la fonction de Bellman

Alternativement, tout comme pour les problèmes d'optimisation en boucle fermée présentés précédemment, une solution classique consiste à paramétrer la fonction J . Il s'agit de la programmation dynamique approximée [Bellman and Dreyfus, 1959]. Dans ce cas, soit $A = (a_i)_i$ le vecteur de paramètres :

$$J(x) = \sum_{i=0}^m a_i j_i(x) = A \cdot (j_i(x))_i$$

où $(j_i)_i$ est une famille prédéfinie de fonctions de l'état.

Programmation dynamique approximée

Algorithme de mise à jour

La mise à jour s'effectue sur les paramètres A alors par :

$$A_n = A_{n-1} + p_n \Delta_{n-1, \pi}(x_n, w_n) \nabla_A J(x_n)$$

Un cas particulier de paramétrage consiste à discrétiser l'état du système en une partition $X = \bigcup_i \mathcal{P}_i$: c'est la solution classiquement utilisée à EdF. Dans ce cas, $(j_i(x))_i = \nabla_A J(x) = (\mathbf{1}_{\mathcal{P}_i}(x))_i$.

Ce type de discrétisation regroupe le cas légèrement plus sophistiqué de l'agrégation des réserves : une telle solution est mise en œuvre dans le MNR⁴.

⁴Modèle National de Régulation de gestion des réserves hydrauliques d'EDF à l'horizon annuel.

Recherche de la stratégie optimale

Problème lié à la minimisation

Nous avons vu que nous pouvons résoudre les problèmes de point fixe du type :

$$T(J) = \mathbb{E}_w(t(J, w))$$

en approximant cette espérance par un tirage. Malheureusement, dans le cas général avec recherche de la stratégie optimale, l'équation de Bellman s'écrit sous la forme :

$$T(J) = \min_u \mathbb{E}_w(t(J, u, w))$$

Il n'est alors pas aisé d'estimer $T(J)$ à l'aide d'un tirage car :

$$\min_u \mathbb{E}_w(t(J, u, w)) \neq \mathbb{E}_w\left(\min_u t(J, u, w)\right)$$

Recherche de la stratégie optimale

Q-Fonction

Néanmoins, on peut souvent se ramener à une situation similaire à la précédente en définissant :

$$Q(x, u) = \mathbb{E}_w (g(x, w, u) + \alpha J(f(x, u, w)))$$

d'où :

$$J(x) = \min_u Q(x, u)$$

Recherche de la stratégie optimale

Q-Learning

A partir de l'équation de la programmation dynamique, cette définition nous amène à l'équation suivante :

$$Q(x, u) = \mathbb{E}_w \left(g(x, w, u) + \alpha \min_{u'} Q(f(x, u', w), u') \right)$$

Celle-ci pouvant dans certain cas être résolue en effectuant comme précédemment des tirages sur w , x , et u . Ce problème est néanmoins significativement plus complexe que le précédent, et nous ne sommes pas encore en mesure de le traiter dans le cas général.

Problème linéaire-quadratique en horizon infini

Description du problème

Dans la suite, on note J la fonction de Bellman.

Soit $(x_t)_t$ un processus auto-régressif :

$$x_t = \gamma x_{t-1} + \eta_t$$

On s'intéresse à un problème de programmation dynamique dont la fonction coût de transition est $g(x) = x^2$, et le taux d'actualisation α , i.e. :

$$J(x) = \mathbb{E} \left(\sum_{t \geq 0} \alpha^t x_t^2 \mid x_0 = x \right)$$

Problème linéaire-quadratique en horizon infini

Equation de Bellman et mise à jour

L'équation de Bellman s'écrit donc :

$$J(x) = \mathbb{E} (x_0^2 + \alpha J(x_1) \mid x_0 = x)$$

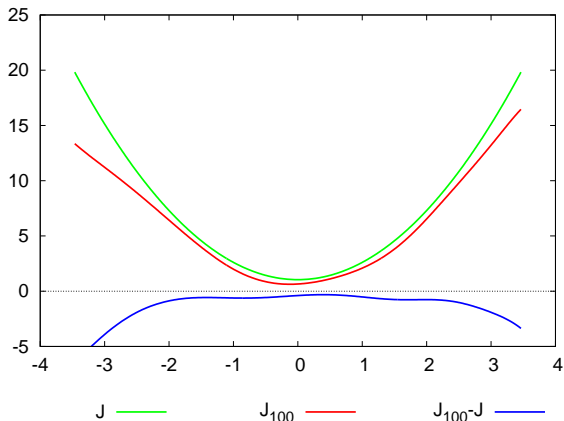
Et la mise à jour s'effectue par :

$$J_n(\cdot) = J_{n-1}(\cdot) + p_n (x_n^2 + \alpha J_{n-1}(x_{n+1}) - J_{n-1}(x_n)) \frac{1}{\epsilon_n} K_n(x_n, \cdot)$$

Notons qu'en toute rigueur, le tirage de x_{n+1} utilisé ci-dessus ne devrait pas être utilisé pour la mise à jour à l'étape $n + 1$.

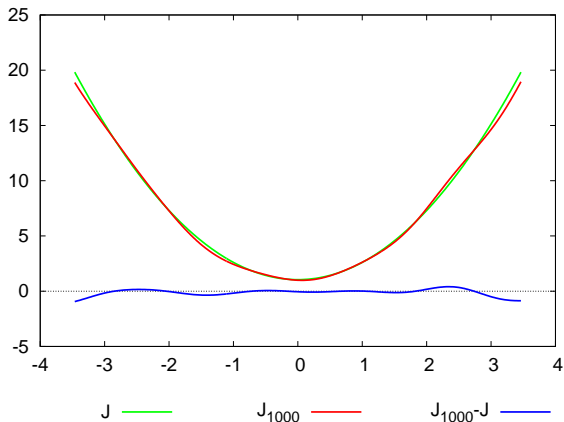
Problème linéaire-quadratique en horizon infini

100 itérations



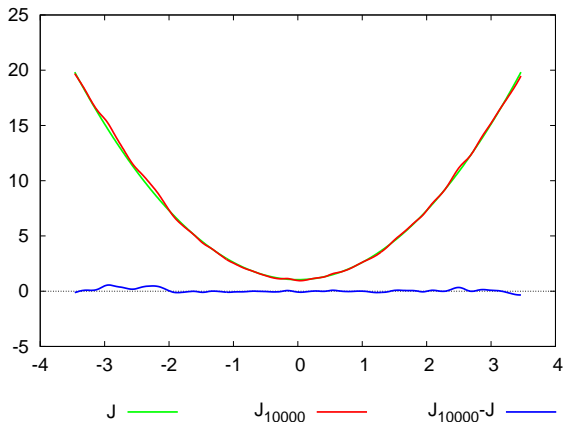
Problème linéaire-quadratique en horizon infini

1000 itérations



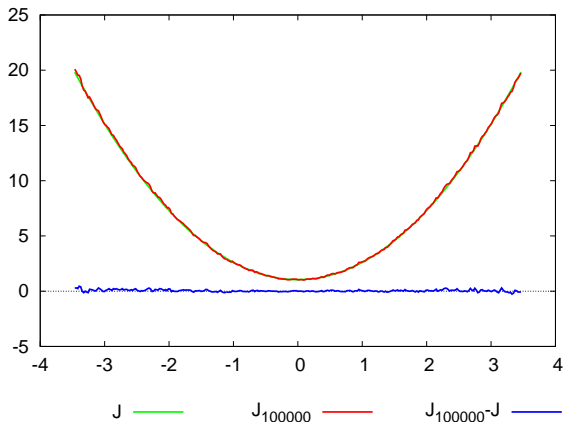
Problème linéaire-quadratique en horizon infini

10000 itérations



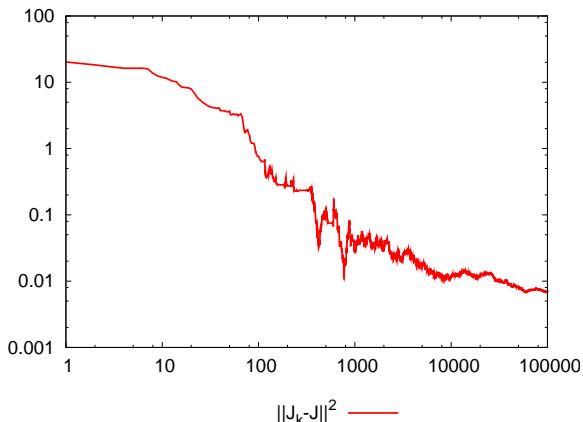
Problème linéaire-quadratique en horizon infini

100000 itérations



Problème linéaire-quadratique en horizon infini

Vitesse de convergence



Valorisation d'une option bermudéenne

Description du problème

De manière similaire à [Roy and Tsitsiklis, 2001], on s'intéresse au problème de valorisation d'une option de vente Bermudéenne où les dates d'exercice sont restreintes à des dates equi-espacées t dans $0, \dots, T$. Le prix du sous-jacent x_t suit une dynamique de Black-Scholes [Black and Scholes, 1973] discrétisée :

$$\ln \frac{x_{t+1}}{x_t} = r - \frac{1}{2}\sigma^2 + \sigma\eta_t$$

où (η_t) est un bruit blanc gaussien de variance unitaire et r est le taux d'intérêt sans risque. Le prix d'exercice est s , ainsi la valeur intrinsèque de l'option est $g(x) = \max(0, s - x)$ quand le prix est x . Définissons le taux d'actualisation $\alpha = e^{-r}$.

Valorisation d'une option bermudéenne

Objectif

Soit x_0 le prix à $t = 0$, notre objectif est de calculer la valeur de l'option :

$$\max_{\tau} \mathbb{E} (\alpha^{\tau} g (x_{\tau}))$$

où τ est pris parmi les temps d'arrêt adaptés à la filtration engendrée par le processus de prix $(x_t)_t$.

Soit $J_t(x)$ la valeur de l'option au temps t si le prix x_t est égal à x . Comme l'option doit être exercée à avant $T + 1$, on a : $J_{T+1}(x) = 0$. Donc, pour tout $t \leq T$:

$$J_t(x) = \max (g(x), \alpha \mathbb{E} (J_{t+1}(x_{t+1}) | x_t = x))$$

Valorisation d'une option bermudéenne

Q fonction

Nous appliquons alors l'itération de la valeur à la fonction Q définie par :

$$Q_t(x) = \alpha \mathbb{E} (J_{t+1}(x_{t+1}) | x_t = x)$$

i.e. le gain espéré à t si l'on n'exerce pas l'option.

Pour tout temps t la valeur de l'option est donnée par

$$J_t(x) = \max(g(x), Q_t(x)).$$

Comme $J_{T+1}(x) = 0$, on a $Q_T(x) = 0$. Nous résolvons donc l'équation de programmation dynamique pour tout x , et tout $t < T$ suivante :

$$Q_t(x) = \alpha \mathbb{E} (\max(g(x_t), Q_{t+1}(x_{t+1})) | x_t = x)$$

Valorisation d'une option bermudéenne

Résolution

En notant, $Q = (Q_t)_t$, le problème s'écrit sous la forme
 $Q = H(Q)$ où :

$$(H(Q)(y))_t = \mathbb{E}(\alpha \max(g(x_t), Q_{t+1}(x_{t+1})) | x_t = y_t)$$

Nous appliquons alors l'algorithme des différences temporelles. Les échantillons $(x^n)_n$ où $x^n = (x_t^n)_t$ sont issus de trajectoires de prix.

Valorisation d'une option bermudéenne

Mise en œuvre

En pratique, comme Q_{t+1}^n est disponible lors du calcul de Q_t^n , la mise à jour s'effectue par :

$$Q_t^n(\cdot) = Q_t^{n-1}(\cdot) + p_n \Delta_{n-1}(x_t^n, x_{t+1}^n) \frac{1}{\epsilon_n} K_n(x_t^n, \cdot)$$

avec :

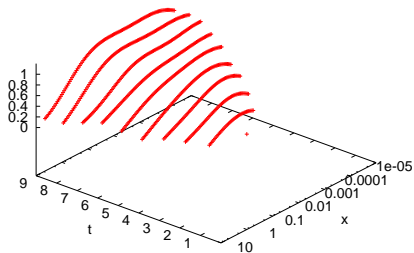
$$\Delta_{n-1}(x, x') = \alpha \max(g(x), Q_{t+1}^n(x')) - Q_t^{n-1}(x)$$

On peut rapprocher ce type d'approches à l'approche non-paramétrique de [Longstaff and Schwartz, 2001].

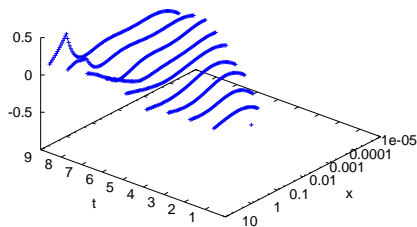
Nous prenons pour l'expérience : $\mu = 1$, $\sigma = 1$, $s = 1$, $x_0 = 1$ et $r = 0.01$ (et donc $\alpha = 0.99$).

Valorisation d'une option bermudéenne

100 itérations



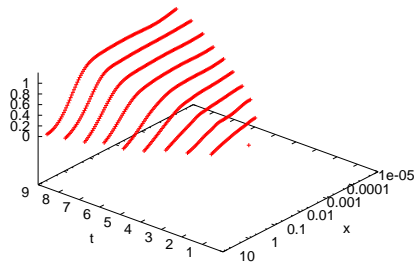
Q^{100}



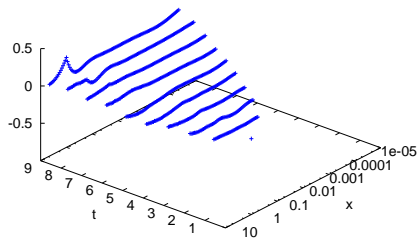
$Q^{100} - Q^*$

Valorisation d'une option bermudéenne

1000 itérations



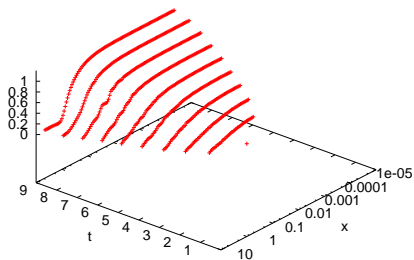
Q^{1000}



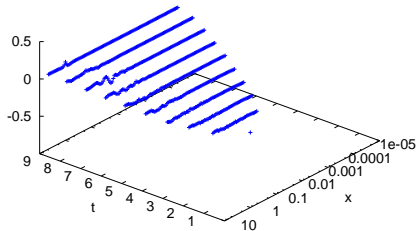
$Q^{1000} - Q^*$

Valorisation d'une option bermudéenne

10000 itérations



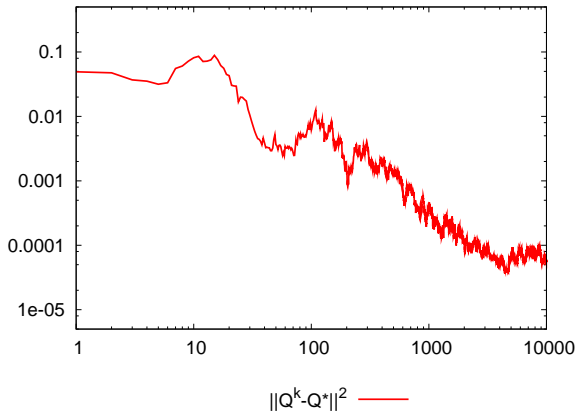
Q^{10000}



$Q^{10000} - Q^*$

Valorisation d'une option bermudéenne

Vitesse de convergence



Conclusion

Nous avons présenté :

- 1 Un nouveau théorème de convergence pour la méthode du gradient perturbé ;
- 2 Une méthode de résolution de problèmes stochastiques en boucle fermée ;
- 3 Une méthode de résolution de problèmes de programmation dynamique.

Ces deux méthodes sont simples à mettre en œuvre, et convergent vers la solution optimale des problèmes.




Extensions

De nombreuses extensions sont possibles. Notamment :





- Approximer le gradient par différences finies (Méthodes des perturbations simultanées[Spall, 1992]) ;
- Accélérer la convergence en prenant des pas plus grands puis en moyennant le résultat[Polyak and Juditsky, 1992] ;
- Déterminer de meilleures heuristiques pour le pas et la fenêtre.

Enfin, il serait certainement intéressant de tester l'algorithme sur un problème de grande taille (le choix du noyaux risque de s'avérer important).




Bibliographie I

-  Barty, K., Roy, J.-S., and Strugarek, C. (2005a).
A perturbed gradient algorithm in hilbert spaces.
Optimization Online.
http://www.optimization-online.org/DB_HTML/2005/03/1095.html.
-  Barty, K., Roy, J.-S., and Strugarek, C. (2005b).
A stochastic gradient type algorithm for closed loop problems.
submitted to SPEPS.
-  Barty, K., Roy, J.-S., and Strugarek, C. (2005c).
Temporal difference learning with kernels.
unpublished.

Bibliographie II

-  Bellman, R. and Dreyfus, S. (1959).
Functional approximations and dynamic programming.
Math tables and other aides to computation, 13 :247–251.
-  Bertsekas, D. and Tsitsiklis, J. (2000).
Gradient convergence in gradient methods.
SIAM Journal in Optimization, 10(3) :627–642.
-  Black, F. and Scholes, M. (1973).
The pricing of options and corporate liabilities.
Journal of Political Economy, 81(3) :637–654.
-  Chen, X. and White, H. (1998).
Nonparametric learning with feedback.
Journal of Economic Theory, 82 :190–222.

Bibliographie III

-  Delyon, B. (1996).
General results on the convergence of stochastic algorithms.
IEEE Transactions on Automatic and Control,
41(9) :1245–1255.
-  Holt, C., Modigliani, F., and Simon, H. (1955).
A linear decision rule for production and employment
scheduling.
Management Science, 2(1).
-  Kiefer, J. and Wolfowitz, J. (1952).
Stochastic estimation of the maximum of a regression
function.
Annals of Mathematical Statistics, 23 :462–466.

Bibliographie IV



Kulkarni, S. and Horn, C. (1996).

An alternative proof for convergence of stochastic approximation algorithms.

IEEE Transactions on Automatic Control, 41(3) :419–424.



Kushner, H. and Clark, D. (1978).

Stochastic Approximation for Constrained and Unconstrained Systems.

Springer-Verlag.






Longstaff, F. A. and Schwartz, E. S. (2001).



Valuing american options by simulation : A simple least squares approach.

Rev. Financial Studies, 14(1) :113–147.

Bibliographie V

-  Polyak, B. T. and Juditsky, A. B. (1992).
Acceleration of stochastic approximation by averaging.
SIAM Journal on Control and Optimization, 30 :838–355.
-  Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
Annals of Mathematical Statistics, 22 :400–407.
-  Roy, B. V. and Tsitsiklis, J. N. (2001).
Regression methods for pricing complex american-style options.
IEEE Trans. on Neural Networks, 12(4) :694–703.

Bibliographie VI

-  Spall, J. (1992).
Multivariate stochastic approximation using a simultaneous perturbation gradient approximation.
IEEE Transactions on Automatic Control, 37 :332–341.
-  Sutton, R. S. (1988).
Learning to predict by the method of temporal differences.
Machine Learning, 3 :9–44.