OPTIMISATION ET CONTRÔLE

Grégoire Allaire
Département de mathématiques appliquées
Ecole Polytechnique

Algorithmes pour l'optimisation sous contraintes

Algorithmes d'optimisation avec contraintes.

Pour un sous-ensemble $K \subset V$ d'un espace de Hilbert (qui représente des contraintes), on veut résoudre

$$\inf_{v \in K} J(v) \, .$$

1. Gradient projeté pour un convexe K.
2. Algorithme d'Uzawa.
3. Pénalisation des contraintes.
4. Méthode du Lagrangien augmenté.
5. Méthodes (ou stratégies) d'approximations successives.
6. Quelques algorithmes utilisant la structure du problème.

Soit K convexe fermé non vide de V. On veut résoudre

$$\inf_{v \in K} J(v)\,.$$

Initialisation: choisir $u^0 \in K$ (par exemple $u^0 = P_K(0)$).
Itérations: pour $n \geq 0$

$$u^{n+1} = P_K\big(u^n - \mu\, J'(u^n)\big)\ ,$$

avec P_K opérateur de projection orthogonale sur K.

Théorème. On suppose que J est α-convexe différentiable ($\alpha > 0$)

$$J(v) \geq J(u) + \langle J'(u), v - u\rangle + \frac{\alpha}{2}\|v - u\|^2 \qquad \forall\, u, v \in V$$

et que J' est Lipschitzien: il existe $L > 0$ tel que

$$\big\|J'(v) - J'(u)\big\| \leq L\|v - u\| \qquad \forall\, u, v \in V\ .$$

Alors, si $0 < \mu < 2\alpha/L^2$, l'algorithme de gradient projeté converge.

Rappel

Théorème. Soit V un espace de Hilbert. Soit $K \subset V$ un convexe fermé non vide. Pour tout $x \in V$, il existe un unique $x_K \in K$ tel que

$$\|x - x_K\| = \min_{y \in K} \|x - y\|.$$

De façon équivalente, x_K est caractérisé par la propriété

$$x_K \in K, \langle x_K - x, x_K - y \rangle \leq 0 \quad \forall y \in K.$$

On appelle P_K l'opérateur de projection orthogonale sur K défini par

$$P_K(x) = x_K.$$

Preuve. Il existe un unique point de minimum $u \in K$.
Condition d'optimalité (inéquation d'Euler)

$$\langle J'(u), v - u \rangle \geq 0 \quad \forall\, v \in K\ .$$

Pour $\mu > 0$, c'est équivalent à

$$\langle u - \big(u - \mu J'(u)\big), v - u \rangle \geq 0 \quad \forall\, v \in K\ .$$

Autrement dit, $u \in K$ est caractérisé par

$$u = P_K\big(u - \mu J'(u)\big) \qquad \text{avec } P_K \text{ projection orthogonale sur } K.$$

On va montrer que, pour $0 < \mu < 2\alpha/L^2$, l'application

$$f(v) = P_K\big(v - \mu J'(v)\big)$$

est strictement contractante. Donc elle admet un unique point fixe $u = f(u)$ et la suite

$$u^{n+1} = P_K\big(u^n - \mu\, J'(u^n)\big) \qquad \text{converge vers } u.$$

Preuve (suite)

L'opérateur de projection P_K est faiblement contractant

$$\|P_K v - P_K w\| \leq \|v - w\|.$$

En effet, si on développe

$$\|P_K v - P_K w\|^2 = \langle P_K v - P_K w, v - w\rangle + \langle P_K v - P_K w, P_K v - v\rangle$$

$$+ \langle P_K v - P_K w, w - P_K w\rangle$$

$$\leq \langle P_K v - P_K w, v - w\rangle \leq \|P_K v - P_K w\| \|v - w\|$$

car par définition de $P_K v$ et de $P_K w$

$$\langle P_K v - v, P_K v - z\rangle \leq 0 \quad \forall z \in K \text{ (prendre } z = P_K w)$$

$$\langle P_K w - w, P_K w - z\rangle \leq 0 \quad \forall z \in K \text{ (prendre } z = P_K v)$$

Preuve (fin)

La fonction $v \to v - \mu J'(v)$ est strictement contractante. En effet,

$$\|v - \mu J'(v) - w + \mu J'(w)\|^2 = \|v - w\|^2 - 2\mu\langle v - w, J'(v) - J'(w)\rangle + \mu^2\|J'(v) - J'(w)\|^2$$

Grâce à l'α-convexité de J

$$\langle v - w, J'(v) - J'(w)\rangle \geq \alpha\|v - w\|^2$$

ainsi que J' Lipschitz

$$\|J'(v) - J'(w)\|^2 \leq L^2\|v - w\|^2$$

on obtient

$$\|v - \mu J'(v) - w + \mu J'(w)\|^2 \leq (1 - 2\mu\alpha + \mu^2 L^2)\|v - w\|^2$$

Si $0 < \mu < \frac{2\alpha}{L^2}$, alors $(1 - 2\mu\alpha + \mu^2 L^2) < 1$ et on conclut.

Exemples d'opérateurs de projection P_K

- Si $V = \mathbb{R}^n$ et $K = \prod_{i=1}^n [a_i, b_i]$, alors pour $x = (x_1, \dots, x_n)$

  $P_K(x) = y \quad$ avec $\quad y_i = \min\left(\max\left(a_i, x_i\right), b_i\right) \quad$ pour $1 \leq i \leq n$.

- Si $V = \mathbb{R}^n$, B matrice $m \times n$ avec $\mathrm{rg}(B) = m \leq n$ et $K = \{x \in \mathbb{R}^n \; Bx = c\}$, alors

  $$P_K(x) = x + B^*(BB^*)^{-1}(c - Bx).$$

  En effet, la condition d'optimalité implique que $(x - P_K(x))$ est orthogonal à $\ker B$ donc appartient à $\mathrm{Im} B^*$, c'est-à-dire $x - P_K(x) = B^* y$. Enfin $y = (BB^*)^{-1}(Bx - c)$ à cause de la condition $BP_K(x) = c$.

Remarque. Pour des convexes fermés K plus généraux, P_K peut être très difficile (en pratique, impossible) à déterminer.

Ensemble de contraintes non convexe

On veut résoudre

$$\inf_{v \in K} J(v)\,,$$

où K est un sous-ensemble fermé non vide, non-convexe de V.

Projeter sur un convexe n'est déjà pas facile...

Mais si K n'est pas convexe, c'est encore pire ! Pour ne pas dire impossible...

Algorithme faisable ou infaisable ?

Pour un problème d'optimisation sous contrainte

$$\inf_{v\in K} J(v) \,,$$

il existe deux types d'algorithmes:

1. faisable si la suite d'itérées u^n appartient à K,
2. infaisable si la suite d'itérées u^n n'appartient pas à K (mais converge vers une limite dans K).

L'algorithme du gradient projeté est faisable mais pour K "compliqué" on ne sait pas calculer l'opérateur P_K.

On va étudier d'autres algorithmes, comme celui d'Uzawa, qui est infaisable mais facile à mettre en oeuvre.

2 - Algorithme d'Uzawa

L'algorithme d'Uzawa recherche un point selle du Lagrangien $\mathcal{L}(v,q) = J(v) + q \cdot F(v)$ pour le problème de minimisation

$$\min_{F(v) \leq 0} J(v) \ ,$$

Rappel (Théorème de Kuhn et Tucker). Si J et F sont convexes et différentiables, alors il existe un point de minimum u avec multiplicateur de Lagrange p si et seulement si (u,p) est un point-selle du Lagrangien

$$\forall\, q \in \mathbb{R}_+^M \qquad \mathcal{L}(u,q) \leq \mathcal{L}(u,p) \leq \mathcal{L}(v,p) \qquad \forall\, v \in V \ .$$

Idée de l'algorithme: alternativement on minimise en v à q fixé, et on maximise en q à v fixé.

Algorithme d'Uzawa (2)

Intuition (heuristique) du Lagrangien pour l'optimisation sous contraintes. On veut résoudre

$$\min_{F(v)\leq 0} J(v)$$

A la place on minimise le Lagrangien, pour $q \geq 0$ fixé,

$$v \ \rightarrow \ \mathcal{L}(v,q) = J(v) + q \cdot F(v)$$

Deux cas de figure: (une seule contrainte, $M = 1$, pour simplifier)

1. si $F(v) < 0$, alors $q = 0$ et on minimise seulement $J(v)$,
2. si $F(v) \geq 0$, alors $q \geq 0$ et on minimise une combinaison de J et F pour à la fois rendre la contrainte satisfaite et diminuer la fonction objectif.

Evidemment, il faut modifier q itérativement pour tenir compte de la contrainte, active ou pas. Si on maximise $q \ \rightarrow \ \mathcal{L}(v,q)$, on augmente q lorsque $F(v) \geq 0$, pour améliorer la satisfaction de la contrainte.

Algorithme d'Uzawa (3)

Initialisation: soit $p^0 \in (\mathbb{R}_+)^M$.
Itérations: on construit les suites (u^n) et (p^n) par

$$\mathcal{L}(u^n, p^n) = \inf_{v \in V} \mathcal{L}(v, p^n) \; ,$$

$$p^{n+1} = P_{\mathbb{R}_+^M}(p^n + \mu F(u^n)) \; ,$$

avec $\mu > 0$ fixé (c'est une itération de l'algorithme du gradient projeté pour maximiser en q).

Remarques.

1. Variante: algorithme d'Arrow-Hurwicz (un seul pas de gradient pour la minimisation en v).
2. On n'est pas garanti d'avoir $F(u^n) \leq 0$ (algorithme infaisable).
3. Algorithme de maximisation du problème dual !
4. Peut être mis en oeuvre pour des contraintes d'égalité.

Algorithme d'Uzawa (4)

Lien avec la dualité. Le problème dual est

$$\sup_{p\in\mathbb{R}^M_+} \left\{ \mathcal{G}(p) = \inf_{v\in V} \mathcal{L}(v,p) \right\}$$

Supposons qu'il existe un unique $v(p)$ tel que

$$\mathcal{G}(p) = \mathcal{L}(v(p),p) \quad \text{et} \quad p \to v(p) \text{ est dérivable.}$$

Comme $\mathcal{L}(v,p) = J(v) + p\cdot F(v)$, la condition d'optimalité donne

$$\mathcal{G}'(p) = \langle \underbrace{J'(v(p)) + p\cdot F'(v(p))}_{=0}, v'(p)\rangle + F(v(p)) = F(v(p)).$$

Donc l'algorithme d'Uzawa est

$$\mathcal{L}(u^n,p^n) = \inf_{v\in V} \mathcal{L}(v,p^n) = \mathcal{G}(p^n)$$
$$p^{n+1} = P_{\mathbb{R}^M_+}(p^n + \mu\mathcal{G}'(p^n))$$

l'algorithme du gradient projeté pour maximiser le dual !

Algorithme d'Uzawa (5)

Théorème. On suppose que J est α-convexe différentiable, que F est convexe et Lipschitzienne de V dans $\mathbb{R}^M$, $\exists\, L > 0$ tel que

$$\|F(v) - F(w)\| \leq L\|v - w\| \quad \forall\, v, w \in V\,,$$

que l'ensemble $\{F(v) \leq 0\}$ est non vide et que les contraintes sont qualifiées en l'unique point de minimum u de J sous les contraintes $F \leq 0$. Si $0 < \mu < 2\alpha/L^2$, alors l'algorithme d'Uzawa converge : $\forall p^0 \in (\mathbb{R}_+)^M$, la suite (u^n) converge vers u.

Lemme (technique). Si J_1 et J_2 sont deux fonctions convexes dont seule J_1 est différentiable, alors la condition d'optimalité (nécessaire et suffisante) pour que u soit un minimum de $J_1 + J_2$ sur un convexe fermé K est

$$\langle J_1'(u), v - u\rangle + J_2(v) - J_2(u) \geq 0 \quad \text{pour tout } v \in K.$$

Algorithme d'Uzawa (6)

Preuve du théorème. On applique le lemme à J et F pour u et u^n (avec leurs multiplicateurs de Lagrange p et p^n)

$$(1) \qquad \langle J'(u), v-u\rangle + p\cdot\big(F(v)-F(u)\big) \geq 0 \quad \forall\, v\in V$$

$$(2) \qquad \langle J'(u^n), v-u^n\rangle + p^n\cdot\big(F(v)-F(u^n)\big) \geq 0 \quad \forall\, v\in V$$

On prend $v=u^n$ dans (1) et $v=u$ dans (2) et on additionne

$$\langle J'(u)-J'(u^n), u^n-u\rangle + (p-p^n)\cdot\big(F(u^n)-F(u)\big) \geq 0\,,$$

d'où par α-convexité de J et en posant $r^n=p^n-p$

$$r^n\cdot\big(F(u^n)-F(u)\big) \leq -\alpha\|u^n-u\|^2\,.$$

D'autre part, on soustrait

$$p^{n+1} = P_{\mathbb{R}^M_+}(p^n+\mu F(u^n)) \quad \text{à} \quad p = P_{\mathbb{R}^M_+}(p+\mu F(u))$$

la projection $P_{\mathbb{R}^M_+}$ étant faiblement contractante, on obtient

$$\|r^{n+1}\| \leq \|r^n+\mu(F(u^n)-F(u))\|$$

En développant

$$\|r^{n+1}\| \leq \|r^n + \mu(F(u^n) - F(u))\|$$

on obtient

$$\|r^{n+1}\|^2 \leq \|r^n\|^2 + 2\mu r^n \cdot \big(F(u^n) - F(u)\big) + \mu^2 \|F(u^n) - F(u)\|^2 \,.$$

Utilisant $r^n \cdot \big(F(u^n) - F(u)\big) \leq -\alpha \|u^n - u\|^2$, il vient

$$\|r^{n+1}\|^2 \leq \|r^n\|^2 + \big(L^2\mu^2 - 2\mu\alpha\big)\|u^n - u\|^2 \,.$$

Si $0 < \mu < 2\alpha/L^2$, alors $2\mu\alpha - L^2\mu^2 > 0$, d'où

$$0 < (2\mu\alpha - L^2\mu^2)\|u^n - u\|^2 \leq \|r^n\|^2 - \|r^{n+1}\|^2 \,.$$

Ainsi la suite $\|r^n\|^2$ est décroissante, minorée, donc convergente, ce qui entraîne que u^n tend vers u.

Preuve du lemme technique

Comme K est convexe, $u + \theta(v - u) \in K$ pour $0 \leq \theta \leq 1$, et u minimum est équivalent

$$J_1(u + \theta(v - u)) + J_2(u + \theta(v - u)) \geq J_1(u) + J_2(u).$$

On fait un développement de Taylor pour J_1 et on utilise la convexité pour J_2

$$J_1(u)+\theta\langle J_1'(u), v-u\rangle+o(\theta)+(1-\theta)J_2(u)+\theta J_2(u) \geq J_1(u)+J_2(u).$$

On élimine les termes de droite et on divise par $\theta \to 0$

$$(1) \qquad \langle J_1'(u), v - u\rangle + J_2(v) - J_2(u) \geq 0$$

Réciproquement, (1) $\Rightarrow$ u minimum de $J_1 + J_2$ par convexité de J_1

$$J_1(v) - J_1(u) \geq \langle J_1'(u), v - u\rangle$$

Par exemple, on remplace

$$(P) \qquad \inf_{v\in\mathbb{R}^N,\ F(v)\leq 0} J(v)$$

par

$$(P_\varepsilon) \qquad \inf_{v\in\mathbb{R}^N} \left(J(v) + \frac{1}{\varepsilon}\sum_{i=1}^{M} \left[\max\left(F_i(v), 0\right)\right]^2 \right).$$

où on a **pénalisé** les contraintes avec un petit paramètre $\varepsilon > 0$.
On utilise ensuite un algorithme sans contrainte.

Proposition. Soit J continue, strictement convexe, infinie à l'infini, et $F(v)$ convexe, continue, avec $K = \{F(v) \leq 0\}$ non vide. Alors la suite des solutions u_ε du problème pénalisé (P_ε) converge vers l'unique solution u du problème (P) quand $\varepsilon \to 0$.

Pénalisation des contraintes (2)

Preuve. Solutions uniques u pour (P) et u_ε pour (P_ε). Soit

$$G(v) = \sum_{i=1}^{M} [\max(F_i(v), 0)]^2 \quad \text{et} \quad J_\varepsilon(v) = J(v) + \frac{G(v)}{\varepsilon}$$

Comme $G(u) = 0$, on a

$$(1) \qquad J(u_\varepsilon) \leq J_\varepsilon(u_\varepsilon) \leq J_\varepsilon(u) = J(u) \, .$$

Donc u_ε est borné dans $\mathbb{R}^N$ car J est infinie à l'infini. Une sous-suite u_ε converge vers une limite u_*. Or $0 \leq G(u_\varepsilon) \leq \varepsilon(J(u) - J(u_\varepsilon))$ donc $G(u_*) = 0$, et $F(u_*) \leq 0$. Comme (1) implique $J(u_*) \leq J(u)$, on en déduit $u_* = u$ et toute la suite converge.

Remarque. Si en plus on suppose une condition de qualification pour les contraintes de (P) alors on peut montrer la convergence des multiplicateurs de Lagrange

$$\lim_{\varepsilon \to 0} \left[\frac{2}{\varepsilon} \max(F_i(u_\varepsilon), 0) \right] = \lambda_i \, .$$

Quelques remarques d'ordre pratique.

- Très facile à mettre en oeuvre !
- En général, on applique une méthode de continuation, c'est-à-dire que la valeur de $\varepsilon > 0$ est diminuée au cours des itérations.
- Malgré cela, cette méthode de pénalisation souffre d'un problème numérique de conditionnement quand ε est petit.
- On peut appliquer la même technique à des contraintes d'égalité.

Pénalisation des contraintes (4)

La technique précédente était une pénalisation extérieure, c'est-à-dire que les solutions pénalisées ne vérifient pas les contraintes (sauf à convergence $\varepsilon \to 0$).

Une approche opposée est de procéder à une pénalisation intérieure. Pour $\varepsilon > 0$, on considère

$$\inf_{v \in \mathbb{R}^N, F(v) < 0} \left(J(v) - \varepsilon \sum_{i=1}^{M} \frac{1}{F_i(v)} \right)$$

où le deuxième terme est appelée une fonction barrière qui empêche $F(v)$ de s'approcher de 0. Les contraintes d'inégalité sont toujours inactives ! **C'est un algorithme faisable.**
Cela nécessite de connaitre une initialisation qui vérifie $F(v) < 0$.

4 - Algorithme du Lagrangien augmenté

Idée: combiner les avantages des méthodes de pénalisation et celles à base de Lagrangien (type Uzawa).

Soit un problème de minimisation avec contraintes d'égalité

$$\inf_{F(v)=0} J(v)$$

où $J : \mathbb{R}^N \to \mathbb{R}$ et $F : \mathbb{R}^N \to \mathbb{R}^M$ sont de classe C^1. Pour $\mu > 0$ c'est équivalent à

$$\inf_{F(v)=0} J(v) + \frac{\mu}{2}\|F(v)\|^2$$

Pour $v \in \mathbb{R}^N$, $\lambda \in \mathbb{R}^M$, $\mu > 0$, on définit le **Lagrangien augmenté**

$$\mathcal{L}_{aug}(v, \lambda, \mu) = J(v) + \lambda \cdot F(v) + \frac{\mu}{2}\|F(v)\|^2$$

où λ est un multiplicateur de Lagrange.

Remarque. $\mu \neq 1/\varepsilon$ car μ ne sera pas très grand en pratique !

Algorithme du Lagrangien augmenté (2)

Principe de l'algorithme:

- Pour μ fixé, on calcule un point selle de $\mathcal{L}_{aug}(v, \lambda, \mu)$.
- On utilise une nouvelle formule de mise à jour du multiplicateur de Lagrange λ.

Soit v^n le point de minimum de $v \to \mathcal{L}_{aug}(v, \lambda^n, \mu)$. On a

$$J'(v^n) + \lambda^n \cdot F'(v^n) + \mu F(v^n) \cdot F'(v^n) = 0.$$

Si v^n était la **vraie** solution, elle vérifierait la condition d'optimalité

$$J'(v^n) + \lambda^* \cdot F'(v^n) = 0.$$

Par comparaison, si $\mathrm{rg} F'(v^n) = M$, on obtient

$$\lambda^* = \lambda^n + \mu F(v^n).$$

Algorithme du Lagrangien augmenté (3)

Algorithme du Lagrangien augmenté: on choisit $\mu > 0$, $\lambda_0 \in \mathbb{R}^M$ et pour $n \geq 0$ e

$$\mathcal{L}_{aug}(v^n, \lambda^n, \mu) = \min_{v \in \mathbb{R}^N} \mathcal{L}_{aug}(v, \lambda^n, \mu),$$

$$\lambda^{n+1} = \lambda^n + \mu F(v^n).$$

De temps en temps, on augmente μ mais il n'est pas nécessaire d'avoir μ très grand (voir polycopié).

Remarques:

- ça ressemble à Uzawa mais la convergence est plus rapide,
- ça ressemble à la pénalisation mais c'est plus robuste.

5 - Méthodes d'approximations successives.

Que faire pour résoudre

$$\inf_{v \in V,\ F(v)=0} J(v)$$

quand J et F n'ont pas de propriétés particulières ?

Idée: remplacer J et F par leurs développements de Taylor d'ordre 1 ou 2.

Initialisation: $v^0 \in V$. Itérations: pour $n \geq 0$

1. Définir des approximations $J^n(v)$ et $F^n(v)$ au voisinage de v^n.
2. Calcul de v^{n+1} comme solution de

$$\inf_{v \in V,\ F^n(v)=0} J^n(v)$$

Remarque. On peut imposer une condition de région de confiance $\|v - v^n\| \leq \epsilon$ pour être certain que l'approximation est correcte.

On linéarise (Taylor d'ordre 1):

$$\inf_{F(v^{n-1})+F'(v^{n-1})\cdot(v-v^{n-1})=0} \left\{ J(v^{n-1}) + J'(v^{n-1}) \cdot (v - v^{n-1}) \right\},$$

Il existe des algorithmes très efficace pour ce problème linéaire (voir le chapitre suivant).

On note v^n un point de minimum et on itère.

Remarque. Si l'infimum vaut $-\infty$, on rajoute la condition de région de confiance (linéaire) $\|v - v^{n-1}\|_\infty \leq \epsilon$.

Programmation quadratique séquentielle (SQP)

Approximation quadratique pour J, linéaire pour F:

$$\inf_{F^n(v)=0} J^n(v),$$

avec

$$F^n(v) = F(v^{n-1}) + F'(v^{n-1}) \cdot (v - v^{n-1})$$

$$J^n(v) = J(v^{n-1}) + J'(v^{n-1}) \cdot (v - v^{n-1}) + \frac{1}{2} Q^{n-1}(v - v^{n-1}) \cdot (v - v^{n-1})$$

mais Q^{n-1} n'est pas la Hessienne de J mais du Lagrangien

$$Q^{n-1} = J''(v^{n-1}) + \lambda^{n-1} \cdot F''(v^{n-1})$$

avec λ^{n-1} le multiplicateur de Lagrange pour l'itération $n-1$.

En effet, il faut rester sur la variété définie par la contrainte !

Programmation quadratique séquentielle (SQP) (2)

Expliquons pourquoi c'est la Hessienne du Lagrangien et pas de J qui apparait. On a

$$(1)\quad J(v) \approx J(v^*) + J'(v^*)\cdot(v-v^*) + \frac{1}{2}J''(v^*)(v-v^*)\cdot(v-v^*)$$

$$(2)\quad F(v) \approx F(v^*) + F'(v^*)\cdot(v-v^*) + \frac{1}{2}F''(v^*)(v-v^*)\cdot(v-v^*)$$

On multiplie (2) par le multiplicateur de Lagrange λ^* et, comme $F(v)=0$, on somme le résultat à (1), ce qui donne l'approximation (SQP), à une constante près, en utilisant la contrainte linéaire.

Remarque. La condition d'optimalité d'ordre 2 affirme que $J''(v^*) + \lambda^* \cdot F''(v^*) \geq 0$ (et pas que $J''(v^*) \geq 0$!).

Remarques sur les approximations successives

- Algorithmes très efficaces pour SLP et SQP (nombreux logiciels: AMPL, IpOpt, Knitro,...).
- Valable pour des contraintes d'égalité et d'inégalité.
- La variable $d = -(v - v^n)$ est une direction de descente. Donc on peut coupler cette approche avec une recherche de pas $\mu > 0$.
  Plutôt que de poser
  $$v^{n+1} = v^n - d^n$$
  on adapte le pas $\mu > 0$ pour garantir la décroissance de J et la satisfaction des contraintes
  $$v^{n+1} = v^n - \mu d^n$$

6 - Quelques algorithmes utilisant la structure du problème

Le but du jeu est de montrer l'importance de la modélisation et de savoir bien utiliser la structure des modèles pour développer des algorithmes d'optimisation efficaces.

Trois exemples (parmi d'autres):

1. Fonctions composées et rétro-propagation du gradient.
2. Optimisation sous contrainte de modèle et état adjoint.
3. Décomposition-coordination.

L'idée n'est pas de rentrer dans les détails mais d'expliquer des idées originales.

Algorithme de la rétro-propagation du gradient

On minimise sans contrainte une fonction objectif J, composée de m fonctions différentiables,

$$\inf_{v \in \mathbb{R}^N} J(v) = J_m \circ J_{m-1} \circ \cdots \circ J_1(v),$$

avec $J_i(v_i) : \mathbb{R}^{N_i} \mapsto \mathbb{R}^{N_{i+1}}$, $1 \leq i \leq m$, et $N_1 = N$ et $N_{m+1} = 1$.

Motivation: deux applications importantes.

1. Réseaux de neurones profonds en intelligence artificielle.
2. Différentiation automatique de programmes informatiques.

Algorithme de la rétro-propagation du gradient

Lemme (dérivation composée). On note $v_i = J_{i-1} \circ \cdots \circ J_1(v)$ pour $2 \leq i \leq m$ et $v_1 = v$. La fonction $J(v) = J_m \circ J_{m-1} \circ \cdots \circ J_1(v)$ est différentiable au sens de Fréchet en tout $v \in \mathbb{R}^N$ et vérifie pour tout $h \in \mathbb{R}^N$

$$J(v+h) = J(v) + J'_m(v_m) \cdot \left(J'_{m-1}(v_{m-1}) \cdots J'_1(v_1)h\right) + o(h),$$

où $J'_i(v_i)$ est une matrice $N_{i+1} \times N_i$, pour $1 \leq i \leq m-1$.

Preuve: classique, voir le polycopié.

Formule utile pour calculer la dérivée directionnelle $J'(v) \cdot h$ (dans le même ordre que l'on calcule $J(v)$).
Mais comment calculer le gradient complet $J'(v)$ autrement qu'en faisant N calculs avec $h = e_i$, $1 \leq i \leq N$?

Remarque (cruciale): $J'_m(v_m)$ est un vecteur (ligne) car l'espace d'arrivée de J_m est $\mathbb{R}$.

Algorithme de la rétro-propagation du gradient

On se propose de calculer autrement le gradient $J'(v)$. Pour cela, on transpose les matrices gradients en passant de (1) à (2)

$$(1) \qquad J'(v)\cdot h = J'_m(v_m)\cdot\Big(J'_{m-1}(v_{m-1})\cdots J'_1(v_1)h\Big)$$

$$(2) \qquad J'(v)\cdot h = \Big(J'_1(v_1)^* J'_2(v_3)^* \cdots J'_{m-1}(v_{m-1})^* J'_m(v_m)\Big)\cdot h$$

Autrement dit, on vient de trouver une formule explicite

$$J'(v) = J'_1(v_1)^* J'_2(v_3)^* \cdots J'_{m-1}(v_{m-1})^* J'_m(v_m),$$

où on a juste interprété $J'_m(v_m)$ comme un vecteur colonne.

L'algorithme de rétro-propagation du gradient est donc l'algorithme du gradient à pas fixe avec cette formule du gradient.
Rétro-propagation: on part du dernier gradient que l'on multiplie par les gradients transposés dans l'ordre décroissant des indices.

Optimisation sous contrainte de modèle et état adjoint

Soit une fonction objectif $J(v, y)$, C^1 de $\mathbb{R}^N \times \mathbb{R}^M$ dans $\mathbb{R}$, et une contrainte de modèle $Ay = b(v)$ où A est une matrice inversible $M \times M$ et $b(v)$ est une fonction C^1 de $\mathbb{R}^N$ dans $\mathbb{R}^M$. On considère

$$\inf_{v \in \mathbb{R}^N,\, y \in \mathbb{R}^M \text{ tel que } Ay - b(v) = 0} J(v, y).$$

Motivation: $v =$ variables d'optimisation, $y =$ état donné par le modèle (EDO, EDP, voir plus tard la théorie du contrôle).
On peut éliminer $y(v) = A^{-1} b(v)$

$$\tilde{J}(v) = J\left(v, A^{-1} b(v)\right)$$

Lemme. Pour tout $h \in \mathbb{R}^N$, on a

$$\tilde{J}'(v) \cdot h = \frac{\partial J}{\partial v}(v, A^{-1} b(v)) \cdot h + \left\langle \frac{\partial J}{\partial y}(v, A^{-1} b(v)) \,,\, A^{-1} b'(v) \cdot h \right\rangle,$$

où $\langle , \rangle$ désigne le produit scalaire de $\mathbb{R}^M$.

Méthode de l'état adjoint

$$\widetilde{J}'(v) \cdot h = \frac{\partial J}{\partial v}(v, A^{-1}b(v)) \cdot h + \left\langle \frac{\partial J}{\partial y}(v, A^{-1}b(v))\,,\; A^{-1}b'(v) \cdot h \right\rangle$$

Si on veut utiliser cette formule pour calculer $\widetilde{J}'(v)$, il faut calculer $A^{-1}b'(v) \cdot h$ en faisant N résolutions de système linéaire avec $h = e_i$, $1 \leq i \leq N$. Trop cher !

Méthode adjointe. On introduit l'état adjoint $p \in \mathbb{R}^M$ solution de

$$A^* p = \frac{\partial J}{\partial y}(v, A^{-1}b(v)).$$

Alors on obtient $\boxed{\widetilde{J}'(v) = \frac{\partial J}{\partial v}(v, A^{-1}b(v)) + \langle p, b'(v) \rangle}$ car

$$\left\langle \frac{\partial J}{\partial y}(v, y))\,,\; A^{-1}b'(v) \cdot h \right\rangle = \left\langle (A^*)^{-1}\frac{\partial J}{\partial y}(v, y)\,,\; b'(v) \cdot h \right\rangle$$

Un seul système linéaire à résoudre pour p !

Décomposition-coordination

On considère un problème d'optimisation sous contraintes d'égalité

$$(1) \qquad \inf_{v \in \mathbb{R}^N,\ F(v)=F_0} J(v),$$

avec $J : \mathbb{R}^N \mapsto \mathbb{R}$, $F : \mathbb{R}^N \mapsto \mathbb{R}^M$ et $F_0 \in \mathbb{R}^M$ un niveau fixé de contrainte. On suppose que le problème est **décomposable**, au sens où $v = (v_1, ..., v_n)$, avec $v_i \in \mathbb{R}^{n_i}$ et $\sum_{i=1}^n n_i = N$,

$$J(v) = \sum_{i=1}^n J_i(v_i) \quad \text{et} \quad F(v) = \sum_{i=1}^n F_i(v_i).$$

Donc le problème (1) est équivalent à

$$\inf_{v_i \in \mathbb{R}^{n_i}} \sum_{i=1}^n J_i(v_i) \quad \text{sous la contrainte} \quad \sum_{i=1}^N F_i(v_i) = F_0.$$

Algorithme de décomposition par les prix

Pour $p \in \mathbb{R}^M$ et $v \in \mathbb{R}^N$, on introduit le Lagrangien

$$\mathcal{L}(v,p) = J(v) + p \cdot (F(v) - F_0) = \sum_{i=1}^{n} (J_i(v_i) + p \cdot F_i(v_i)) - p \cdot F_0.$$

L'algorithme d'Uzawa, pour une initialisation $p_0 \in \mathbb{R}^M$, construit les suites (u^k) et (p^k), pour $k \geq 0$, et un pas $\mu > 0$,

$$\mathcal{L}(u^k, p^k) = \inf_{v \in \mathbb{R}^N} \mathcal{L}(v, p^k),$$

$$p^{k+1} = p^k + \mu(F(u^k) - F_0).$$

L'hypothèse de décomposition rend la minimisation du Lagrangien très facile car $u^k = (u_1^k, ..., u_n^k)$, avec u_i^k solution de

$$J_i(u_i^k) + p^k \cdot F_i(u_i^k) = \inf_{v_i \in \mathbb{R}^{n_i}} \Big(J_i(v_i) + p^k \cdot F_i(v_i) \Big).$$

Chacun de ces problèmes est de plus petite taille et indépendant des autres.

On l'appelle algorithme de **décomposition par les prix**.

Interprétation de la décomposition par les prix

- Une entreprise doit produire une quantité F_0 de biens.
- Elle a n unités, indicées par i, qui a ses propres variables de production v_i.
- Chaque unité a un coût de production $J_i(v_i)$ et produit une quantité $F_i(v_i)$.

Algorithme de décomposition par les prix: on itère en k

1. La direction de l'entreprise prédit un prix des biens $-p^k$.
2. Chaque unité optimise son gain
$$\inf_{v_i \in \mathbb{R}^{n_i}} \left(J_i(v_i) + p^k \cdot F_i(v_i) \right)$$
qui est le coût de production moins le prix de vente fois la quantité produite.
3. A l'itération suivante l'entreprise ajuste ses prix $-p^{k+1}$ pour produire la quantité requise F_0.