Optimisation sous contraintes
Examen

Le partiel est prévu pour une durée de 2h30.

La rigueur de la rédaction est un élément d'évaluation important.

Un problème sous contraintes

Pour $c \in \mathbb{R}^n$, on considère le problème

$$\inf_{x \in C} \langle c, x \rangle, \tag{P}$$

où $C = \{x \in \mathbb{R}^n / \|x\|_2 \leq 1, x \succeq 0\}$.

Notation : pour $z \in \mathbb{R}^n$, on écrira $z^- \in \mathbb{R}^n$ défini pour $i \in \{1, \dots n\}$ par $(z^-)_i = \begin{cases} z_i & \text{si } z_i < 0, \\ 0 & \text{sinon} \end{cases}$.

1. Montrer que le problème (P) est bien posé, et en donner un minimiseur global évident si $c \geq 0$.

   L'ensemble des contraintes $C := \{x \in \mathbb{R}^n,\ x \geq 0, \|x\|_2 \leq 1\}$ est manifestement borné puisqu'il est inclus dans la boule unité fermée. Il est aussi fermé dans $\mathbb{R}^n$ par continuité des fonctions $x \mapsto x_i$, $i \in \{1, \dots, n\}$ et $x \mapsto \|x\|_2$. Il est donc compact, et comme la fonction objectif est continue, le problème est bien posé.

   Dans le cas de figure où $c \geq 0$, on a $\langle c, x \rangle \geq 0$ pour tout $x \in C$. Or en $x = 0 \in C$, la fonction objectif prend la valeur 0 : c'est donc que 0 est un minimiseur global du problème.

On suppose **dans toute la suite** qu'il existe $i_0 \in \{1, \dots n\}$ tel que $c_{i_0} < 0$.

2. Écrire l'ensemble C grâce à $n+1$ contraintes d'inégalité (les plus simples possibles) $\mathcal{C}^1$ sur $\mathbb{R}^n$, notées $f_0, f_1, \dots, f_n$, où f_0 est associée à la contrainte $\|x\|_2 \leq 1$.

   On remarque que

   $$C = \{x \in \mathbb{R}^n / \forall i \in \{0, \dots, n\}, f_i(x) \leq 0\},$$

   où $f_0 : x \mapsto \|x\|_2^2 - 1$ et pour tout $i \in \{1, \dots, n\}$, $f_i : x \mapsto -x_i$. Ces fonctions sont bien $\mathcal{C}^1$ sur $\mathbb{R}^n$.

3. Montrer que tous les points de C sont réguliers par rapport aux contraintes le défissant. *On pourra différencier les cas en fonctions de si $x \in C$ satisfait $\|x\|_2 < 1$ ou $\|x\|_2 = 1$.*

   Pour $x \in \mathbb{R}^n$, on a $\nabla f_0(x) = 2x$, $\nabla f_i(x) = -e_i$, $i \in \{1, \dots n\}$, où $(e_i)_{1 \leq i \leq n}$ est la base canonique de $\mathbb{R}^n$.

   Soit maintenant $x \in C$. Il faut montrer par définition l'indépendance linéaire des $\nabla f_i(x)$ pour $i \in I(x)$.

   Si $\|x\|_2 < 1$, la contrainte f_0 n'est pas active, et une partie (potentiellement vide) des n autres l'est. Comme la famille $(-e_1, \dots, -e_n)$ est libre, la sous famille $(\nabla f_i(x))_{i \in I(x)}$ de $(-e_1, \dots, -e_n)$ (potentiellement vide) est donc libre également.

   Si la contrainte f_0 est active, i.e. $\|x\|_2 = 1$, alors $x \neq 0$. Ainsi $J(x) := \{i \in \{1, \dots n\}, x_i = 0\}$ est un sous-ensemble strict de $\{1, \dots n\}$. Si la famille d'intérêt était liée, il existerait $\lambda_0 \in \mathbb{R}$, $(\lambda_j)_{j \in J(x)}$ non tous nuls tels que $\lambda_0 x = \sum_{j \in J(x)} \lambda_j e_j$. Nécessairement $\lambda_0 \neq 0$ par liberté de la base canonique. Mais alors pour tout $j \in J(x)$, on a $x_j = 0$ (contraintes actives), on a forcément $\lambda_j = 0$. D'où $\lambda_0 x = 0_{\mathbb{R}^n}$ i.e. $x = 0$. Contradiction car $x \neq 0$. Ainsi la famille d'intérêt est libre.

4. Soit x un minimiseur global du problème.

   (a) Déduire de la précédente question qu'il existe $\lambda_0 \geq 0$, $\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{R}^n_+$ tels que

   $$c + 2\lambda_0 x - \lambda = 0_{\mathbb{R}^n}$$

avec $\lambda_0 = 0$ si $\|x\|_2 < 1$, et pour tout $i \in \{1,\dots,n\}$, $\lambda_i = 0$ si $x_i > 0$.
Notons f la fonction objectif du problème d'optimisation (P), qui est une fonction $\mathcal{C}^1$ (même $\mathcal{C}^\infty$) sur $\mathbb{R}^n$.
Comme x est une solution du problème (P), on a $x \in C$ par définition. D'après la précédente question, on sait donc que $x \in C$ est point régulier pour les contraintes définissant C. Donc les contraintes sont qualifiées en x i.e. $T_C(x) = T_C^\ell(x)$, et donc $T_C(x)^* = T_C^\ell(x)^*$. Or comme x est un minimum global de f sur C, on a $\nabla f(x) \in T_C(x)^*$, d'où $\nabla f(x) \in T_C^\ell(x)^*$. Mais on sait que

$$T_C^\ell(x)^* = \{-\sum_{i\in I(x)} \lambda_i \nabla f_i(x) / \forall i \in I(x), \lambda_i \geq 0.\}$$

Ainsi il existe $(\lambda_i)_{i\in I(x)}$ une famille de nombre réels positifs telle que

$$c = -\sum_{i\in I(x)} \lambda_i \nabla f_i(x),$$

puisque $\nabla f(x) = c$. Complétons la famille $(\lambda_i)_{i\in I(x)}$ en une famille $(\lambda_i)_{0\leq i\leq n}$ de $(\mathbb{R}_+)^{n+1}$ en imposant $\lambda_i = 0$ si $i \notin I(x)$, c'est-à-dire $\lambda_0 = 0$ si $\|x\| < 1$ et pour tout $i \in \{1,\dots,n\}$, $\lambda_i = 0$ si $-x_i < 0$. Ainsi on a

$$c + \sum_{i=0}^{n} \lambda_i \nabla f_i(x) = 0_{\mathbb{R}^n},$$

Comme $\nabla f_0(x) = 2x$ et $\nabla f_i(x) = -e_i$ pour $i \in \{1,\dots,n\}$, on a bien

$$c + 2\lambda_0 x - \underbrace{\sum_{i=1}^{n} \lambda_i e_i}_{\lambda=(\lambda_1,\dots,\lambda_n)} = 0_{\mathbb{R}^n}. \tag{0.1}$$

(b) Justifier que $\lambda_0 > 0$.
Si on avait $\lambda_0 = 0$, on trouverait $c = \lambda \succeq 0$, contredisant l'hypothèse faite sur c. Donc $\lambda_0 > 0$.

(c) Démontrer que

$$x = -\frac{c^-}{\|c^-\|_2}.$$

Comme $x \succeq 0$, on doit avoir $\lambda \succeq c$. On distingue désormais selon le signe de c_i pour $i \in \{1,\dots,n\}$ en exploitant la relation (0.1) pour l'indice i.
Si $c_i < 0$, on a $x_i > \frac{\lambda_i}{2\lambda_0} \geq 0$ donc $i \notin I(x)$. Ainsi par construction, $\lambda_i = 0$ et donc $x_i = -\frac{c_i}{2\lambda_0}$.
Si $c_i = 0$, alors $x_i = \frac{\lambda_i}{2\lambda_0}$. Si $x_i > 0$, alors par construction $\lambda_i = 0$ et donc $x_i = 0$, contradiction. D'où $x_i = 0$ et $\lambda_i = 0$.
Enfin, si $c_i > 0$, comme $\lambda_i \geq c_i > 0$, on a forcément $x_i = 0$.
Finalement, $x = -\frac{c^-}{2\lambda_0}$. Par ailleurs, comme $\lambda_0 > 0$ impose $\|x\|_2 = 1$, et donc $2\lambda_0 = \|c\|_2$, ce qui achève de montrer la formule demandée.

Algorithme de Frank-Wolfe

On se donne C un ensemble non vide, convexe et fermé borné de $\mathbb{R}^n$. f étant une fonction de classe C^1 sur $\mathbb{R}^n$, convexe sur C et de gradient L-Lipschitz sur C. On considère le problème (bien posé)

$$\begin{array}{ll} \text{min.} & f(x) \\ \text{s.c.} & x \in C, \end{array}$$

dont on note $p^\star$ l'optimum. Pour le résoudre, on étudie l'*algorithme de Frank-Wolfe*. Cet algorithme initialisé en $x_0 \in C$ quelconque est défini pour tout $k \in \mathbb{N}$ par

$$\begin{cases} s_k \in \arg\min_{s \in C} \langle \nabla f(x_k), s \rangle, \\ x_{k+1} = (1-\lambda_k)x_k + \lambda_k s_k, \end{cases}$$

où, pour $k \in \mathbb{N}$, $\lambda_k = \frac{2}{k+2}$.

1. (a) Sous les hypothèses satisfaites par f, montrer que l'algorithme est bien défini.
   Il suffit de montrer que le problème

   $$\begin{array}{ll} \text{min.} & \langle u, s \rangle \\ \text{s.c.} & s \in C, \end{array}$$

   est bien posé pour $u \in \mathbb{R}^n$ fixé, ce qui est le cas car la fonction objectif $s \mapsto \langle u, s \rangle$ est continue et l'ensemble des contraintes compact.

   (b) Que présuppose cet algorithme pour pouvoir être appliqué en pratique ?
   L'algorithme suppose qu'on sait déterminer (explicitement ou par un algorithme efficace) un minimiseur du problème paramétré par $u \in \mathbb{R}^n$

   $$\begin{array}{ll} \text{min.} & \langle u, s \rangle \\ \text{s.c.} & s \in C. \end{array}$$

   S'il y en a plusieurs, il faut s'être donné une règle permettant de faire un choix.

   (c) À l'aide des résultats de l'exercice 1, expliciter la partie conduisant au calcul de s_k sous forme de pseudo-code dans le cas où l'ensemble C est donné par les contraintes $x \geq 0$, $\|x\|_2 \leq 1$.
   Dans le cas des contraintes du premier exercice, on a vu qu'on disposait de formules explicites pour le minimiseur, données par $s = 0$ si $u \geq 0$, et par $s = -\frac{u^-}{\|u^-\|_2}$ sinon.

Algorithme 1 Frank-Wolfe - Calcul de s_k

if $\nabla f(x_k) \geq 0$ **then**
  $s_k = 0$
else
  $s_k = -\frac{\nabla f(x_k)^-}{\|\nabla f(x_k)^-\|_2}$
end if
return s_k

On s'intéresse désormais à la convergence de l'algorithme.

2. Dans cette question uniquement, on se place dans le cas particulier où $n = 1$,

$$f : x \mapsto \tfrac{1}{2}x^2, \quad C = [-1, 1], \quad x_0 = 1.$$

(a) Montrer que

$$\forall j \in \mathbb{N}, \qquad s_{2j} = -1,\ s_{2j+1} = 1, \quad x_{2j} = \frac{1}{2j+1},\ x_{2j+1} = -\frac{1}{2j+1}.$$

Notons dans un premier temps que $\nabla f(x) = x$ pour tout $x \in \mathbb{R}$. En outre, si $\nabla f(x) > 0$, $s = \arg\min_{s\in C} \langle \nabla f(x), s\rangle = \{-1\}$, alors que si $\nabla f(x) < 0$, $s = \arg\min_{s\in C} \langle \nabla f(x), s\rangle = \{1\}$. On procède bien entendu par récurrence. On suppose la propriété vraie au rang k, et on la démontre au rang $k+1$. Si k est pair, on a donc $s_k = -1$, $x_k = \frac{1}{k+1}$. On calcule alors

$$x_{k+1} = \frac{1}{k+2}\left(k\frac{1}{k+1} - 2\right) = \frac{1}{k+2}\frac{-k-2}{k+1} = -\frac{1}{k+1}.$$

En outre, comme $\nabla f(x_{k+1}) = x_{k+1} < 0$, $s_{k+1} = 1$. Si maintenant k est impair, on a $s_k = 1$, $x_k = -x_{k-1} = -\frac{1}{k}$, et donc

$$x_{k+1} = \frac{1}{k+2}\left(-k\frac{1}{k} + 2\right) = \frac{1}{k+2}.$$

En outre, comme $\nabla f(x_{k+1}) = x_{k+1} > 0$, $s_{k+1} = -1$.

(b) Montrer alors que la suite $(f(x_k) - p^\star)$ converge vers 0 en donnant la vitesse de convergence.

Ici, $p^\star = 0$. D'après ce qui précède,

$$f(x_k) = \frac{1}{2}\frac{1}{(k+1)^2} \text{ si } k \text{ est pair}, \quad f(x_k) = f(-x_{k-1}) = \frac{1}{2}\frac{1}{k^2} \text{ si } k \text{ est impair},$$

ce qui montre que la suite $(f(x_k) - p^\star)$ tend vers 0 comme $\frac{1}{2k^2}$.

3. Définissant le diamètre de C via

$$d_C := \sup_{(x,y)\in C^2} \|x - y\|,$$

on veut montrer que

$$\forall k \in \mathbb{N}^*, \quad f(x_k) - p^\star \leq 2L\frac{d_C^2}{k+2}.$$

On donne le lemme suivant valable pour une suite positive (α_k), et $M > 0$:

$$\forall k \in \mathbb{N},\ \alpha_{k+1} \leq (1 - \lambda_k)\alpha_k + M\lambda_k^2 \quad \Longrightarrow \quad \forall k \in \mathbb{N}^*,\ \alpha_k \leq \frac{4M}{k+2}.$$

(a) Montrer que

$$\forall x, y \in C, \quad f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|x - y\|^2.$$

f étant de classe C^1, et utilisant l'inégalité de Cauchy-Schwarz ainsi que le caractère lispchitzien de ∇f

$$\begin{aligned} f(y) - f(x) - \langle \nabla f(x), y - x \rangle &= \int_0^1 \langle \nabla f(x + t(y-x)) - \nabla f(x), y - x \rangle \, dt \\ &\leq \int_0^1 \|\nabla f(x + t(y-x)) - \nabla f(x)\| \|x - y\| \\ &\leq \int_0^1 t \, dt \|x - y\|^2 = \frac{1}{2} L \|x - y\|^2. \end{aligned}$$

(b) En déduire que

$$\forall k \in \mathbb{N}, \quad f(x_{k+1}) \leq f(x_k) + \lambda_k \langle \nabla f(x_k), x^\star - x_k \rangle + \frac{L}{2} d_C^2 \lambda_k^2,$$

où $x^\star$ est un minimiseur du problème.

On écrit utilisant successivement la définition de x_{k+1}, le caractère minimisant de s_k et enfin le fait que C est borné :

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &= f(x_k) + \lambda_k \langle \nabla f(x_k), s_k - x_k \rangle + \frac{L}{2} \lambda_k^2 \|s_k - x_k\|^2 \\ &\leq f(x_k) + \lambda_k \langle \nabla f(x_k), x^\star - x_k \rangle + \frac{L}{2} \lambda_k^2 \|s_k - x_k\|^2 \\ &\leq f(x_k) + \lambda_k \langle \nabla f(x_k), x^\star - x_k \rangle + \frac{L}{2} d_C^2 \lambda_k^2 \end{aligned}$$

(c) Conclure à l'aide de la suite définie pour $k \in \mathbb{N}$ par $\alpha_k = f(x_k) - p^\star$.

Avec la notation introduite et d'après la question précédente,

$$\alpha_{k+1} \leq \alpha_k + \lambda_k \langle \nabla f(x_k), x^\star - x_k \rangle + \frac{L}{2} d_C^2 \lambda_k^2$$

Par convexité de f, $f(x_k) - p^\star = f(x_k) - f(x^\star) \geq \langle \nabla f(x_k), x_k - x^\star \rangle$, ce qui donne finalement

$$\alpha_{k+1} \leq (1 - \lambda_k)\alpha_k + + \frac{L}{2} d_C^2 \lambda_k^2$$

L'application du lemme avec $M = \frac{1}{2} L d_C^2$ clôt la preuve.

4. À la lumière de l'ensemble des questions précédentes, quels avantages et inconvénients voyez-vous à cet algorithme par rapport à l'algorithme du gradient projeté ?

L'algorithme du gradient projeté suppose qu'on sait calculer (explicitement ou efficacement) la projection, alors que l'algorithme de Frank-Wolfe repose sur la résolution (explicite ou rapide) du problème d'optimisation sous-jacent avec fonction objectif linéaire. Selon les situations, on préférera donc l'un ou l'autre des algorithmes.[1]

1. En fait, il est beaucoup plus courant de savoir résoudre efficacement un problème linéaire comme celui qui apparaît dans l'algorithme de Frank-Wolfe, qu'un problème quadratique qu'est le calcul d'une projection.

Contrairement à l'algorithme de gradient projeté, l'algorithme de Frank-Wolfe requiert que C soit borné.

Enfin, on sait que le gradient projeté converge linéairement lorsque la fonction est fortement convexe ; l'exemple donné (avec objectif fortement convexe et de gradient 1-Lipschitz) en dimension 1 montre que la convergence en $\frac{1}{k}$ obtenue théoriquement ne peut être améliorée de manière systématique pour les fonctions fortement convexes.

Remarque culturelle : l'algorithme de Frank-Wolfe est aussi utilisé de manière préférentielle lorsque l'on cherche (pour des raisons liées à l'application visée) spécifiquement des points *extrémaux* de l'ensemble des contraintes, c'est-à-dire les points qui ne sont le milieu d'aucun segment inclus dans C. Il s'agit donc de points de la frontière de C qui, de manière informelle, en sont des "sommets".