Ensimag 2A – automne 2013
Examen du vendredi 20 décembre
Florent Cadoux, Jérôme Malick, Sofia Zaourar

Optimisation numérique - Examen déc. 2013

- Tout document papier est autorisé.
- Barème approximatif : 5 – 15.
- La qualité de la présentation et de l'argumentation sera un élément important de la notation.
- L'énoncé fait quatre pages. Les deux exercices sont indépendants.

Exercice 1 – Résolution explicite avec KKT. Soit le problème d'optimisation dans $\mathbb{R}^2$

$$(\mathrm{P}) \left\{ \begin{array}{l} \min \quad xy^2 + 1/x \\ \quad (x,y) \in C \end{array} \right.$$

où C est le sous-ensemble de $\mathbb{R}^2$ défini par

$$C = \left\{ (x,y) \in \mathbb{R}^2 : \ x \geqslant 0, \ y \geqslant 0, \ xy \geqslant 1, \ x+y \leqslant 3 \right\}.$$

a) Dessiner C. Justifier que C est convexe et compact.

b) Montrer qu'il existe une solution à (P).

c) Justifier que les contraintes de positivité de x et y sont inactives en tout point réalisable.

d) Écrire les conditions nécessaires d'optimalité.

e) En raisonnant par l'absurde sur les équations, montrer que les deux contraintes doivent être actives à une solution de (P).

f) Résoudre le problème (P).

Exercice 2 – Optimisation des performances d'un réseau. Cet exercice porte sur un problème classique d'optimisation dans les réseaux de télécom, appelé "multicommodity flow". On souhaite minimiser le temps de transfert d'information dans le réseau ; après modélisation, on obtient le problème d'optimisation

$$\left\{ \begin{array}{l} \min \quad \displaystyle\sum_{j=1}^{n} f_{c_j}\left(\sum_{k=1}^{K} x_j^k \right) \\ \quad x^k \in P_k, \quad \text{pour } k = 1, \ldots, K \end{array} \right.$$

dont les données sont les suivantes :

- Variables : $x^1, \ldots, x^K$ sont K vecteurs de $\mathbb{R}^n$ (et on note donc $x_j^k \in \mathbb{R}$ la j-ème composante du vecteur $x^k \in \mathbb{R}^n$) ;
- Contraintes : chaque x^k vérifie un certain nombre de contraintes affines qui définissent un polyèdre (convexe) P_k de $\mathbb{R}^n$ qui s'avère être inclus dans $\mathbb{R}_+^n$ (tous les points ont toutes leurs composantes positives). On suppose aussi que les P_k sont compacts.
- Objectif : à chaque composante j, est associé un coût défini, pour $c_j > 0$ fixé, par la fonction convexe $f_{c_j} : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$

$$f_{c_j}(t) = \left\{ \begin{array}{ll} \dfrac{t}{c_j - t} & \text{si } 0 \leqslant t < c_j \\ +\infty & \text{sinon.} \end{array} \right.$$

Ce problème se prête à une résolution par dualité, c'est ce que nous allons étudier dans cet exercice. Considérons la variable supplémentaire $y \in \mathbb{R}^n$ pour formuler le problème comme

$$\left\{\begin{array}{l} \min \quad \displaystyle\sum_{j=1}^{n} f_{c_j}(y_j) \\ \quad \sum_{k=1}^{K} x^k - y = 0 \\ \quad y \in \mathbb{R}^n, x^k \in P_k, \quad \text{pour } k = 1, \ldots, K. \end{array}\right. \tag{1}$$

a) Les "fonctions conjuguées" des f_{c_j} vont apparaitre dans le problème dual. Donnons la définition de la conjuguée pour une fonction quelconque d'une variable réelle $f\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$. On rappelle que le domaine de f est le sous-ensemble (de $\mathbb{R}$)

$$\operatorname{dom} f = \{t \in \mathbb{R} : \ f(t) < +\infty\}.$$

On définit f^*, la fonction conjuguée de f, par

$$\forall \lambda \in \mathbb{R}, \qquad f^*(\lambda) = \sup_{t \in \operatorname{dom} f} \{t\lambda - f(t)\}.$$

Montrer que $f^*\colon \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ est une fonction convexe.

b) Soit $c_j > 0$; montrer que la valeur de la fonction conjuguée de f_{c_j} en $\lambda_j \in \mathbb{R}$ s'exprime comme $f^*_{c_j}(\lambda_j) = \max\{0, -m(c_j, \lambda_j)\}$ où $m(c_j, \lambda_j)$ est la valeur optimale du problème convexe

$$\left\{\begin{array}{l} \inf \quad -\lambda t + t/(c_j - t) \\ \quad 0 < t < c_j. \end{array}\right.$$

c) Résoudre le problème ci-dessus pour $\lambda_j \in \mathbb{R}$ fixé, en distinguant deux cas : $\lambda_j > 1/c_j$ et $\lambda_j \leqslant 1/c_j$. En déduire que

$$f^*_{c_j}(\lambda_j) = \left\{\begin{array}{cl} (\sqrt{c_j\lambda_j} - 1)^2 & \text{si } \lambda_j \geqslant 1/c_j \\ 0 & \text{sinon.} \end{array}\right.$$

Revenons à présent au problème (1).

d) On dualise la contrainte couplant les x^k et y entre eux : écrire le problème sous la forme du cours, introduire la variable duale $\lambda \in \mathbb{R}^n$ et le lagrangien associé.

e) Montrer que la fonction duale s'écrit

$$\theta(\lambda) = \sum_{j=1}^{n} f^*_{c_j}(\lambda_j) + \sum_{k=1}^{K} \sigma_{P_k}(-\lambda)$$

où $f^*_{c_j}$ est la fonction conjuguée de f_{c_j} et σ_{P_k} est fonction support de l'ensemble P_k.

f) En utilisant que les P_k sont inclus dans $\mathbb{R}^n_+$, montrer que : si une composante i de λ est telle que $\lambda_i \leqslant 1/c_i$ alors $\sigma_{P_k}(-\lambda) \geqslant \sigma_{P_k}(-\tilde{\lambda})$, où $\tilde{\lambda}$ est le vecteur de $\mathbb{R}^n$ tel que

$$\tilde{\lambda}_i = 1/c_i \qquad \text{et} \qquad \tilde{\lambda}_j = \lambda_j \text{ pour tout } j \neq i.$$

En déduire que $\theta(\lambda) \geqslant \theta(\tilde{\lambda})$. En déduire finalement que le problème dual garde la même valeur optimale si on ajoute la contrainte $\lambda \in \Lambda$ avec

$$\Lambda = \{\lambda \in \mathbb{R}^n : \ \lambda_j \geqslant 1/c_j \text{ pour tout } j = 1, \ldots, n\}.$$

g) Montrer que le problème dual peut ainsi se formuler

$$\text{(D)} \quad \inf_{\lambda \in \Lambda} \quad \Phi(\lambda) + \Psi(\lambda)$$

avec deux fonctions Φ et Ψ convexes et Φ de classe C^∞ (sur l'intérieur de Λ).

h) On suppose qu'il existe une solution à (D). Montrer qu'il n'y a pas de saut dual.

La suite de l'exercice porte sur la résolution numérique du problème dual.

i) Soit $\lambda \in \mathbb{R}^n$ et $k \in \{1, \dots, K\}$. Montrer qu'il existe $x^k(\lambda) \in P_k$ tel que $\sigma_{P_k}(\lambda) = \lambda^\top x^k(\lambda)$. Montrer aussi que $x^k(\lambda) \in \partial\sigma_{P_k}(\lambda)$.

j) Soient $\varphi_1 \colon \mathbb{R}^n \to \mathbb{R}$ et $\varphi_2 \colon \mathbb{R}^n \to \mathbb{R}$ deux fonctions convexes. Montrer que si $g_1 \in \partial\varphi_1(x)$ et $g_2 \in \partial\varphi_2(x)$, alors $g_1 + g_2 \in \partial(\varphi_1 + \varphi_2)(x)$.

k) Soient $\varphi \colon \mathbb{R}^n \to \mathbb{R}$ une fonction convexe. Montrer que si g est un sous-gradient de φ en $-x$, alors $-g$ est un sous-gradient en x de la fonction $y \mapsto \varphi(-y)$. En déduire que $-\sum_k x^k(-\lambda) \in \partial\Psi(\lambda)$.

l) On suppose que l'on dispose de procédures qui calculent les $x^k(\lambda)$. Écrire précisement ce que fait un simulateur de Ψ. En déduire ce que fait un simulateur de θ.

m) Expliquer brièvement ce qu'est la méthode des faisceaux pour résoudre le problème (D). Se placer à l'itération T : noter $\lambda^1, \dots, \lambda^t, \dots, \lambda^T$ les itérés précédents, et écrire précisement le problème quadratique qui calcule λ^{T+1}.

n) Comme en TP, on peut résoudre le problème quadratique de l'itération T en le mettant sous format standard et appeler le solver quadratique de scilab (`qld`). L'aide scilab sur cette fonction `qld` est fournie en dernière page du sujet. Supposons avoir les entrées Q_T, p_T, C_T, b_T, $(c_i)_T$, $(c_s)_T$ et me de l'appel de `qld` pour le problème à l'itération T. Quelles sont les entrées qui sont modifiées au problème quadratique de l'itération suivante $(T+1)$?

o) Écrire explicitement la mise à jour des entrées de `qld` pour formuler le problème quadratique de l'itération suivante $(T+1)$.

On dispose donc d'une approche par dualité pour résoudre le problème initial. On va maintenant améliorer cette approche en construisant une méthode (de type faisceaux) adaptée à la structure particulière du problème dual (donc plus efficace en pratique).

p) Écrire le modèle des plans sécants de σ_{P_k} à l'itération T. En déduire un modèle de Ψ (exploitant le fait que c'est une somme).

q) Écrire le modèle quadratique de Newton de Φ en λ^T.

r) Proposer un modèle pour la fonction duale $\Phi + \Psi$ avec lequel on construit une méthode de faisceaux adaptée. Écrire le problème quadratique qui calcule l'itéré suivant λ^{T+1}.

Pour information : cet exercice porte sur un problème classique d'optimisation dans les réseaux de télécom, appelé "multicommodity flow". Voici une intuition d'où vient ce problème :

- Il y a K paquets à faire passer dans un réseau, représenté par un graphe orienté à n arcs. On veut minimiser le temps de trajet total des K paquets à travers ce réseau.
- À un arc j, on associe sa capacité maximale $c_j > 0$, et une fonction convexe croissante f_{c_j} qui mesure sa "congestion" et explose donc en c_j (la fonction de congestion choisie dans cet exercice s'appelle la fonction de Kleinrock).
- Les contraintes P^k codent le chemin pris par le paquet k d'un point à un autre dans le réseau. Chaque P_k est donc défini par $x^k \geqslant 0$ et des contraintes affines du type $Ax^k = a^k$, avec A la matrice d'incidence du graphe et a^k représentant le flot à transporter (a^k n'a que 2 coordonnées non nulles qui ont des valeurs opposées). Simuler Ψ consiste ainsi à résoudre K problèmes de plus court chemin dans le graphe (cf cours de RO en 1A).

qld - linear quadratic programming solver

Calling Sequence

```
[x,lagr]=qld(Q,p,C,b,ci,cs,me [,tol])
[x,lagr,info]=qld(Q,p,C,b,ci,cs,me [,tol])
```

Parameters

Q: real positive definite symmetric matrix (dimension n x n).
p: real (column) vector (dimension n)
C: real matrix (dimension (me + md) x n)
b: RHS column vector (dimension (me + md))

ci: column vector of lower-bounds (dimension n).
If there are no lower bound constraints, put ci = [].
If some components of x are bounded from below, set the other (unconstrained) values of ci to a very large negative number (e.g. ci(j) = -number_properties('huge').

cs: column vector of upper-bounds. (Same remarks as above).
me: number of equality constraints (i.e. C(1:me,:)*x = b(1:me))
tol:Floatting point number, required précision.

x: optimal solution found.
lagr: vector of Lagrange multipliers.
If lower and upper-bounds ci,cs are provided, lagr has n + me + md components and lagr(1:n) is the Lagrange vector associated with the bound constraints and lagr (n+1 : n + me + md) is the Lagrange vector associated with the linear constraints. (If an upper-bound (resp. lower-bound) constraint i is active lagr(i) is > 0 (resp. <0). If no bounds are provided, lagr has only me + md components.

info: integer, return the execution status instead of sending errors.
info==1 : Too many iterations needed
info==2 : Accuracy insufficient to statisfy convergence criterion
info==5 : Length of working array is too short
info==10: The constraints are inconsistent

Description

Minimize 0.5*x'*Q*x + p'*x

under the constraints

```
 C(j,:) x = b(j),  j=1,...,me
 C(j,:) x <= b(j), j=me+1,...,me+md
 ci <= x <= cs
```