Ensimag 2A – automne 2015
TDs optimisation
Alejandro Blumentals, Franck Iutzeler, Jérôme Malick

TP 4 – algorithmes d'optimisation convexe I

Objectif : Dans ce premier TP sur les méthodes d'optimisation convexe non-différentiable, nous écrivons un simulateur pour une fonction non-différentiable que nous minimisons ensuite par la méthode des sous-gradients.

Exercice 1 – Affectation linéaire. Reprenons le problème d'affectation évoqué lors du premier cours :

$$(\mathrm{P}) \begin{cases} \min \quad \sum_{i=1}^n \sum_{j=1}^n a_{ij} u_{ij} \\ \quad \sum_{j=1}^n u_{ij} = 1, \quad \text{pour tout } i = 1, \dots, n, \\ \quad \sum_{i=1}^n u_{ij} = 1, \quad \text{pour tout } j = 1, \dots, n, \\ \quad u_{ij} \in \{0,1\}, \quad \text{for all } i, j = 1, \dots, n. \end{cases}$$

a) Écrire (P) sous forme matricielle, et donner une interprétation « matricielle » du problème.

b) On veut dualiser les n premières contraintes dans (P) : écrire le problème (P) sous la forme du cours (c'est-à-dire exhiber, φ, c et $\mathcal{U}$).

c) Écrire le lagrangien $L(\cdot;\cdot)$ associé à ce problème. Montrer qu'il existe une matrice U_λ qui maximise $L(\cdot;\lambda)$ à λ fixé. Définir θ la fonction duale associée.

d) Montrer qu'on peut calculer U_λ explicitement. Indice : remarquer que la maximisation de $L(\cdot,\lambda)$ se découple par rapport à chaque colonne u_i de la matrice U.

Exercice 2 – Simulateur de la fonction duale. Pour la suite du TP, on fixe les données : on suppose qu'on est en dimension $n = 10$ et que les coûts sont donnés par la matrice

```
A = [1:n]'*[1:n];
```

Calculer la matrice U_λ (utiliser des options de la commande `max` de Scilab). Utiliser cette matrice U_λ pour écrire un simulateur `simdual` de la fonction duale θ sous la forme des simulateurs des TPs précédents

```
[th,g] = simdual(lam)
```

Les sorties sont ici, pour l'entrée `lam` $= \lambda$
– `th` $= \theta(\lambda) = L(U_\lambda, \lambda)$, la valeur de la fonction duale,
– `g` $= g(\lambda) = -c(U_\lambda)$ (avec les notations du cours).

Remarque : Mettre uniquement λ en paramètre d'entrée du simulateur pour bien garder la forme des TPs précédents. Rigoureusement, la matrice A devrait aussi être en paramètre du simulateur, mais pour faire simple, on la considère comme une variable globale.

Exercice 3 – Illustration en deux dimensions. Pour donner une idée de la forme de θ, on visualisera les courbes de niveaux d'une restriction de θ à un plan - ce qui peut se faire de la manière suivante. En se basant sur la fonction d'affichage des TPs précédents, implémenter une fonction `simdual2d` qui prend en entrée $(\lambda_1, \lambda_2) \in \mathbb{R}^2$, qui ajoute $n-2$ composantes nulles, puis qui appelle `simdual` en

$$\lambda = (\lambda_1, \lambda_2, 0, \dots, 0).$$

Exercice 4 – Méthode de sous-gradient. En s'inspirant du cas différentiable, une première idée pour minimiser la fonction duale est faire faire simplement une méthode de sous-gradient

$$\lambda_{k+1} = \lambda_k - t_k g(\lambda_k).$$

Voir le cours pour les motivations, les avantages et (surtout !) les inconvénients de cette méthode.

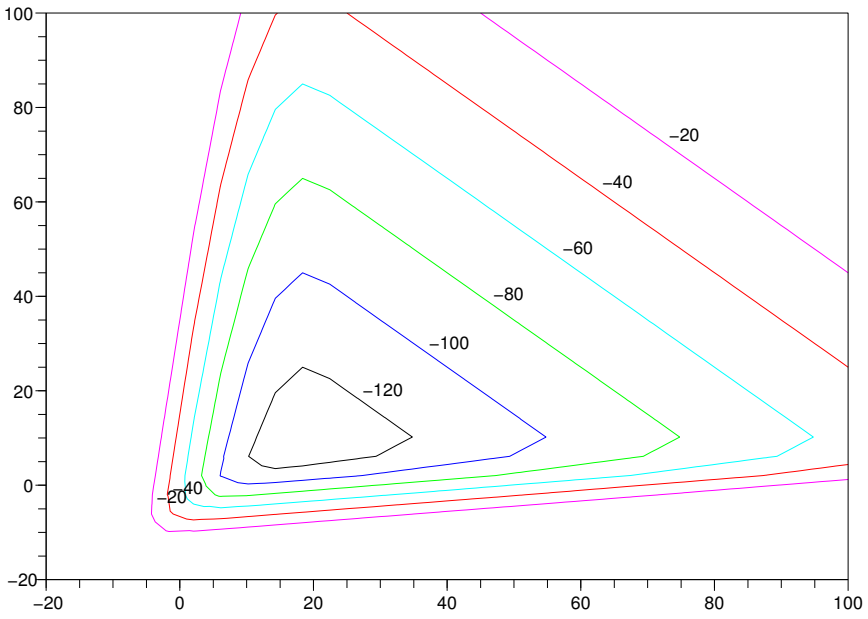

Figure 1 Contours de la fonction duale θ autour du minimum

Modifier l'algorithme du TP1 pour implémenter la méthode de sous-gradient pour maximiser θ avec deux types de stratégie pour le pas t_k :

— $t_k = \frac{1}{k}$;

— $t_k = q^k$, avec $q \in]0, 1[$ donné (essayer plusieurs valeurs - prendre $q \geqslant 0.9$).

Ne pas se préocuper du test d'arrêt et mettre simplement un nombre maximal d'itérations. Que finira-t-il par se passer si on laisse courir cette méthode sans aucun test d'arrêt ?