
Mini-Projet

Pedro M. V. de Carvalho

23 janvier 2015

1 INTRODUCTION

Le problème c'est de minimiser une fonction $f_I(\cdot)$ d'une liste de k objets choisis parmi n , chacun avec un coût bidimensionnel (c_1^i, c_2^i) . La fonction à minimiser est :

$$f_I(y) = \max\{\alpha y_1 + (1 - \alpha)y_2 : \alpha \in [\alpha_{min}, \alpha_{max}]\} \quad (1.1)$$

où $y = (\sum_{i=1}^n c_1^i x_i, \sum_{i=1}^n c_2^i x_i)$, $x = (x_1, \dots, x_n)$ est le vecteur binaire avec $x_i = 1$ si l'objet i est choisi et 0 sinon, $\alpha_{min} < \alpha_{max}$, $\alpha_{min} < 1$ et $0 < \alpha_{max}$.

Pour faire ce projet, j'ai utilisé le langage de programmation Python. J'ai utilisé des listes et des fonctions disponibles dans la librairie Numpy.

2 RÉSULTATS PRÉLIMINAIRES

2.1 QUESTION 1

Tout d'abord, il faut trouver trois vecteurs y , y' et y'' avec les propriétés

$$f_I(y) < f_I(y') \text{ et } f_I(y' + y'') < f_I(y + y'') \quad (2.1)$$

pour montrer que le principe d'optimalité n'est pas vérifié. Prenons $\alpha_{min} = 0.1$, $\alpha_{max} = 1000$, $y = (1, 4)$, $y' = (1, 6)$ et $y'' = (5, 1)$. Nous avons $f_I(y) = 3.7 < 5.5 = f_I(y')$ et $f_I(y' + y'') = 6.9 < 1005 = f_I(y + y'')$. Q.E.D.

2.2 QUESTIONS 2, 3 ET 4

Ces questions ont été répondues par les fonctions "vecs", "naive", et "lexialgo" dans le fichier part2.py.

2.3 QUESTION 5

L'algorithme naïf a complexité $O(n^2)$ pour les comparaisons systématiques. Le tri lexicographique a complexité $O(n \log n)$ (qui est la complexité du quicksort) et le parcours a complexité $O(n)$, et donc l'algorithme de la question 4 a complexité $O(n \log n + n) = O(n \log n)$. L'exécution du programme part2.py montre les résultats d'exécution, et le fichier part2.png montre des courbes générés par Excel par les données dans part2.txt.

3 UNE PREMIÈRE PROCÉDURE DE RÉOLUTION

3.1 QUESTION 6

Pour montrer que le point minimax est forcément dans l'ensemble non-dominé, il suffit de montrer que, pour tout point dominé, il y a toujours un point qui le domine et qui est mieux. Soit $y = (y_1, y_2)$ un point dominé. Pour toute intervalle I , il existe un point $y' = (y'_1, y'_2) \neq y$ qui domine y et avec la propriété $f_I(y') < f_I(y)$. Si y' domine y , $y'_1 \leq y_1$ et $y'_2 \leq y_2$. Alors $f_I(y) = (y_1 - y_2)\alpha + y_2$ et $f_I(y') = (y'_1 - y'_2)\alpha' + y'_2$ pour certains $\alpha, \alpha' \in [\alpha_{min}, \alpha_{max}]$. Comme on va démontrer à la question 8, si $y_1 > y_2$ alors $\alpha = \alpha_{max}$, et $\alpha = \alpha_{min}$ sinon. Choisissons y' de façon que $y_1 - y_2 = y'_1 - y'_2$, d'où $\alpha = \alpha'$. Dans ce cas là :

$$\begin{aligned} y'_2 &< y_2 \\ y'_2 + (y_1 - y_2)\alpha &< y_2 + (y_1 - y_2)\alpha \\ y'_2 + (y_1 - y_2)\alpha' &< y_2 + (y_1 - y_2)\alpha \\ y'_2 + (y'_1 - y'_2)\alpha &< y_2 + (y_1 - y_2)\alpha \\ f_I(y') &< f_I(y) \end{aligned} \tag{3.1}$$

Alors, quelque soit y non-dominé, et pour tout intervalle I , il y a toujours un vecteur y' qui domine y et qui est préféré à y . Q.E.D.

3.2 QUESTION 7

Soit $P(i, j)$ le sous-problème restreint à la sélection de j objets dans $\{1, \dots, i\}$. On a :

$$P(i, j) = \begin{cases} \{(0, 0)\} & \text{si } j = 0 \\ \emptyset & \text{si } j > i \\ y(i) & \text{si } i = j = 1 \\ OPT(P(i-1, j-1) + y(i) \cup P(i-1, j)) & \text{sinon} \end{cases} \tag{3.2}$$

où $y(i)$ est le coût de l'objet i tout seul. Il faut donc calculer $P(n, k)$, soit de façon récursive, soit séquentielle. Pour éviter des stack-overflows, on va l'implémenter séquentiellement. La complexité de cet algorithme est $O(knM)$ où M est une borne supérieure sur la valeur d'un objectif. Cet algorithme est implémenté par la fonction "P" dans le fichier part3.py.

3.3 QUESTION 8

On peut réarranger les éléments de $f_I(y)$:

$$f_I(y) = \max\{(y_1 - y_2)\alpha + y_2 : \alpha \in [\alpha_{min}, \alpha_{max}]\} \quad (3.3)$$

Le terme à être maximisé est linéaire en α , et il est strictement croissant ssi $y_1 > y_2$ et strictement décroissant ssi $y_1 < y_2$, d'où on peut conclure que le terme ne peut être maximisé que pour les bornes de l'intervalle I , c'est-à-dire pour $\alpha = \alpha_{min}$ ou $\alpha = \alpha_{max}$. Il est donc facile à implémenter une fonction avec complexité $O(n)$ qui trouve le vecteur qui minimise $f_I(y)$ pour un intervalle I donné parmi n vecteurs. Cet algorithme est implémenté par les fonctions "f" et "q8" dans le fichier part3.py.

3.4 QUESTION 9

Il suffit d'exécuter le fichier part3.py pour voir les résultats.

4 UNE SECONDE PROCÉDURE DE RÉOLUTION

4.1 QUESTION 10

La règle de I -dominance est :

$$y \text{ } I\text{-domine } y' \text{ si } \begin{cases} \forall \alpha \in I, \alpha y_1 + (1 - \alpha) y_2 \leq \alpha y'_1 + (1 - \alpha) y'_2 \\ \exists \alpha \in I, \alpha y_1 + (1 - \alpha) y_2 < \alpha y'_1 + (1 - \alpha) y'_2 \end{cases} \quad (4.1)$$

Si NI est l'ensemble des points non I -dominés et ND est l'ensemble des points non-dominés au sens de Pareto, pour montrer que $NI \subseteq ND$ il suffit de montrer que $y \in NI \rightarrow y \in ND$. On raisonne par absurde : Supposons que y n'est pas non-dominé. Donc il existe un point $y' \neq y$ tel que $y'_1 \leq y_1$ et $y'_2 \leq y_2$. Mais si c'est le cas, il est forcément vrai que $\forall \alpha > 0, \alpha y'_1 \leq \alpha y_1$ et $\forall \alpha < 1, (1 - \alpha) y'_2 \leq (1 - \alpha) y_2$, et au moins une de ces inégalités est stricte car $y \neq y'$. Comme $\alpha_{min} < 1$ et $\alpha_{max} > 0$, on a $\exists \alpha, (\alpha \in I) \wedge (0 < \alpha < 1) \wedge (\alpha y'_1 + (1 - \alpha) y'_2 < \alpha y_1 + (1 - \alpha) y_2)$, et donc y n'est pas non I -dominé. Q.E.D.

Il faut maintenant donc montrer qu'un point minimax est inclus dans NI . Si y est minimax, alors $\max\{\alpha y_1 + (1 - \alpha) y_2 : \alpha \in I\}$ est minimal. Mais s'il existe un point y' qui I -domine y , alors $\forall \alpha > 0, \alpha y'_1 \leq \alpha y_1$ et $\forall \alpha < 1, (1 - \alpha) y'_2 \leq (1 - \alpha) y_2$, et pour que y soit minimax il faut qu'il existe un tel $\alpha' \in I$ que $\alpha' y'_1 + (1 - \alpha') y'_2 > \alpha y_1 + (1 - \alpha) y_2$. Mais si c'est le cas, alors $\alpha' y_1 + (1 - \alpha') y_2 > \alpha y_1 + (1 - \alpha) y_2$, et α n'est pas l'élément qui maximise $\alpha y_1 + (1 - \alpha) y_2$, contradiction. Donc il n'existe pas y' qui I -domine y si y est un point minimax. Q.E.D.

4.2 QUESTION 11

Il faut montrer que la règle de I -dominance est équivalente à la règle de I' -dominance :

$$y \text{ } I' \text{-domine } y' \text{ si } \begin{cases} \forall \alpha \in \{\alpha_{min}, \alpha_{max}\}, \alpha y_1 + (1 - \alpha) y_2 \leq \alpha y'_1 + (1 - \alpha) y'_2 \\ \exists \alpha \in \{\alpha_{min}, \alpha_{max}\}, \alpha y_1 + (1 - \alpha) y_2 < \alpha y'_1 + (1 - \alpha) y'_2 \end{cases} \quad (4.2)$$

Soit $I(y, y') \leftrightarrow y \text{ } I \text{-domine } y'$, $I'(y, y') \leftrightarrow y \text{ } I' \text{-domine } y'$ et $f_\alpha(y) = \alpha y_1 + (1 - \alpha) y_2$. Ce qu'il faut montrer donc est $I(y, y') \leftrightarrow I'(y, y')$. Supposons sans perte de généralité que $y_1 > y_2$. Alors, comme à la question 8, on peut réarranger les termes : $f(\alpha, y) = (y_1 - y_2)\alpha + y_2$, et on voit que $f(\alpha, y)$ est une fonction strictement croissante en α .

On montre premièrement que $I(y, y') \rightarrow I'(y, y')$. C'est claire que $\forall \alpha \in I, f(\alpha, y) \leq f(\alpha, y') \rightarrow \forall \alpha \in \{\alpha_{min}, \alpha_{max}\}, f(\alpha, y) \leq f(\alpha, y')$. Alors, $f(\alpha, y)$ est une droite en α pour y fixé. Cela veut dire que, si $I(y, y')$, il y a au plus un point α' ou $f(\alpha', y) = f(\alpha', y')$ où les droites se croisent, et ce point doit être soit α_{min} , soit α_{max} (sinon il est impossible que $\forall \alpha \in I, f(\alpha, y) \leq f(\alpha, y')$, car l'inégalité est inversé sur le point α') ; tous les autres points α dans I doivent avoir $f(\alpha, y) < f(\alpha, y')$, et particulièrement l'un des points α_{min} ou α_{max} . On peut conclure que $\exists \alpha \in I, f(\alpha, y) < f(\alpha, y') \rightarrow \exists \alpha \in \{\alpha_{min}, \alpha_{max}\}, f(\alpha, y) < f(\alpha, y')$. Alors $I(y, y') \rightarrow I'(y, y')$.

Montrons maintenant que $I'(y, y') \rightarrow I(y, y')$. C'est aussi immédiate que $\exists \alpha \in \{\alpha_{min}, \alpha_{max}\}, f(\alpha, y) < f(\alpha, y') \rightarrow \exists \alpha \in I, f(\alpha, y) < f(\alpha, y')$. Et, à cause de la linéarité et croissance stricte en α , $\forall \alpha \in \{\alpha_{min}, \alpha_{max}\}, f(\alpha, y) \leq f(\alpha, y') \rightarrow \exists \alpha \in I, f(\alpha, y) \leq f(\alpha, y')$. On a donc prouvé que $I(y, y') \leftrightarrow I'(y, y')$ pour le cas où $y_1 > y_2$. Le cas invers est similaire, et donc on a prouvé que les deux dominances sont équivalentes pour tous les cas. Q.E.D.

Alors nous devons maintenant transformer une instance Π du problème de détermination des points non I -dominés en une instance Π' du problème de détermination des points non-dominés (au sens de Pareto). On peut comparer αy_1 avec $\alpha y'_1$ et $(1 - \alpha) y_2$ avec $(1 - \alpha) y'_2$ pour $\alpha \in \{\alpha_{min}, \alpha_{max}\}$. Une modification dans la fonction "lexialgo" dans le fichier part4.py montre cette transformation.

4.3 QUESTION 12

L'exécution du programme part4.py montre les résultats d'exécution, et le fichier part4.png montre des courbes générés par Excel par les données dans part4.txt.