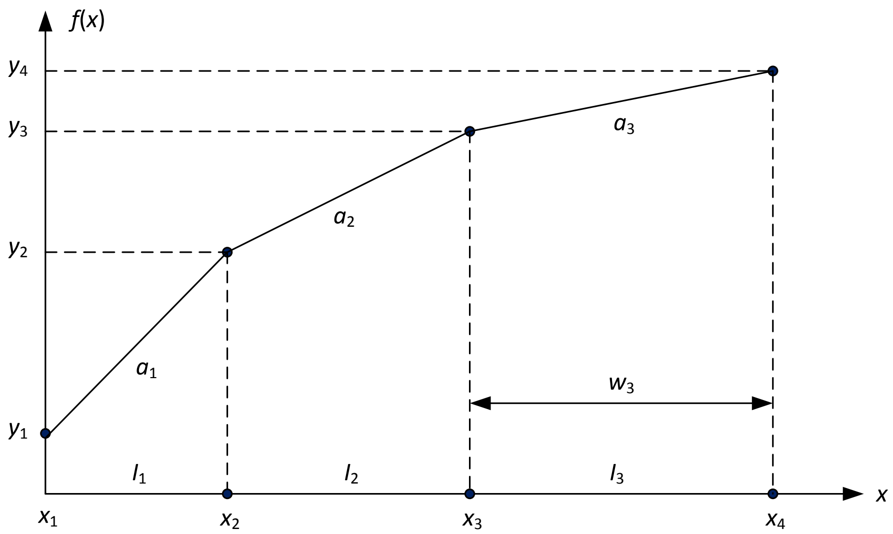
Fonctions linéaires par morceaux

1. Introduction

Les fonctions linéaires par morceaux (*piecewise linear functions*) sont fréquentes en optimisation, par exemple pour modéliser des coûts dégressifs ou approximer des fonctions non linéaires. La figure illustre le coût d'une commande en fonction du nombre d'articles x : après un coût fixe y_1, la fonction comporte trois morceaux avec des coûts unitaires décroissants a_1, a_2, a_3. Comment linéariser cette fonction?

Dans la suite, on considère une fonction $f(x)$ avec p morceaux linéaires. Chaque morceau k, $k = 1 \dots p$, est défini sur un intervalle $I_k = (x_k, x_{k+1})$ de largeur $w_k = x_{k+1} - x_k$, qui forme un segment de pente $a_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$ joignant les points (x_k, y_k) et (x_{k+1}, y_{k+1}).



2. Méthode

On utilise $p - 1$ variables binaires z_k, égales à 1 ssi l'intervalle k est complètement consommé, et p variables u_k qui représentent combien on consomme dans chaque intervalle, voir les contraintes (7). La variable x peut se décomposer selon la contrainte (2) et la fonction-objectif selon l'équation (1) :

(1) $f(x) = y_1 + \sum_{k=1}^{p} a_k u_k$

(2) $x = x_1 + \sum_{k=1}^{p} u_k$

(3) $\forall k = 1 \dots p-1 : w_k z_k \leq u_k$

(4) $\forall k = 1 \dots p-1 : \ u_{k+1} \leq w_{k+1} z_k$

(5) $\forall k = 1 \dots p-2 : \ z_k \geq z_{k+1}$

(6) $\forall k = 1 \dots p-1 : \ z_k \in \{0,1\}$

(7) $\forall k = 1 \dots p : \ 0 \leq u_k \leq w_k$

Le nouvel objectif est valide si les intervalles sont consommés de gauche à droite. En d'autres termes, si l'intervalle j est incomplètement consommé ($0 < u_j < w_j$), alors les intervalles précédents doivent être complètement consommés tandis que les intervalles suivants doivent être non entamés.

Ceci est assuré grâce aux contraintes (2) à (6). Montrons que ces contraintes sont valides.

Preuve de validité :

D'après (5), le vecteur z est formé d'une suite de 1 suivie d'une suite de 0, une des suites pouvant être vide. Sans perte de généralité, supposons que le premier 0 soit à l'indice j, c'est-à-dire que $z_{j-1} = 1$ et $z_j = 0$.

Pour $k = 1, 2, \dots, j-1$, on a donc $z_k = 1$. Dans ce cas, les contraintes (3) équivalent à $w_k \leq u_k$ tandis que les contraintes (4) deviennent $u_{k+1} \leq w_{k+1}$, ce qui est déjà assuré par (7). Comme les contraintes (7) précisent que $w_k \geq u_k$, on donc a $u_k = w_k$ pour $k = 1, 2, \dots, j-1$. En clair, les intervalles I_1 à I_{j-1} sont complètement consommés.

Pour $k = j, j+1, \dots, p-1$, les contraintes (3) deviennent $u_k \geq 0$ mais cela est déjà garanti par les contraintes (7). Les contraintes (4) sont équivalentes à $u_{k+1} \leq 0$. Comme $u_{k+1} \geq 0$ d'après (7), ces variables sont donc nulles, ce qui signifie que les intervalles I_{j+1} à I_p ne sont pas entamés.

Finalement, u_j peut varier librement entre 0 et w_j. □

Considérons par exemple la fonction de la figure avec ∈ I_2. Les contraintes (2) à (6) impliquent dans ce cas que $z_1 = 0$, $z_2 = 1$, $u_1 = w_1$ et $u_3 = 0$. Tout se passe comme si le modèle se réduisait à :

(1) $f(x) = y_1 + w_1 a_1 + a_2 u_2$
(2) $x = x_1 + w_1 + u_2$
(7) $0 \leq u_2 \leq w_2$

Remarques

1. Si f est convexe, croissante et doit être minimisée, les contraintes (3) à (6) sont inutiles. En effet, un solveur va utiliser au maximum le premier intervalle (le moins coûteux) avant d'entamer le second.

2. En enlevant $u_p \leq w_p$ dans les contraintes (7), on peut gérer des fonctions dont le dernier intervalle est ouvert à droite. On peut aussi conserver w_p mais lui donner une grande valeur.

3. Si la fonction passe par l'origine, x_1 et y_1 peuvent être supprimés.

4. La modélisation se simplifie pour $p = 2$ morceaux. Elle ne nécessite que la variable binaire z_1 pour le premier intervalle. On la nomme plus simplement z et la contrainte (6) n'a plus lieu d'être. Il reste :

   (1) $f(x) = y_1 + a_1 u_1 + a_2 u_2$

   (2) $x = x_1 + u_1 + u_2$

   (3) $w_1 \cdot z \leq u_1$

   (4) $u_2 \leq w_2 \cdot z$

   (6) $z \in \{0,1\}$

   (7) $0 \leq u_1 \leq w_1, 0 \leq u_2 \leq w_2$

   Si $z = 1$, alors on a $u_1 = w_1$ et u_2 quelconque. Si $z = 0$, on a u_1 quelconque et $u_2 = 0$.

5. Dans certaines applications, on a besoin de savoir à quel intervalle x appartient. Le numéro d'intervalle est égal à $1 + \sum_{k=1}^{p-1} z_k$.