Arbre enraciné de profit maximum avec des contraintes de capacité

Martin Debouté et Basile Blayac

10 janvier 2022

Table des matières

1 Introduction

Étant donné un ensemble de sites V et une matrice de distance $D = (d_{ij})$ entre chaque paire de sites, le problème du p-centre consiste à sélectionner p sites de V afin qu'ils agissent comme des centres et à affecter les autres sites au centre disponible le plus proche tout en minimisant la plus grande distance entre un site et son centre. Des contraintes supplémentaires, telles que des contraintes de précédence ou des contraintes de capacité, peuvent apparaître dans certaines applications. Ces problèmes sont souvent résolus par des méthodes de décomposition où le problème original est décomposé en sous-problèmes consistant à construire des centres réalisables.

Le but de ce projet est de proposer des modèles mathématiques pour la résolution d'un sous-problème associé à un site donné jouant le rôle de centre.

2 Présentation du problème

Soient $V = \{0, \ldots, n-1\}$ un ensemble de sites et $r \in V$ un centre prédéfini. A chaque site $i \in V$, on associe deux poids w_i^1 et w_i^2, une distance d_i entre le site i et le centre r, et un profit p_i (pouvant être négatif). On définit également pour chaque site i un prédécesseur $\pi_i \in V$ devant être pris si i est pris dans la solution. On suppose que les données satisfont la propriété que la distance au centre d'un sommet i est supérieure ou égale à la distance au centre de son prédécesseur π_i. L'ensemble des prédécesseurs défini un arbre enraciné en le centre r. Soient C^1 et C^2 deux capacités.

Le problème consiste à sélectionner un ensemble de sommets qui seront affectés au centre r, dont le poids total (centre inclus), pour chacun des poids w^1 et w^2 est inférieur aux capacités C^1 et C^2, et tel que si un sommet i est sélectionné, alors sont prédécesseur π_i est également sélectionné.

L'objectif du problème est de maximiser le profit total des sites sélectionnés (en incluant le centre r) auquel on soustrait la distance au centre (également appelée rayon) du sommet sélectionné le plus éloigné.

Si pour un site i, son prédécesseur vaut $\pi_i = -1$, soit i est le centre r et il doit être pris dans la solution, soit le site i n'est pas le centre et auquel cas, le site ne peut pas être pris dans la solution (*cf.* l'implémentation des modèles en Python où l'on définit un $x_{-1} = 0$).

2.1 Premier modèle

Considérons les variables :

— $x_i = \begin{cases} 1 \text{ si le site } i \in V \text{ est sélectionné,} \\ 0 \text{ sinon.} \end{cases}$

— $R \in \mathbb{R}$ la valeur du rayon (la distance au centre du sommet sélectionné le plus éloigné).

Nous pouvons donc écrire notre modèle sous cette forme :

$$\max \sum_{i \in V} p_i x_i - R \tag{1}$$

$$sc \sum_{i \in V} w_i^1 x_i \leq C^1 \tag{2}$$

$$\sum_{i \in V} w_i^2 x_i \leq C^2 \tag{3}$$

$$x_i \leq x_{\pi_i}, \forall i \in V \setminus \{r\} \tag{4}$$

$$x_i d_i \leq R, \forall i \in V \tag{5}$$

$$x_r = 1 \tag{6}$$

L'objectif (1) consiste à maximiser la somme des p_i tels que $x_i = 1$, moins la valeur R qui correspond au rayon. Les contraintes (2) et (3) s'assurent que la somme des w_i^1 et w_i^2 respectivement pour $x_i = 1$ ne dépassent pas C^1 et C^2 respectivement. La contrainte (4) vérifie que si le prédécesseur d'un x_i est nul alors x_i est nul. La contrainte (5) s'assure que R est bien supérieur à tous les d_i pour lesquels $x_i = 1$. Et enfin, (6) on ne veut pas de solution avec un vecteur x nul donc $x_r = 1$.

2.2 Second modèle

Il est possible de modéliser le rayon de manière plus précise. Pour cela, on définit la fonction $\sigma : V \mapsto \{1, \ldots, n\}$ qui associe à chaque site $i \in V$ sa position dans l'ordre croissant des distances des sites au centre et $\sigma^{-1}(k)$ la fonction inverse qui renvoie le site en position k dans cet ordre ($0 = d_r = d_{\sigma^{-1}(1)} \leq d_{\sigma^{-1}(2)} \leq \cdots \leq d_{\sigma^{-1}(n)}$). Pour tout site $i \in V \setminus \{r\}$ différent du centre r, on pose $\Delta_i = d_i - d_{\sigma^{-1}(\sigma(i)-1)}$ la différence de distance entre le site i et son prédécesseur dans l'ordre défini pas σ.

Considérons les variables suivantes :

— $x_i = \begin{cases} 1 \text{ si le site } i \in V \text{ est sélectionné,} \\ 0 \text{ sinon.} \end{cases}$

— $y_i = \begin{cases} 1 \text{ si le site } i \in V \text{ a une distance plus petite ou égale que le rayon du centre,} \\ 0 \text{ sinon.} \end{cases}$

Notons qu'un sommet peut avoir une distance plus petite que le rayon, sans nécessairement être sélectionné dans la solution. En revanche, un sommet sélectionné doit forcement avoir une distance plus petite ou égale au rayon. Notez également que le rayon d'un centre est la somme des Δ_i sur tous les sites i ayant une distance plus petite ou égale au rayon.

Nous pouvons donc écrire notre modèle sous cette forme :

$$\max \sum_{i \in V} p_i x_i - \sigma(i) y_i \tag{7}$$

$$sc \sum_{i \in V} w_i^1 x_i \leq C^1 \tag{8}$$

$$\sum_{i \in V} w_i^2 x_i \leq C^2 \tag{9}$$

$$x_i \leq x_{\pi_i}, \forall i \in V \setminus \{r\} \tag{10}$$

$$\sum_{k=i}^{n} x_{\sigma^{-1}(k)} \leq y_{\sigma^{-1}(i)}(n-i), \forall i \in V \tag{11}$$

$$x_r = 1 \tag{12}$$

L'objectif (7) consiste à maximiser la somme des p_i tels que $x_i = 1$, moins la valeur du rayon qui ici est égale la somme des d_i inférieur au rayon, donc tel que leur $y_i = 1$. Les contraintes (8) et (9) s'assurent que la somme des w_i^1 et w_i^2 respectivement pour $x_i = 1$ ne dépassent pas C^1 et C^2 respectivement. La contrainte (10) vérifie que si le prédécesseur d'un x_i est nul alors x_i est nul. La contrainte (11) s'assure que chaque y_i est supérieur à la somme des x_k telle que d_k est supérieure ou égale à d_i. Et enfin, (12) on ne veut pas de solution avec un vecteur x nul donc $x_r = 1$.

3 Résultats

Nos deux modèles trouvent les solutions optimales de toutes les instances avec un facteur de tolérance relatif $rtol := 10^{-3}$ et un facteur de tolérance absolue $atol := 10^{-6}$. On vérifie expérimentalement (*cf.* le Jupyter Notebook 'bilan.ipynb') que :

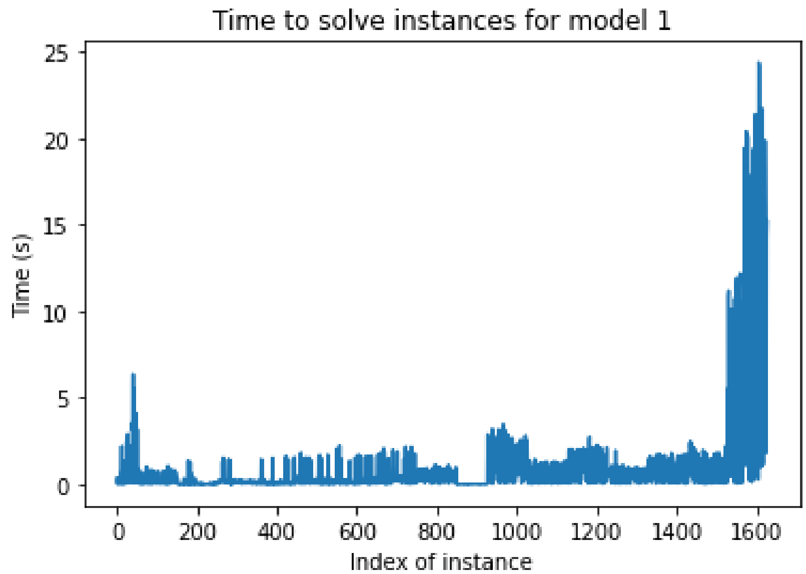
$|optimal\_solutions - solutions\_model\_1| \leq (atol + rtol * |solutions\_model\_1|)$,

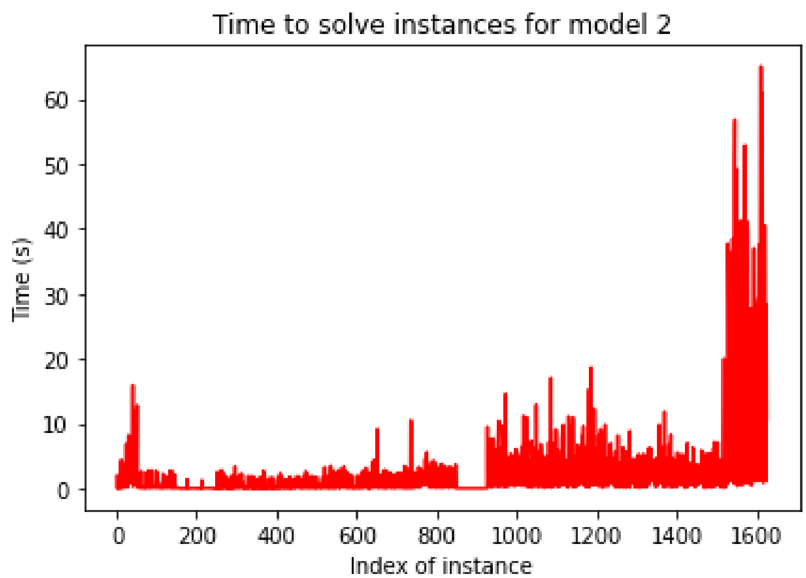
de même pour le second modèle,

$|optimal\_solutions - solutions\_model\_2| \leq (atol + rtol * |solutions\_model\_2|)$

et inversement.

Cependant les graphiques ci-dessous témoignent du fait que quelque soit la difficulté de l'instance, le premier modèle est en général deux fois plus rapide que le second modèle dans la résolution du problème :





Toutefois il est important de souligner que ces conclusions sont le résultat de nos expériences avec notre implémentation en Python, et donc que l'observation ≪ le premier modèle est meilleur que le second ≫ n'est pas une vérité générale.