UPMC SORBONNE UNIVERSITÉS

Module RP
Résolution de Problèmes

Projet

Placement de centres logistiques

On considère dans ce projet la résolution du problème de placement de centres logistiques en utilisant des techniques de résolution heuristique et exacte.

Un barême indicatif est indiqué pour chacun des exercices.

1 Définition du problème

Une nouvelle enseigne veut s'installer dans le pays et désire ouvrir un certain nombre de centres logistiques, c'est-à-dire des plateformes de stockage et de redistribution de ses produits. On considère ici un ensemble I de villes pouvant recevoir un centre logistique et $f_i \in \mathbb{R}_+$, $i \in I$, le coût d'installation d'un centre dans la ville i. Dans une ville qui ne contient pas de centre logistique, les magasins de cette enseigne seront gérés par un unique centre logistique situé dans une ville proche : on note $c_{ij} \in \mathbb{R}_+$, $i, j \in I$, le coût de gestion de la ville j par un centre ouvert dans la ville i. Le problème de placement de centres logistiques consiste ainsi à déterminer un ensemble S de villes où vont être installés des centres ainsi qu'une affectation $\sigma : (I \setminus S) \to S$ des villes sans centre à l'un des centre de S, de façon à ce que la somme des coûts d'installation et de gestion, *i.e*

$$\sum_{i \in S} f_i + \sum_{j \in I \setminus S} c_{\sigma(j)j}$$

soit minimale.

Ce problème s'appelle le *Uncapacitated Facility Location Problem* (UFLP). Il a été démontré NP-difficile (en tant que cas particulier du *set covering problem*). Lorsque $f_i = 0$ pour tout $i \in I$ et que l'on fixe un nombre p de centres à ouvrir, le problème est alors appelé *problème du p-médian*.

2 Partie théorique : Approximation pour le problème UFLP

On s'intéresse ici à un algorithme de 4-approximation pour le problème UFLP basé sur la programmation linéaire.

Dans cette partie, nous allons supposer que :
- toute ville $i \in I$ peut être affectée à tout centre placé dans la ville $j \in I$
- les coûts c sont symétriques, *i.e.* $c_{ij} = c_{ji}$, $\forall i, j \in I$,
- les coûts c respectent l'inégalité triangulaire, *i.e.*

$$c_{ij} \leq c_{ik} + c_{kj} \quad \forall i, j, k \in I.$$

Exercice 1 : Modélisation (*1 point)*

On considère le programme linéaire en nombres entiers (P) suivant :

$$Min \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in I \setminus \{i\}} c_{ij} x_{ij}$$

$$\begin{aligned}
x_{ij} &\leq y_i && \forall i \in I, \forall j \in I \setminus \{i\} && (1)\\
y_j + \sum_{i \in I \setminus \{j\}} x_{ij} &= 1 && \forall j \in I && (2)\\
y_i &\geq 0 && \forall i \in I &&\\
x_{ij} &\geq 0 && \forall i \in I, \forall j \in I \setminus \{i\} &&\\
y_j, x_{ij} &\text{ entiers} && \forall i \in I, \forall j \in I \setminus \{i\} && (3)
\end{aligned}$$

Q 1.1. Montrer que les variables x et y sont des variables binaires. Préciser à quoi correspondent les valeurs prises par les variables x et y.

Q 1.2. Montrer que résoudre (P) est équivalent à résoudre le problème UFLP.

On appelle $(\tilde{P})$ le programme linéaire (continu) obtenu en enlevant la contrainte d'intégrité (3) au programme linéaire en nombres entiers (P). On dit que $(\tilde{P})$ est la relaxation linéaire de (P).

On note v_j, $j \in I$, les variables duales associées aux contraintes (2) de $(\tilde{P})$. On réécrit les inégalités (1) comme étant

$$y_i - x_{ij} \geq 0 \quad \forall i \in I, \ \forall j \in I \setminus \{i\}$$

et on note alors w_{ij}, $i \in I$, $j \in I \setminus \{i\}$, les variables duales associées à ces contraintes dans $(\tilde{P})$.

Q 1.3. Prouver que le programme linéaire $(\tilde{D})$ suivant est le dual de $(\tilde{P})$.

$$
\begin{aligned}
Max \quad & \sum_{i \in I} v_i & & \\
& \sum_{j \in I \setminus \{i\}} w_{ij} + v_i \leq f_i & \forall i \in I & (4) \\
& v_j - w_{ij} \leq c_{ij} & \forall i \in I,\ \forall j \in I \setminus \{i\} & (5) \\
& w_{ij} \geq 0 & \forall i \in I,\ \forall j \in I \setminus \{i\} & \\
& v_i \in I\!R & \forall i \in I &
\end{aligned}
$$

Exercice 2 : Etude de l'algorithme : partie 1 *(2 points)*

On considère une solution optimale (x^*, y^*) de $(\tilde{P})$ et une solution optimale (v^*, w^*) du dual $(\tilde{D})$. En utilisant des propriétés sur l'arrondi de (x^*, y^*), nous allons produire une solution entière pour (P) qui sera au plus 4 fois le coût d'une solution entière optimale de (P).

On définit le graphe $G_{x^*} = (I, E)$ ayant pour ensemble de sommets I et possédant une arête $ij \in E$ si et seulement si $x^*_{ij} + x^*_{ji} > 0$.

Q 2.1. En utilisant le théorème des écarts complémentaires, montrer que pour toute arête de G_{x^*}, on a

$$c_{ij} \leq v^*_j \tag{6}$$

On considère l'algorithme d'arrondi suivant

$k \leftarrow 1$
Tant qu'il existe une ville ni centre, ni affectée à un centre
  Prenons $j_k \in I$ une ville ni centre, ni affectée à un centre
  et ayant la valeur $v^*_{j_k}$ la plus petite possible
  $C_k \leftarrow \{j_k\}$
  Pour tout $i \in I$ non rencontré jusqu'ici
    et distant dans G_{x^*} d'au plus 2 arêtes de j_k
    $C_k \leftarrow C_k \cup \{i\}$
  Fin Pour tout
  On note alors i_k le sommet avec f_{i_k} minimum dans C_k
  $k \leftarrow k + 1$
Fin Tant que

Le coût de la solution fournie par l'algorithme est donc

$$z_{algo} = \sum_{l=1}^{k} f_{i_l} + \sum_{l=1}^{k} \sum_{j \in C_l} c_{i_l j}$$

Nous allons montrer que cette solution est au plus 4 fois la solution optimale entière de (P).

On s'intéresse tout d'abord à borner le deuxième terme de la fonction objective.

Q 2.2. Montrer que, pour $l \in \{1, \ldots, k\}$,

$$c_{i_l j} \leq c_{ij} + c_{ij_l} + c_{i_l j_l} \quad \forall l \in \{1, \ldots, k\} \;\; \forall i, j \in C_l$$

Q 2.3. En utilisant le résultat 6 et l'algorithme, en déduire que

$$c_{i_l j} \leq 3v_j^* \quad \forall j \in C_l \tag{7}$$

Exercice 3 **: Etude de l'algorithme : partie 2** *(2 points)*

On s'intéresse à présent à borner le premier terme de la fonction objective.

Q 3.1. Montrer que,

$$x_{ij_l}^* f_{il} \leq x_{ij_l}^* f_i \quad \forall l \in \{1, \ldots, k\} \;\; \forall i \in I \setminus \{j_k\}$$

Q 3.2. En utilisant les inégalités de (P), en déduire que

$$f_{i_l} \leq y_{j_l}^* f_{i_l} + \sum_{i \in I \setminus \{j_l\}} x_{ij_l}^* f_i \quad \forall l \in \{1, \ldots, k\}$$

On remarque que, dans l'inégalité précédente,

$$\sum_{i \in I \setminus \{j_l\}} x_{ij_l}^* f_i = \sum_{(i,j_l) \in E} x_{ij_l}^* f_i$$

par construction des arêtes de G.

Q 3.3. Montrer que

$$f_{i_l} \leq y_{j_l}^* f_{i_l} + \sum_{(i,j_l) \in E} y_i^* f_i \quad \forall l \in \{1, \ldots, k\}$$

En sommant ces inégalités, on obtient

$$\sum_{l=1}^{k} f_{i_l} \leq \sum_{l=1}^{k} y_{j_l}^* f_{j_l} + \sum_{l=1}^{k} \sum_{(i,j_l) \in E} y_i^* f_i$$

De plus, par construction des ensembles C_l, il apparaît que j_l et $j_{l'}$ pour deux ensembles C_l et $C_{l'}$ distincts n'ont aucun sommet i en commun. Ainsi, on obtient

$$\sum_{l=1}^{k} f_{i_l} \leq \sum_{l=1}^{k} y_{j_l}^* f_{j_l} + \sum_{i \in I \setminus \{j_1, \ldots, j_k\}} y_i^* f_i$$

D'où le résultat

$$\sum_{l=1}^{k} f_{i_l} \leq \sum_{i \in I} y_i^* f_i \tag{8}$$

On en déduit donc que

$$z_{algo} \leq \sum_{i \in I} y_i^* f_i + 3 \sum_{j \in I} v_j^*$$

Notons z_{opt} la solution optimale entière de (P) et z^* la solution optimale de $(\tilde{P})$ et $(\tilde{D})$.

Q 3.4. En utilisant l'écriture du dual $(\tilde{D})$, montrer que $\sum_{j \in I} v_j^* \leq z_{opt}$.

Q 3.5. Montrer que $\sum_{i \in I} y_i^* f_i \leq z_{opt}$

Q 3.6. En déduire que l'algorithme produit une 4-approximation du problème UFLP.

3 Partie expérimentale : méthodes de résolution

Sur la page http ://www-desir.lip6.fr/∼fouilhoux/documentens.php, vous trouverez plusieurs documents.

Exercice 4 : Manipulation des instances *(2 points)*

Nous allons travailler sur des instances créées à partir des coordonnées (longitude, latitude) des villes française. Vous trouverez les caractéristiques de ces villes dans le fichier villes_france.csv. Ce fichier contient pour chaque commune numérotée de 1 à 36830, son département, son nom (dans un format simplifié), sa population au 1er janvier 2010 et ses coordonnées.

Plusieurs instances de tailles diverses pour le problème sont données au format .flp suivant :

```
3 p
0 num0 f0
1 num1 f1
2 num2 f2
0 1 c_0_1
0 2 c_0_2
1 0 c_0_1
2 0 c_2_0
```

- où 3 est le nombre de villes,
- si $p \geq 1$ correspond au problème p-median où p est le nombre de centres à déterminer ; ou alors $p = -1$ si l'instance est une instance de l'UFLP,
- les 3 lignes suivantes correspondent aux villes : un numéro repérant la ville, le numéro de la ville dans l'encodage du fichier villes_france.csv et son poids f,
- les 4 suivantes indiquent les affectations possibles ville à ville en indiquant les numéros (i, j) de la liste précédente et le coût de gestion c_{ij} correspondant à cette affectation.

Ces instances sont basées sur les villes françaises ou les villes des départements : certaines sont limitées aux grandes villes. **Attention :** toutes les villes ne sont pas utilisables en tant que centre pour toutes autres villes.

Les instances fournies sur le site du projet ne concernent que le problème UFLP. Les instances pour le problème p-median s'obtiennent en ignorant le coup f_i d'installation et en choisissant par vous-même des valeurs pour p.

Q 4.1. Implémenter une méthode de chargement des instances dans l'objectif des trois parties suivantes (méthodes exactes, arrondi et heuristiques).

Q 4.2. Réaliser une visualisation de ces instances et surtout des solutions. Vous pouvez par exemple utiliser facilement le format postscript (voir documentation sur la page citée plus haut).

Exercice 5 : Méthode exacte *(3 points)*

En s'inspirant du programme linéaire en nombres entiers proposés dans la partie théorique, on peut facilement déduire des formulations du problème UFLP et p-médian lorsque toutes les villes ne peuvent pas servir de centres pour toutes les autres villes.

Q 5.1. En utilisant un solveurs de votre choix (gurobi, cplex, xpress, glpk), implémenter une méthode permettant de lire les instances des problèmes UFLP et p-médian.

Q 5.2. Tester les capacités de résolution de ce solveur sur les instances proposées.

Exercice 6 : Méthode d'approximation par arrondi *(3 points)*

On reprend ici la méthode de 4-approximation par arrondi introduite dans la partie théorique du projet.

Q 6.1. En utilisant le langag de votre choix, implémenter cette méthode d'arrondi.

Q 6.2. Comparez les solutions obtenues aux solutions exactes obtenues par un solveur entier.

Exercice 7 : Méthode heuristique *(7 points)*

Lorsque les instances sont de dimensions trop importantes, les solveurs entiers nécessitent un temps trop important pour être utilisable pour le problème. On se propose de développer une méthode heuristique pour le résoudre.

Une documentation sur la mise en œuvre de méthodes heuristiques est en ligne sur la page indiquée plus haut.

Q 7.1. En utilisant un langage de programmation de votre choix, vous développerez une méthode de recherche heuristique (descente stochastique, recuit simulé, méthode tabou ou algorithme génétique) pour le problème UFLP et p-median.

En utilisant la relaxation linéaire des formulations entières de la partie précédente, on obtient une borne inférieure z_b sur la valeur optimale entière. A tout moment de votre méthode heuristique, on peut considérer la meilleure solution réalisable rencontrée $z_{bestsol}$. On peut donc considérer tout au long du déroulement de l'algorithme l'augmentation de la borne inférieure et la réduction de la borne supérieure. On mesure cet écart en tant qu'écart relatif appelé

souvent gap :

$$gap = \frac{z_{bestsol} - z_b}{z_{bestsol}}.$$

Ainsi, dans le pire des cas, la borne supérieure est égale à z_{opt} : on a ainsi $(1+gap)z_{bestsol} = z_{opt}$ ce qui signifie qu'au pire $z_{bestsol}$ est à $gap\%$ de l'optimum.

Q 7.2. Mener une analyse expérimentale des capacités de votre algorithme en utilisant la mesure de gap.

4 Documents à fournir

Vous rédigerez un petit rapport contenant la description de votre travail, c'est-à-dire :
- la réponse aux questions théoriques
- une description (rapide) de la mise en œuvre de la méthode exacte
- un description synthétique de votre méthode heuristique
- les jeux d'essais et votre analyse

Joignez également votre code et vos jeux d'essais.

Dates du projet

Sujet distribué en TD **le 26 mars**.
Tous les documents sont à rendre en TD **le 5 mai**.
Une mini-soutenance aura lieu la semaine du **4 mai** (date à préciser).