Sorbonne Université 22 Octobre 2018
MASTER ANDROIDE M1 – MOGPL

Projet MOGPL

Résolution de problèmes d'affectation et de localisation

Introduction et objectifs

Le problème de localisation équitable modélise un problème réel consistant à localiser des services (hopital, administration, caserne de pompiers, bibliothèque, piscine) sur une zone géographique de manière à partager équitablement l'accès à ces ressources par les populations. Selon les cas, la notion d'équité peut porter sur la distance, le temps ou plus généralement le coût d'accès des utilisateurs potentiels (clients) à la ressource qui leur est affectée. Dans ce sujet, on s'intéresse à un seul type de ressource (hopital ou piscine ou bibliothèque etc) et on se place du point de vue d'un superviseur (agent, logiciel de contrôle) qui connaît la position géographique des clients et celle des sites dans lesquels une ressource peut être implantée et qui, étant donné le coût d'accès de chaque client à chaque site, doit déterminer une localisation des ressources répartissant "optimalement" les coûts d'accès des clients à ces ressources. L'objet du projet est d'implanter et tester des modèles computationnels s'appuyant sur la programmation linéaire en variables réelles, entières ou mixtes et permettant au superviseur de déterminer des solutions équitables à ce problème de localisation (la notion d'équité restant à préciser).

Travail à réaliser

Soient $I = \{1, \ldots, n\}$ un ensemble de n villes situées sur le territoire de référence et v_i la population de la ville i pour tout $i \in N$. On suppose ici qu'on a déjà choisi les k villes dans lesquelles les ressources seront localisées ($k < n$) et on note $J \subseteq I$ l'ensemble des indices de ces villes ($|J| = k$). Lorsqu'une ressource est localisée dans une ville j ($j \in J$) on doit définir son secteur de service, c'est-à-dire l'ensemble des villes qui auront accès à la ressource située en j. Le secteur de service d'une ressource située dans la ville j est caractérisé par les variables $x_{ij}, i \in I$ qui valent 1 si j est le point d'accès des habitants de la ville i à la ressource (on dit alors que i appartient au secteur j) et 0 sinon. La matrice $n \times k$ de terme général x_{ij} sera notée X et caractérise une solution potentielle (réalisable ou non) pour la définition des k secteurs. Bien entendu, plusieurs villes peuvent appartenir à un même secteur de service. En revanche une ville n'appartient qu'à un unique secteur et les secteurs forment une partition des n villes.

Pour des raisons de capacité de service et d'équilibrage des charges, la population totale des villes composant un secteur ne doit pas dépasser la quantité $\gamma = \frac{1+\alpha}{k} \sum_{i=1}^{n} v_i$ où α est un paramètre strictement positif donné (par exemple $\alpha = 0.1$). On note d_{ij} la distance moyenne qui sépare un habitant de la ville i d'une ressource située en j et on suppose les distances d_{ij} connues pour tout couple (i, j). L'intérêt du maire d'une ville i est que sa ville se trouve proche de la ville j où se trouve la ressource dont sa commune dépend. On suppose donc que le coût (disutilité) de se voir affecté au secteur j pour le maire de la ville i est mesuré par d_{ij}.

1°) Dans ce contexte, formuler un programme linéaire en variables mixtes qui détermine les secteurs de service des k ressources de manière à minimiser le coût social de la solution, défini par la fonction :

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{k} d_{ij} x_{ij}$$

Appliquer ce programme mathématique sur les données fournies dans le répertoire `data` accessible sur le site Moodle de l'UE. Ces données concernent les 36 villes des Hauts de Seine. On calculera les solutions f-optimales pour $k = 3, 4, 5$ (en choisissant soi-même la localisation des ressources) avec $\alpha = 0.1$ ou $\alpha = 0.2$ et on visualisera les solutions trouvées sur la carte. Pour chaque solution trouvée, déterminer la satisfaction moyenne des maires et la satisfaction du maire le moins bien servi (mesurée par la distance de sa ville à la ressource qui lui est affectée).

2°) On envisage maintenant de minimiser la fonction suivante :

$$g(x) = \max_{i=1,\dots,n} \sum_{j=1}^{k} d_{ij} x_{ij} + \epsilon f(x)$$

avec $\epsilon = 10^{-6}$. Expliquer l'intérêt de minimiser la fonction g plutôt que la fonction f. Calculer les solutions g-optimales pour $k = 3, 4, 5$ sur les mêmes instances que celles résolues à la question 1. On visualisera ici encore les solutions trouvées sur la carte et, pour chaque solution trouvée, on déterminera la satisfaction moyenne des maires et la satisfaction du maire le moins bien servi. Calculer pour chaque instance considérée le *prix de l'équité* défini par le ratio

$$PE = 1 - \frac{f(x^*_f)}{f(x^*_g)}$$

où x^*_fet x^*_g sont respectivement les solutions trouvées en minimisant f et g. Que peut-on conclure ici ?

3°) On suppose maintenant que la localisation des ressources n'est plus donnée a priori et que seul le nombre de ressources à localiser est connu. Reprendre le problème formulé à la question 2 et l'étendre à ce cas plus général pour déterminer la localisation optimale de k ressources et les secteurs de service associés. La solution ne consiste évidemment pas à énumérer explicitement les sous ensembles de k villes parmi n pour appliquer le modèle de la question précédente, mais plutôt à modifier le programme mathématique utilisé à la question 2 pour intégrer le fait que les positions des ressources font maintenant partie des variables de décision. Appliquer le nouveau modèle proposé aux données des Hauts de Seine pour $k = 3, 4, 5$. Comparer la qualité des résultats obtenus avec ceux obtenus lorsque les localisations étaient choisies 'à la main'.

Mise en oeuvre avec Gurobi

Il vous est suggéré de développer vos programmes en Python (mais d'autres langages sont possibles si vous préférez) et d'utiliser Gurobi pour résoudre les problèmes d'optimisation. Gurobi est disponible sur les machines situées en salle TME. Pour ceux qui ont un ordinateur, il est aussi possible de charger sur votre machine individuelle la version éducation du solveur gurobi (http ://www.gurobi.com/) dans sa version individuelle, en vous connectant avec votre machine depuis le domaine de l'université.

Data

Pour effectuer des tests numériques et pouvoir visualiser les résultats, les fichiers de données suivants vous sont fournis. En complément, vous pourrez éventuellement utiliser d'autres données et d'autres cartes pour des tests complémentaires.

- villes92 : liste des 36 villes du 92,
- 92.gif et 92.png : cartes du département 92 avec positionnement des 36 villes,
- coordvilles92.txt : coordonnées des villes sur la carte fournie,
- distances92.txt : distances d_{ij} réelles entre les villes,
- populations92.txt : populations des villes.

Echéances et déliverables

Les binômes devront remettre un rapport (de 5 pages environ, ne pas dépasser 10 pages) au format pdf contenant une description des modèles et tests réalisés et les réponses aux questions, ainsi qu'une archive compressée (.zip) contenant leurs programmes (commentés). Le rapport et l'archive sont à envoyer par e-mail au responsable du groupe dans lequel vous êtes inscrit au plus tard le mardi 11 décembre 2018 à 23h59. Une soutenance aura lieu la semaine du 17 décembre en salle machine (sur le créneau de la dernière scéance de TD-TME). Chaque binôme doit donc avoir son code prêt à être éxécuté sur son compte à la PPTI. Les binômes présenteront en 5 minutes leur travail (simplement sur la base du code, pas de transparent ni autre documents), puis répondront aux questions. Il pourra être demandé d'exécuter le code sur des instances autres que celle du projet. Votre programme devra donc pouvoir résoudre des instances fournies sur des fichiers .txt (au même format que ceux utilisés donnés dans l'archive). Par ailleurs, il est impératif de remettre le rapport et d'exécuter la soutenance dans le groupe où vous êtes inscrit.