Projet MOGPL

Optimisation appliquée à la localisation d'unités de soin et à la prise en charge des patients

Introduction et objectifs

Le problème de localisation d'unités de soins et de gestion de patients sur plusieurs zones géographiques de manière à répartir la charge des services et optimiser la prise en charge des patients est un sujet d'actualité. L'objet du projet est, dans le cadre de scénarios simplifiés pour les besoins de l'exercice, d'implanter et tester des méthodes d'optimisation vues en MOGPL pour optimiser la localisation d'unités de soin ou l'affectation de patients à des unités existantes et répartir la charge au cours du temps.

1 - Répartition de patients dans les unités de soin

Soient $I = \{1, \dots, n\}$ 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 se place dans un premier scénario où k unités spéciales de traitement des patients atteints par une forme sévère d'une nouvelle maladie ont été implantées dans k villes de I, $(k < n)$ et on note $J \subset I$ l'ensemble des indices de ces villes. Pour chaque unité $j \in J$ on doit définir son secteur de service. La définition des secteurs de service est caractérisée par les variables x_{ij}, $i \in I, j \in J$ qui valent 1 si l'on décide que les patients de la ville i seront traités, en cas de besoin, dans l'unité j (on dit alors que i appartient au secteur j) et 0 sinon. La matrice $n \times k$ de terme général x_{ij} caractérise une solution potentielle pour la définition des k secteurs. Bien entendu, plusieurs villes peuvent appartenir à un même secteur. 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 $\alpha = 0.1$ ou $\alpha = 0.2$). On note d_{ij} la distance moyenne qui sépare un habitant de la ville i d'une unité de soin située en j et on suppose les distances d_{ij} connues pour tout couple (i, j) (il s'agit en fait des distances entre les villes, voir annexe).

1.1 Dans ce contexte, formuler un programme linéaire (éventuellement en variables Booléenes) qui détermine les secteurs de service des k unités de soin de manière à minimiser la distance moyenne de chaque habitant à l'unité de soin dont il dépend (on ne prend en compte que les habitants des villes $v_i, i = 1, \dots, n$) . Peut-on traiter le même problème par un algorithme de flot maximum à coût minimum (dans l'affirrmative on expliquera comment, dans la négative on expliquera pourquoi).

1.2 Coder le programme linéaire et faire une application numérique en utilisant les 15 villes données en annexe. On choisira 3 villes arbitrairement dans lesquelles on supposera qu'une unité de soin spéciale a été implantée et on calculera alors les secteurs de services de ces 3 unités. On choisira ensuite 4 villes au lieu de 3 et on reprendra la même question. On choisira $\alpha = 0.1$ puis $\alpha = 0.2$ (voire plus s'il n'existe pas de solution réalisable avec des valeurs plus petites). On comparera ensuite la qualité des solutions obtenues.

2 - Localisation optimale des unités de soin

On considère maintenant un deuxième scénario dans lequel la localisation des unités de soin n'est pas encore décidée et que seul le nombre k de telles unités est connu. Il s'agit alors de déterminer la localisation des unités de soin et de définir leurs secteurs de service.

2.1 Reprendre le problème de la section précédente dans le cas de ce scénario plus général pour déterminer la localisation optimale de k unités de soin et les secteurs de service associés, toujours avec le même objectif et les mêmes contraintes. 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 linéaire proposé pour intégrer le fait que les positions des unités de soin font maintenant partie des variables de décision. Si besoin est, on pourra éventuellement limiter le nombre de variables x_{ij} en spécifiant un rayon maximal admissible pour un secteur de service (distance maximum d'une ville à l'unité dont elle dépend) ce qui permet aussi de limiter l'amplitude de déplacement des patients. Coder ce nouveau modèle et l'appliquer aux données pour $k = 3, 4, 5$ et donner les localisations optimales des unités. Comparer la qualité des solutions obtenues. Comparer également ces solutions à celles obtenues en première partie.

2.2 La solution précédente avantage les habitants des grandes villes qui ont plus de poids ; de plus elle ne permet pas de contrôler l'équité de l'accès au soin (distance de chaque individu à son unité de soin). Si $f(i)$ désigne le numéro de la ville dans laquelle est située l'unité de soin des habitants de la ville v_i, alors $d(i, f(i))$ représente la distance d'un individu de la ville v_i à son unité de soin. Une solution équitable pourrait alors consister à minimiser la quantité $\max_{i \in I} d(i, f(i))$. Ecrire un programme linéaire en variables mixtes qui reprend la question 2.1 avec ce nouvel objectif. Coder ce nouveau modèle et l'appliquer aux données pour $k = 3, 4, 5$ et donner les localisations des unités. Comparer la qualité des solutions obtenues à celle de la question précédente en terme de distance maximum d'un individu à son unité de soin.

3 - Equilibrage des charges des unités de soin

On considère maintenant un troisième scénario dans lequel 5 unités de soin ont été construites (par exemple celles trouvées à la question 2.2 pour k = 5) et sont dimensionnées pour accueillir chacune jusqu'à 100 patients. On se place alors dans une situation où la demande à un instant donné pour chaque unité est caractérisée par $p = (p_1, \ldots, p_5)$ où p_j est le nombre de patients du secteur j nécessitant une prise en charge (on suppose que ces patients sont arrivés dans leur unité de soin et que $\sum_{i=1}^{5} p_i \leq 500$.

3.1 On s'intéresse alors à équilibrer (autant que faire se peut) les charges des différentes unités en déplaçant des patients d'une unité à l'autre. Pour cela on cherche à trouver l'ensemble des déplacements à coût minimum réalisant cet équilibrage, le coût d'un déplacement étant proportionnel à la distance parcourue. Montrer que ce problème peut être formulé comme un problème de flot maximum à coût minimum dans un graphe que l'on représentera, ou comme un problème de transport.

3.2 Résoudre ce problème par la méthode de votre choix pour différentes instances de p, certaines ayant au moins une composante supérieure à 100.

Mise en oeuvre avec Gurobi

Nous recommandons une implantation en python. Pour résoudre les programmes linéaires, les étudiants pourront charger sur leur machine la version éducation du solveur gurobi (http ://www.gurobi.com/). Le solveur est également disponible en salle machine de la PPTI, accessible par ssh comme vu en tme.

Modalités, échéances et déliverables

Le projet est à réaliser en binôme. Les binômes devront remettre une archive zip contenant un fichier pdf nommé Nom1_Nom2_GRi.pdf (noms des auteurs et groupe) avec un rapport qui suivra le plan du sujet en répondant aux questions et en présentant les modèles proposés et les résultats des tests numériques, un court fichier readme.txt qui présentera les programmes réalisés et les informations utiles pour les utiliser, puis les programmes eux-mêmes. Les délivrables seront soumis sur moodle au plus tard le 3 Janvier 2021. Les soutenance de projet auront lieu la semaine du 4 Janvier 2021, lors des créneaux habituels des TDs.

Annexe

Les données utiles pour les applications numériques sont rassemblées ci-dessous et sont également disponibles dans les fichiers villes.xlsx et villes.csv figurant dans le répertoire du projet.

Population	Ville	Toulouse	Nice	Nantes	Montpellier	Strasbourg	Bordeaux	Lille	Rennes	Reims	Saint-Étienne	Toulon	Le Havre	Grenoble	Dijon	Angers
479553	Toulouse	0														
340017	Nice	562	0													
309346	Nantes	585	1143	0												
285121	Montpellier	242	326	824	0											
280966	Strasbourg	949	790	860	791	0										
254436	Bordeaux	245	803	347	482	942	0									
232787	Lille	894	1157	597	963	549	799	0								
216815	Rennes	697	1186	107	894	827	458	572	0							
182460	Reims	812	955	515	787	347	717	199	483	0						
172565	Saint-Étienne	536	490	662	322	550	530	749	699	546	0					
171953	Toulon	469	149	1051	233	866	711	1065	1092	862	398	0				
170147	Le Havre	848	1119	384	917	695	651	316	280	344	720	1026	0			
158454	Grenoble	532	465	786	297	533	654	803	823	600	155	372	767	0		
156920	Dijon	644	660	638	492	332	640	501	620	298	251	567	506	304	0	
152960	Angers	635	1062	88	771	773	396	511	126	429	574	969	295	699	551	0