MOGPL

Modélisations, Optimisation, Graphes
et Programmation Linéaire

Rapport de Projet

Étudiants:	*Numéros:*
Jonathan MORENO	3530148
Keyvan BEROUKHIM	3506789

Décembre 2018

1 Présentation du problème

1.1 Introduction

Nous cherchons ici à résoudre un problème de localisation et d'affectation de ressources. Étant donné un ensemble de n villes et un nombre de ressource k, on cherche à déterminer où positionner les ressources et à quelle ressource affecter chaque ville.

Il faut choisir k localisations parmi n puis une affectation parmi k pour chaque ville. Le nombre de combinaisons théoriques est donc $\binom{n}{k}k^n$.
Avec $n = 36$, on obtient :

k	1	2	3	4	5
combinaisons	36	10^{13}	10^{21}	10^{26}	10^{31}

Le nombre de combinaison augmente exponentiellement avec k, Dès $k = 2$, il n'est pas viable d'énumérer et d'évaluer toutes les combinaisons possibles.

Afin de répondre au problème, nous le modélisons par un **problème linéaire**. Ce problème est résolu en python avec l'aide du solveur de programme linéaire Gurobi.

1.2 Modélisation

Nous travaillons ici avec la carte des 36 plus grandes villes du département 92. On a donc le nombre de villes $n = 36$.
Une contrainte du problème est que la population totale couverte par une ressource ne doit pas être trop grande. La population maximale couverte par une ressource est définie par $\gamma = \frac{1+\alpha}{k} V$.
V étant la somme des populations de chaque villes.
Pour $\alpha = 0$, il faudrait que les k secteurs ne couvrent pas une population dépassant V/k. Toutes les villes doivent être couvertes donc chaque secteur devrait couvrir exactement V/k.
Avec α non nul, on autorise les secteurs à couvrir $\alpha\, V/k$ personnes supplémentaires, soit une augmentation proportionnelle à α.
On peut remarquer qu'il n'y a pas forcément de solution réalisable si α est trop petit.

Le nombre de ressources k et le facteur de relaxation α sont les paramètres d'instance du problème.

Le coût d'affectation d'une ville à une ressource est matérialisé par la distance qui les sépare. La matrice des distances D est connue pour tout i, j entre 1 et n (on remarque que les distances ne sont pas forcément symétriques).

Dans la suite du document, X représentera la matrice des variables d'affectation, i l'indice d'une ville et j l'indice d'une ressource. X_{ij} vaut 1 si la ville i s'approvisionne en j, et 0 sinon.
La distance entre la ville i et son point d'accès au service est obtenue en calculant :

$$d(i) = \sum_j D_{ij} X_{ij}$$

Afin qu'une ville soit dans un unique secteur, on ajoute les contraintes :

$$\forall i \ \ \sum_j X_{ij} = 1$$

Afin que la population couverte par la ressource j soit inférieure à γ, on ajoute les contraintes :

$$\forall j \ \ \sum_i V_i \, X_{ij} \leq \gamma$$

2 Minimisation de la moyenne

Le premier problème que l'on se pose est de fixer à la main les k secteurs et de déterminer les affectations minimisant la *moyenne* des distances. X est donc une matrice binaire de taille $n \times k$. Minimiser la moyenne des distances est équivalent à minimiser la somme des distances, la fonction objectif est donc :

$$f(X) = \sum_i \sum_j D_{ij} \, X_{ij}$$

Une fois l'affectation optimale calculée, on calcule la distance maximale entre une ville et sa ressource par la formule :

$$d_{max} = \max_i \sum_j D_{ij} X_{ij}$$

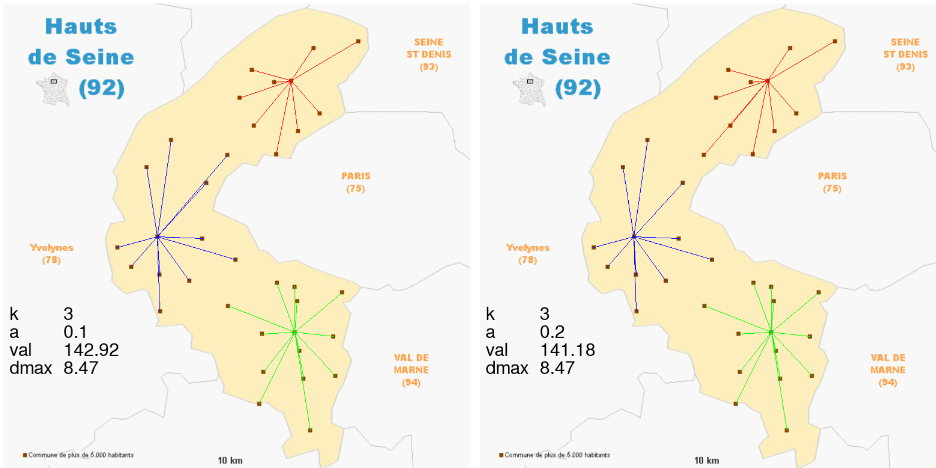

Figure 1: résultats obtenus avec k=3 pour des localisations de ressources fixées. ($\alpha = 0.1$ à gauche et $\alpha = 0.2$ à droite)
Ces images représentent la répartition optimale des villes du 92 en 3 groupes minimisant la distance moyenne entre les villes et leurs ressources.

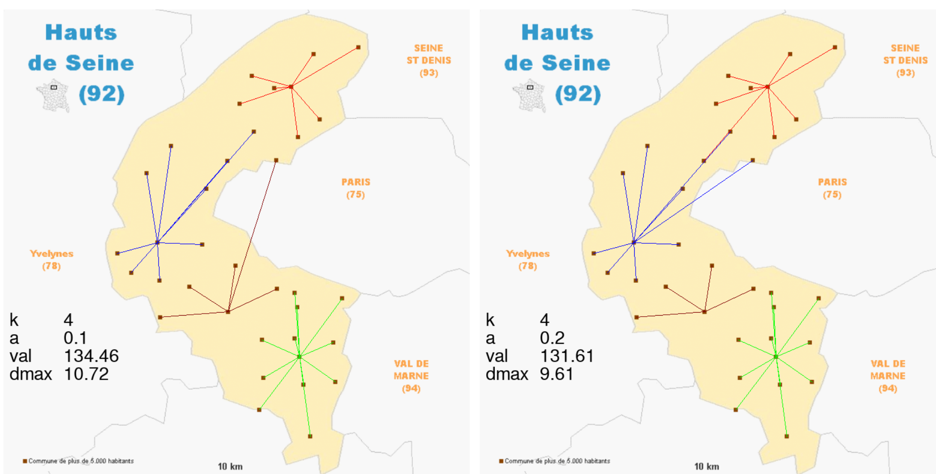

Figure 2: résultats obtenus avec k=4 pour des localisations de ressources fixées.
Relâcher un peu la contrainte en passant de $\alpha = 0.1$ (à gauche) à $\alpha = 0.2$ (à droite) permet de diminuer la valeur de l'objectif de quelques points.

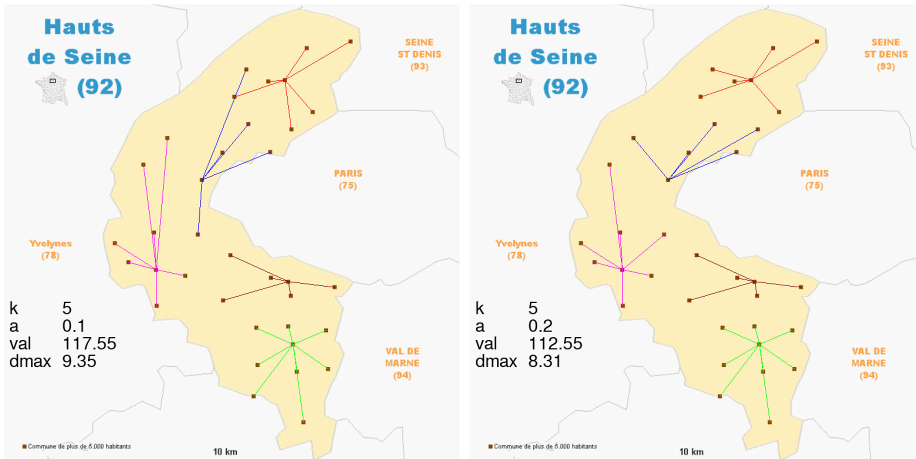

Figure 3: résultats obtenus avec k=5 pour des localisations de ressources fixées.

A la seule vue du graphique de droite, on remarque qu'il suffit de changer la localisation de la ressource du secteur bleu pour réduire la valeur de l'objectif.

3 Minimisation de la plus grande distance

Le deuxième problème diffère du premier seulement au niveau de la fonction objectif, on cherche ici à minimiser la plus grande distance :

$$g(X) = \max_i \sum_j D_{ij} X_{ij} \quad + \epsilon f(X)$$

Le premier terme de la somme représente la distance pour le maire le moins bien servi. Avec epsilon suffisamment petit, le deuxième terme permet, pour deux solutions ayant la même évaluation de préférer celle qui minimise la moyenne des distances.

Pour représenter le max, on ajoute une variable d_{max} réelle et les contraintes :

$$\forall i \;\; d_{max} \geq \sum_j D_{ij} X_{ij}$$

Le prix de l'équité est défini par :

$$PE = 1 - \frac{f(x_f^*)}{f(x_g^*)}$$

Par définition de x_f^*, $f(x_g^*) \leq f(x_f^*)$. Le prix de l'équité est donc bien positif.

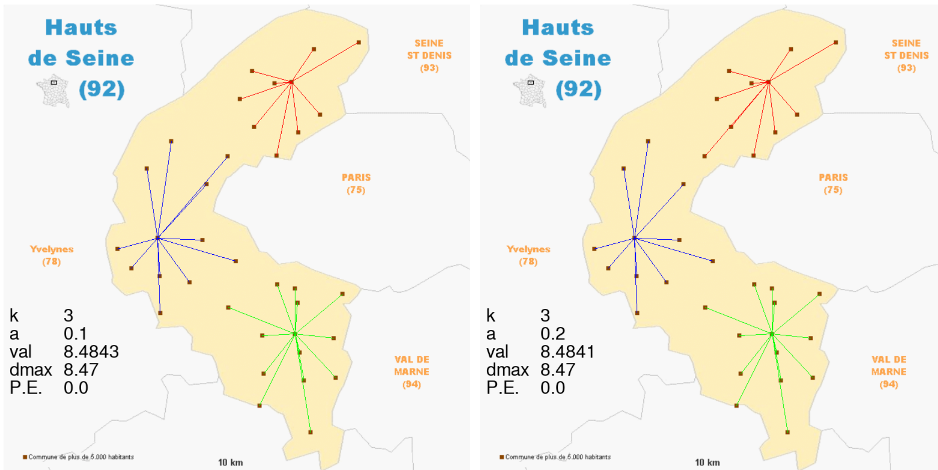

Figure 4: résultats obtenus avec k=3 pour des localisations de ressources fixées.
On peut ici observer la répartition optimale des villes du 92 en 3 groupes afin de minimiser la distance maximale séparant une ressource d'une ville.

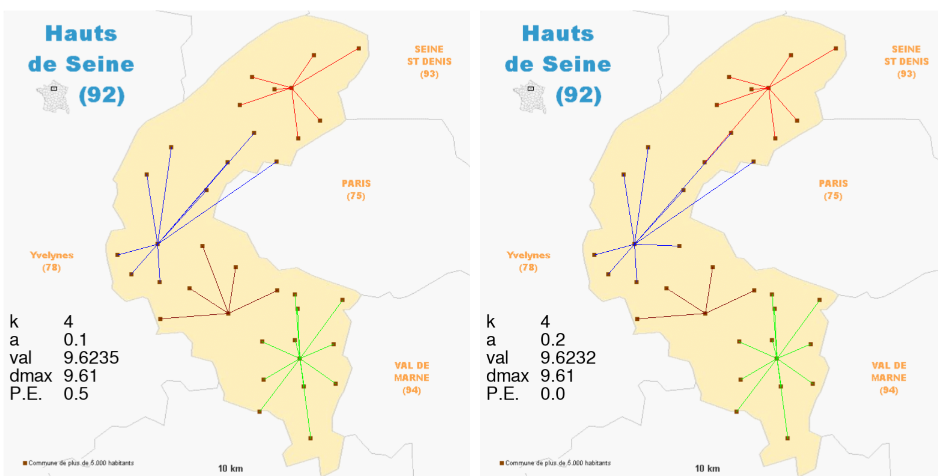

Figure 5: résultats obtenus avec k=4 pour des localisations de ressources fixées.
Atténuer la contrainte sur α permet ici de diminuer le prix de l'équité. Sur cette instance, les optimums selon f et g deviennent alors égaux.

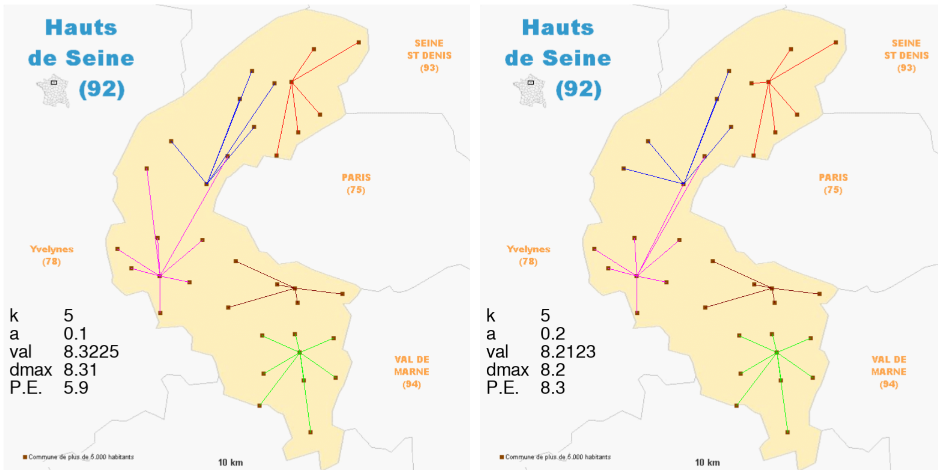

Figure 6: résultats obtenus avec k=5 pour des localisations de ressources fixées

Dans ce cas l'affaiblissement de la contrainte entraîne d'un coté l'amélioration de la valeur de l'objectif mais d'un autre coté l'augmentation du prix d'équité.

4 Détermination des localisations optimales

Dans le troisième problème, la localisation des ressources n'est pas fixée. X est donc ici une matrice $n \times n$ où X_{ij} vaut 1 si la ville i s'approvisionne dans la ville j. Une ville j est un point d'accès si il y a au moins un coefficient à 1 dans la colonne j, on représente cela par des variables binaires égales au maximum de la colonne :

$$\forall j \;\; isSector_j = \max_i X_{ij}$$

La présence d'exactement k point d'accès est représentée par la contrainte :

$$\sum_j isSector_j = k$$

On conserve aussi les contraintes et la fonction objectif du problème précédent.

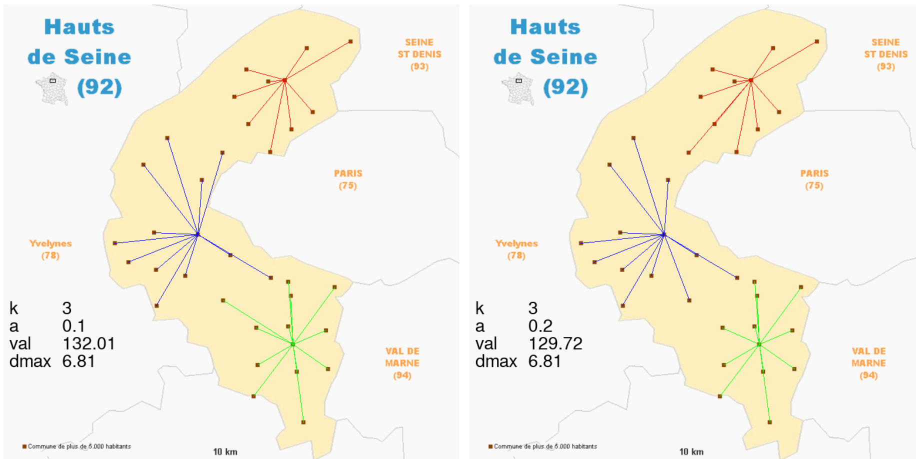

Figure 7: minimisation de la moyenne avec k=3 pour des localisations de ressources libres.

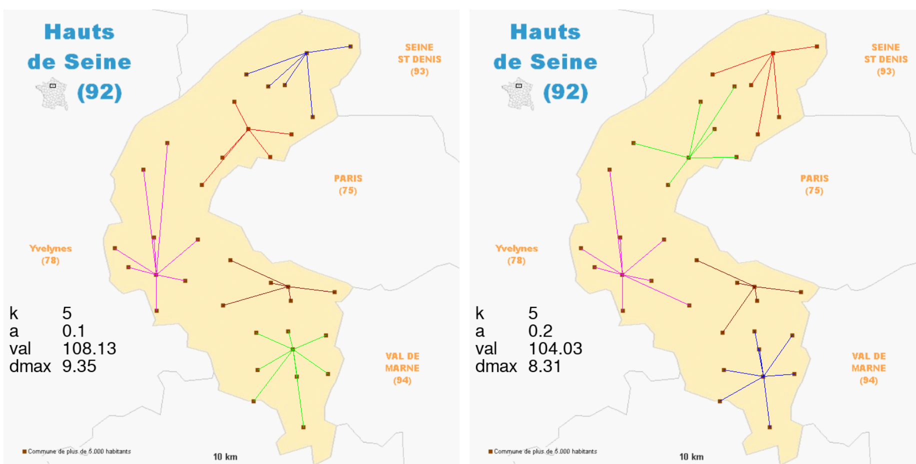

Figure 8: minimisation de la moyenne avec k=5 pour des localisations de ressources libres.

On remarque qu'atténuer la contrainte a provoqué un changement de localisation de ressource pour 2 secteurs.

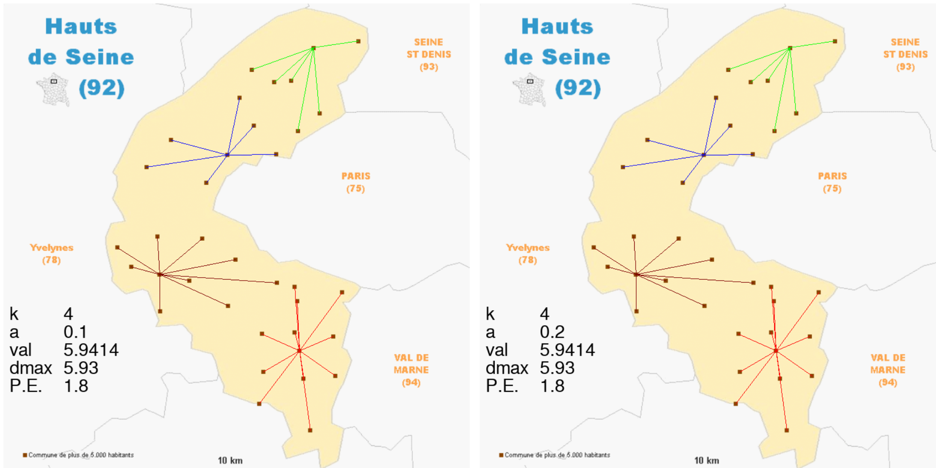

Figure 9: minimisation de la distance maximale avec k=4 pour des localisations de ressources libres.
Ici, atténuer la contrainte n'a aucun impact.

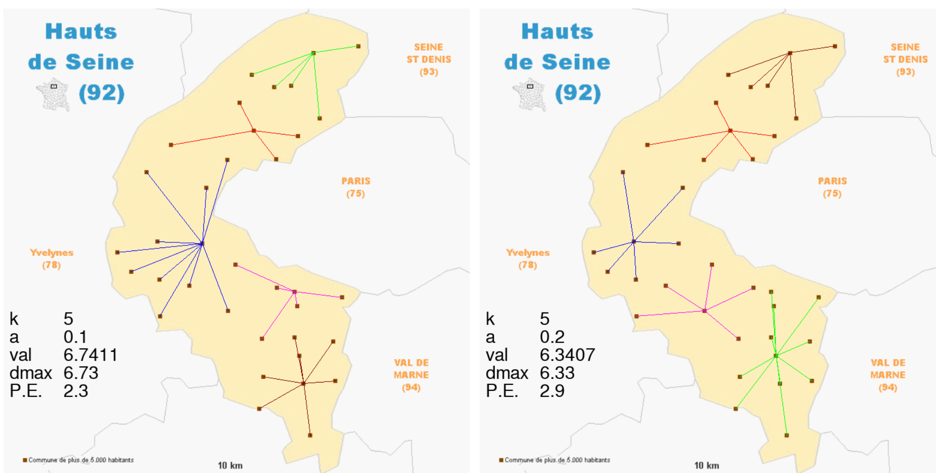

Figure 10: minimisation de la distance maximale avec k=5 pour des localisations de ressources libres.
Ici, diminuer la contrainte réduit à la fois la distance maximale et la distance moyenne mais le prix de l'équité augmente néanmoins.

5 Conclusion

La modélisation du problème en programme linéaire a pu se faire de manière relativement simple, et le nombre de variables et de contraintes obtenues est en $O(n^2)$. Quels que soient les paramètres, le solveur résout le problème en un temps négligeable. La modélisation du problème en un **problème linéaire** se révèle donc **très opportune** là où un algorithme classique n'aurait pas été utilisable.