

SDP - CentraleSupélec

PFIZER PROJECT

Optimiser l'assignation des commerciaux à des territoires

janvier 2025



Maceo Duriez

maceo.duriez@student-cs.fr

Gaétan Jacquemin

gaetan.jacquemin@student-cs.fr

Ilann Amiaud-Plachy

ilann.amiaudplachy@student-cs.fr

1 Introduction

Dans le cadre de ce projet, nous nous sommes intéressés à la problématique de l'assignation optimale des territoires pour les représentants commerciaux de Pfizer en Turquie. L'évolution des besoins du marché pharmaceutique et la nécessité d'une gestion plus efficace des ressources ont conduit à une réévaluation de la répartition des zones géographiques. Le projet vise à appliquer les méthodes du cours pour répondre à plusieurs objectifs : la minimisation des distances parcourues, l'équilibre des charges de travail entre les représentants, ainsi que la limitation des perturbations causées par les changements de territoires.

Ce rapport présente les différentes étapes de notre démarche, depuis la modélisation mathématique jusqu'à l'optimisation d'abord mono-objectif avec Gurobi, puis multi-objectif par la méthode de contrainte "epsilon". Et finalement la méthode UTA pour l'apprentissage des préférences et le choix de la solution finale parmi l'ensemble de solutions non-dominées données par la méthode epsilon.

Contents

1	Introduction	1
2	Étape 1 : Modélisation du Problème d'Assignation	2
2.1	Formulation des modèles mono-objectifs	2
2.2	Résolution avec GUROBI	3
2.3	Exploration des solutions non-dominées	3
3	Étape 2 : Extension du Modèle d'Assignation	6
3.1	Résolution des instances 100 briques / 10 SRs	6
3.1.1	Modèle utilisé	6
3.1.2	Résultats obtenus	6
3.2	Modélisation de l'affectation partielle des briques	7
3.2.1	Modèle modifié	7
3.2.2	Implémentation et comparaison des résultats	8
3.3	Impact de l'augmentation de la demande	9
3.3.1	Résultats de l'analyse	10
3.4	Conclusion	10
4	Étape 3 : Optimisation de la Localisation des Bureaux	11
4.1	Modélisation bi-objective	11
4.2	Nouvelle disruption, nouveaux objectifs	12
5	Étape 4 : apprentissage des préférences	13
5.1	Modélisation du score	13
5.2	Optimisation du score	13
5.3	Résultats	14

2 Étape 1 : Modélisation du Problème d'Assignment

Le travail se fera pour 4 SR et 22 territoires à attribuer pour nous verrons si notre modèle tient à l'échelle pour 10 SR et 100 zones dans l'étape 2.

2.1 Formulation des modèles mono-objectifs

Dans un premier temps, deux modèles mathématiques sont formulés pour résoudre le problème d'assignation des représentants de vente aux territoires. Le premier modèle vise à minimiser la distance totale parcourue par les représentants, tandis que le second cherche à minimiser la perturbation causée par une nouvelle répartition des zones.

Soit :

- x_{ij} une variable binaire indiquant si la zone i est assignée au représentant j .
- d_{ij} la distance entre le centre du territoire du représentant j et la zone i .
- p_i la charge de travail nouvelle associée au changement de l'affectation de la zone i .

L'objectif de minimisation de la distance est donné par :

$$\min \sum_{i,j} d_{ij} x_{ij} \quad (1)$$

Cet objectif permet de minimiser la distance parcourue par l'ensemble des représentants, sans prendre en compte si la distance parcourue est équilibrée.

L'objectif de minimisation de la perturbation est donné par :

$$\min \sum_i p_i x_{ij} \quad (2)$$

Nous avons de même essayé un second objectif de minimisation pour la distance :

$$\min \max \sum_{i,j} d_{ij} x_{ij} \quad (3)$$

Cet objectif permet de rendre la répartition plus équitable, car on minimise la distance parcourue par le représentant qui en parcourt le maximum. Chaque représentant aura alors un travail plus équitable.

Sous contraintes :

- Chaque zone est assignée à exactement un représentant.
- La charge de travail de chaque représentant est équilibrée (c'est à dire, inférieure à 1.2 et supérieure à 0.8).
- Les représentants ne peuvent pas être déplacés

2.2 Résolution avec GUROBI

Les modèles formulés sont implémentés et résolus avec l'optimiseur GUROBI. L'instance test comprend 22 zones et 4 représentants de vente.

Les étapes de l'implémentation incluent :

- Définition des variables de décision.
- Définition des contraintes du modèle.
- Résolution et analyse des solutions obtenues.

La solution pour la minimisation mono-objective de la nouvelle charge de travail (disruption) donne une valeur de 0.170.

La solution optimale du mono-objectif de minimisation de la distance totale donne une valeur de 56.

Pour pouvoir transformer l'optimisation non linéaire du Min(Max), il suffit seulement de rajouter une variable z qui est le maximum de toutes les distances parcourues par les représentants. On peut alors ajouter une contrainte pour que z soit supérieur à toutes les distances parcourues par les représentants.

2.3 Exploration des solutions non-dominées

Une approche par contrainte-épsilon est utilisée pour générer l'ensemble des solutions non-dominées. L'idée est d'imposer des contraintes successives sur l'un des objectifs (la distance totale) tout en minimisant l'autre (disruption).

L'analyse est réalisée avec trois niveaux de contraintes sur la charge de travail :

- 1. $[0.8, 1.2]$
- 2. $[0.85, 1.15]$
- 3. $[0.9, 1.1]$

Chaque ensemble de solutions est visualisé à l'aide de graphes comparant la distance et la perturbation ci-dessous :

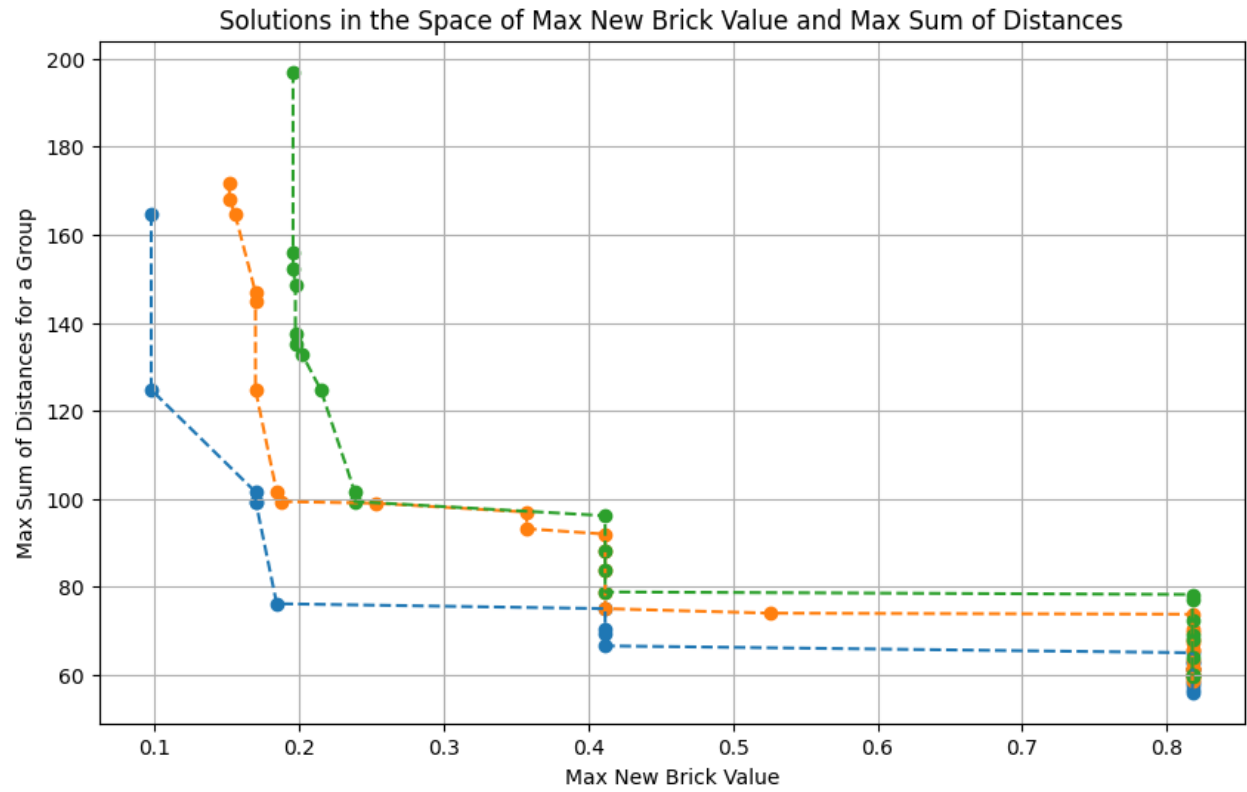


Figure 1: Solutions non-dominées avec l'objectif de distance Min pour des charges de travail selon 1. en bleu, 2. en orange et 3. en vert

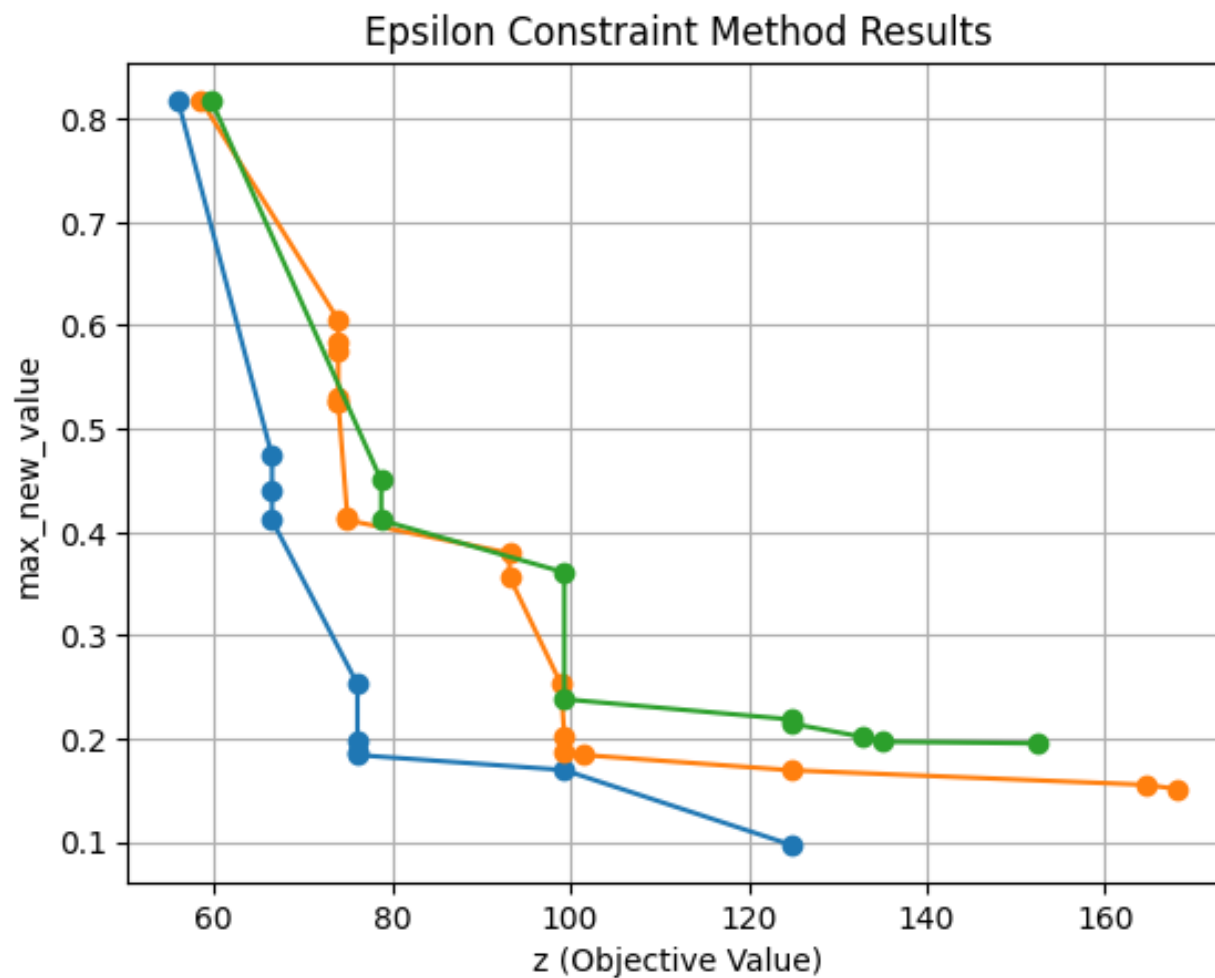


Figure 2: Solutions non-dominées avec l'objectif de distance MinMax pour des charges de travail selon 1. en bleu, 2. en orange et 3. en vert

3 Étape 2 : Extension du Modèle d'Assignment

Dans cette deuxième étape, nous explorons des extensions du modèle développé dans l'étape 1. Nous cherchons à tester la capacité du modèle initial à résoudre des instances plus complexes et à explorer des configurations supplémentaires, notamment l'affectation partielle des briques et l'impact de l'augmentation de la demande, caractérisé par la création d'un nouveau poste de représentant. Sur les graphiques suivants, z est la somme des distances parcourues par chacun.

3.1 Résolution des instances 100 briques / 10 SRs

Nous commençons par analyser la capacité du modèle initial à résoudre des instances où 100 briques doivent être assignées à 10 représentants de vente (SRs). L'objectif est de vérifier si notre modèle est capable de traiter cette augmentation de la taille du problème et de produire des solutions efficaces. On doit alors pré-processer les données pour obtenir les distances entre les briques et les SRs dans le même format que l'étape 1.

3.1.1 Modèle utilisé

Le modèle utilisé pour cette étape est une extension du modèle multi-objectif avec epsilon-constraint présenté dans l'étape 1, avec des variables x_{ij} représentant l'assignation des briques aux SRs et des distances d_{ij} qui mesurent l'effort de chaque représentant en fonction des briques attribuées.

3.1.2 Résultats obtenus

Après résolution de l'instance avec 100 briques et 10 SRs, nous avons observé que le modèle a pu fournir une solution satisfaisante en un temps raisonnable, bien que le temps de calcul ait beaucoup augmenté par rapport à l'instance initiale de 22 zones et 4 SRs. Et à raison : il y a 1000 variables binaires alors qu'il n'y en avait que 88 dans l'instance précédente.

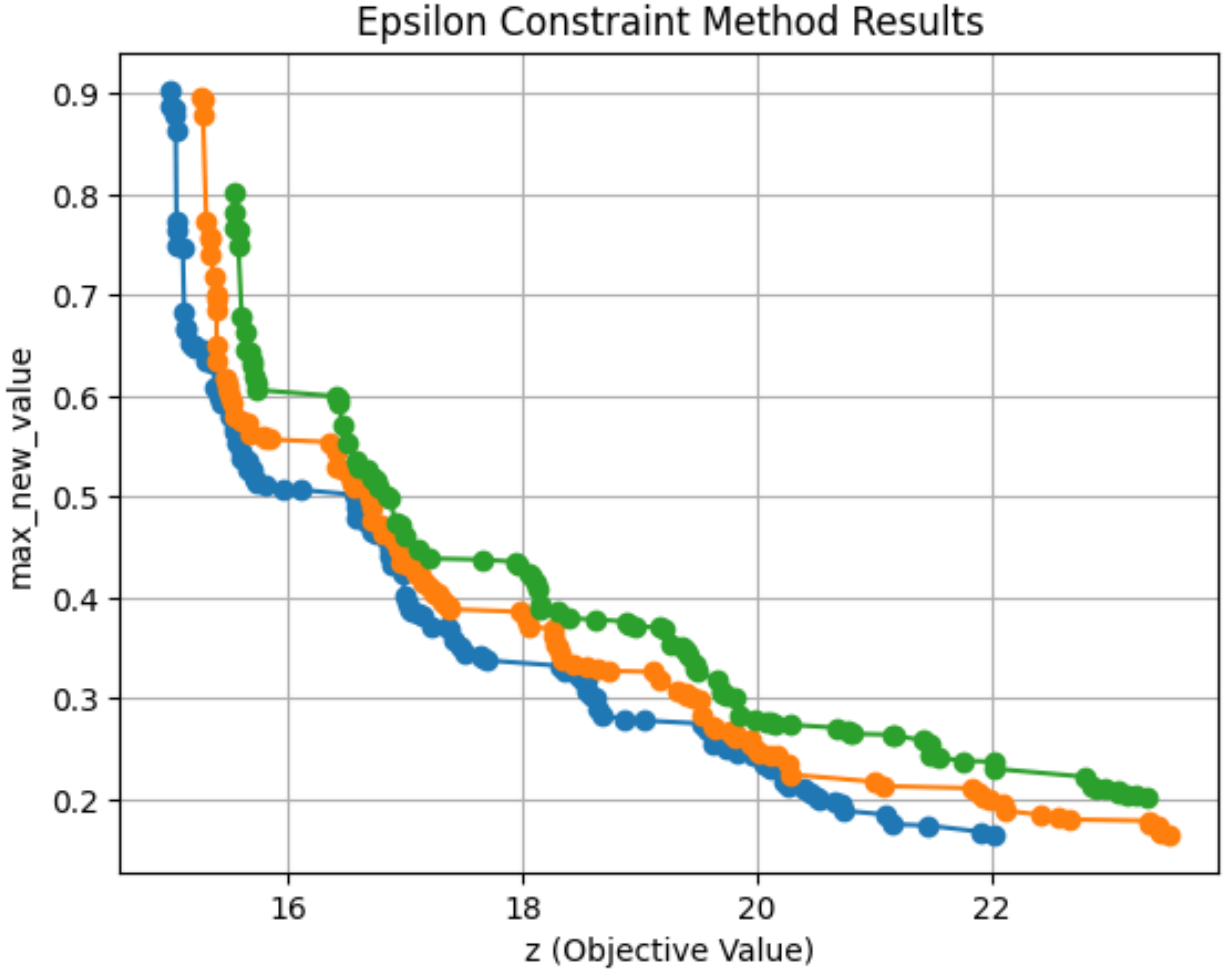


Figure 3: Solutions non-dominées pour 100 briques et 10 SRs

L'ensemble des solutions non-dominées (front de pareto) est représenté ci-dessus, où chaque point représente une solution obtenue pour différentes valeurs de la distance totale et de la charge de travail. On a tracé 3 courbes pour 3 niveaux de contraintes sur la charge de travail : $[0.8, 1.2]$, $[0.85, 1.15]$ et $[0.9, 1.1]$.

3.2 Modélisation de l'affectation partielle des briques

Une autre extension que nous avons étudiée est l'affectation partielle des briques, c'est-à-dire la possibilité d'assigner une brique à plusieurs SRs. Cette modélisation peut offrir plus de flexibilité et de solutions intéressantes, surtout dans le cas où certaines zones ont des exigences de demande plus élevées.

3.2.1 Modèle modifié

Pour prendre en compte cette affectation partielle, nous avons seulement changé le type des valeurs x_{ij} de binaire à continue entre 0 and 1 (non obligatoire, mais permet à notre modèle

de converger plus vite ayant moins de valeurs à aller chercher) L'objectif reste de minimiser la distance totale ainsi que la charge de travail maximale, mais cette fois-ci, les valeurs x_{ij} ne sont pas forcément égales à 0 ou 1, et permettent de répartir les zones de manière plus flexible.

3.2.2 Implémentation et comparaison des résultats

L'implémentation de cette nouvelle possibilité a été réalisée de manière similaire à celle du modèle précédent, en utilisant l'optimiseur GUROBI pour résoudre le problème. La comparaison des résultats obtenus avec et sans affectation partielle montre que l'affectation partielle permet de réduire la distance totale dans certains cas, mais augmente grandement la complexité du modèle. En effet, nous sommes passées de 1000 variables binaires (0 ou 1) à 1000 variables continues (nombre réel entre 0 et 1), ce qui augmente grandement le nombre de possibilités et donc le temps de calcul.

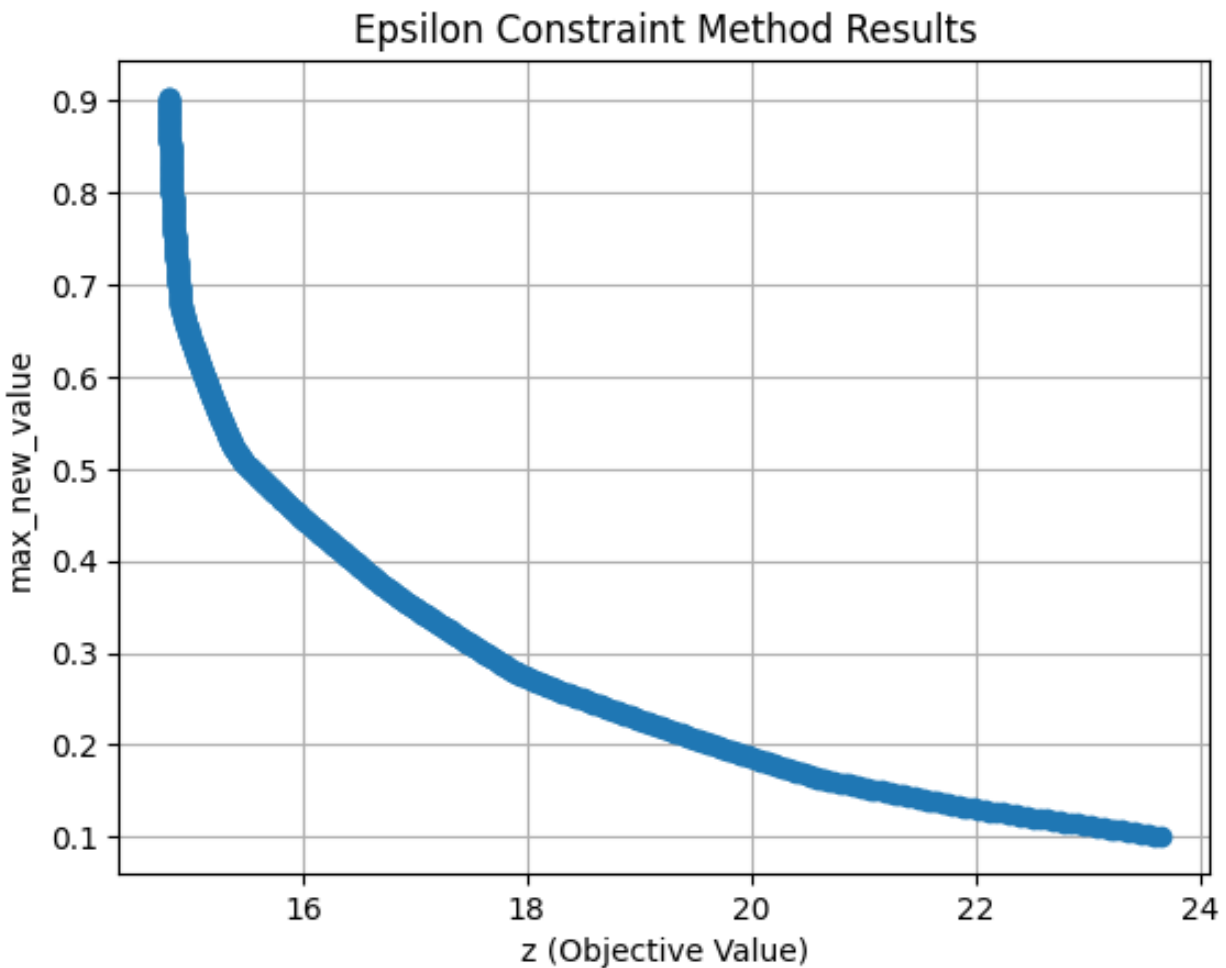


Figure 4: solutions avec affectation partielle

On peut voir que le front de pareto ne comporte plus de sauts et est plus lisse, ce qui

signifie que l'affectation partielle permet de trouver beaucoup plus de solutions, et est bien plus flexible.

3.3 Impact de l'augmentation de la demande

Supposons maintenant une augmentation uniforme de la demande dans toutes les briques, par exemple une augmentation de 25%. Dans ce scénario, il devient nécessaire d'ajouter un nouveau représentant de vente. La question se pose alors de savoir où localiser son bureau, c'est-à-dire, quel centre de brique choisir.

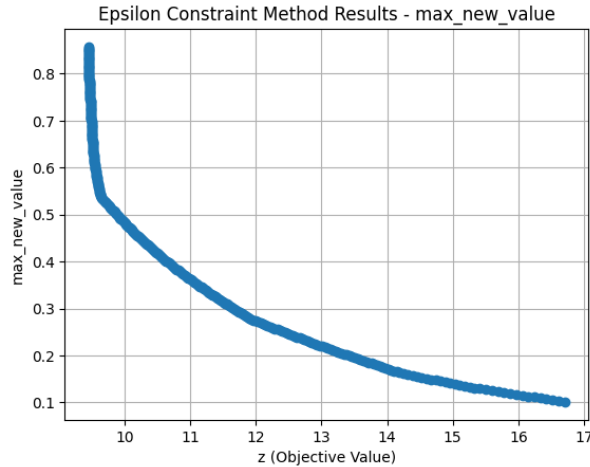
Nous devons construire un modèle permettant de déterminer l'emplacement optimal du bureau du nouveau représentant, en tenant compte de la minimisation de la distance totale parcourue par l'ensemble des représentants. Pour ce faire, nous avons modifié le modèle initial en ajoutant une nouvelle variable y_j qui indique si le bureau du nouveau représentant est situé dans la brique j . Il faudra alors rajouter et modifier des contraintes, ainsi que rajouter une colonne à x_{ij} pour prendre en compte cette nouvelle variable.:

- Le bureau du nouveau représentant doit être situé dans une seule brique.

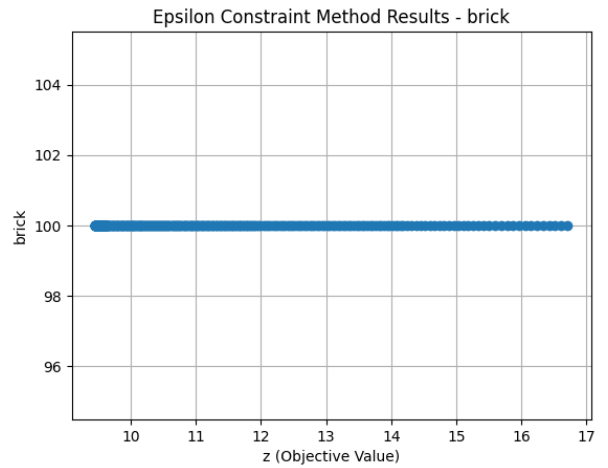
$$\sum_{j=1}^{100} y_j = 1 \quad (4)$$

$$\min \sum_{i,j}^{m,n} d_{ij} x_{ij} + \sum_{jk}^{n,n} D_{kj} y_j x_{k,n+1} \quad (5)$$

avec D_{ij} la distance entre le centre de la brique j et le bureau du nouveau représentant (on aura calculer chacune des possibilités en amont).



(a) Solutions avec affectation partielle et nouveau SR



(b) Numéro de la brique assigné au nouveau SR

3.3.1 Résultats de l'analyse

Après l'augmentation de la demande et l'ajout du nouveau représentant, le modèle a déterminé que le bureau du nouveau représentant devait être localisé à la 100 ième brique, quelque soit notre choix de solutions sur le front de pareto. Un nouveau représentant permet de baisser les valeurs de charges de travail maximales et de distance parcourue.

3.4 Conclusion

Dans cette étape, nous avons exploré plusieurs extensions du modèle initial, notamment la résolution d'instances plus grandes, l'affectation partielle des briques et l'impact de l'augmentation de la demande. Les résultats montrent que le modèle peut s'adapter à des problèmes de plus grande envergure, et que l'affectation partielle ainsi que l'ajout de nouveaux représentants peuvent contribuer à une meilleure répartition des charges de travail et à une minimisation de la distance totale, mais amène à une complexité de résolution grandissant énormément.

4 Étape 3 : Optimisation de la Localisation des Bureaux

4.1 Modélisation bi-objective

Dans cette étape, la localisation des bureaux des représentants de vente est prise en compte comme une variable de décision supplémentaire. L'objectif est de minimiser simultanément la distance totale parcourue et de répartir équitablement la charge de travail entre les représentants.

Soit :

- y_j la position du bureau du représentant j .
- c_{jk} la distance entre le nouveau bureau y_j et la zone k .

On a donc de nouveau un critère bi-objectif mais cette fois avec plus de données en entrée. Le premier objectif est la minimisation de la distance totale parcourue par les SR et le second objectif est la minimisation de la charge de travail maximale d'un SR. En résolvant avec Gurobi, on trouve cet espace de solutions :

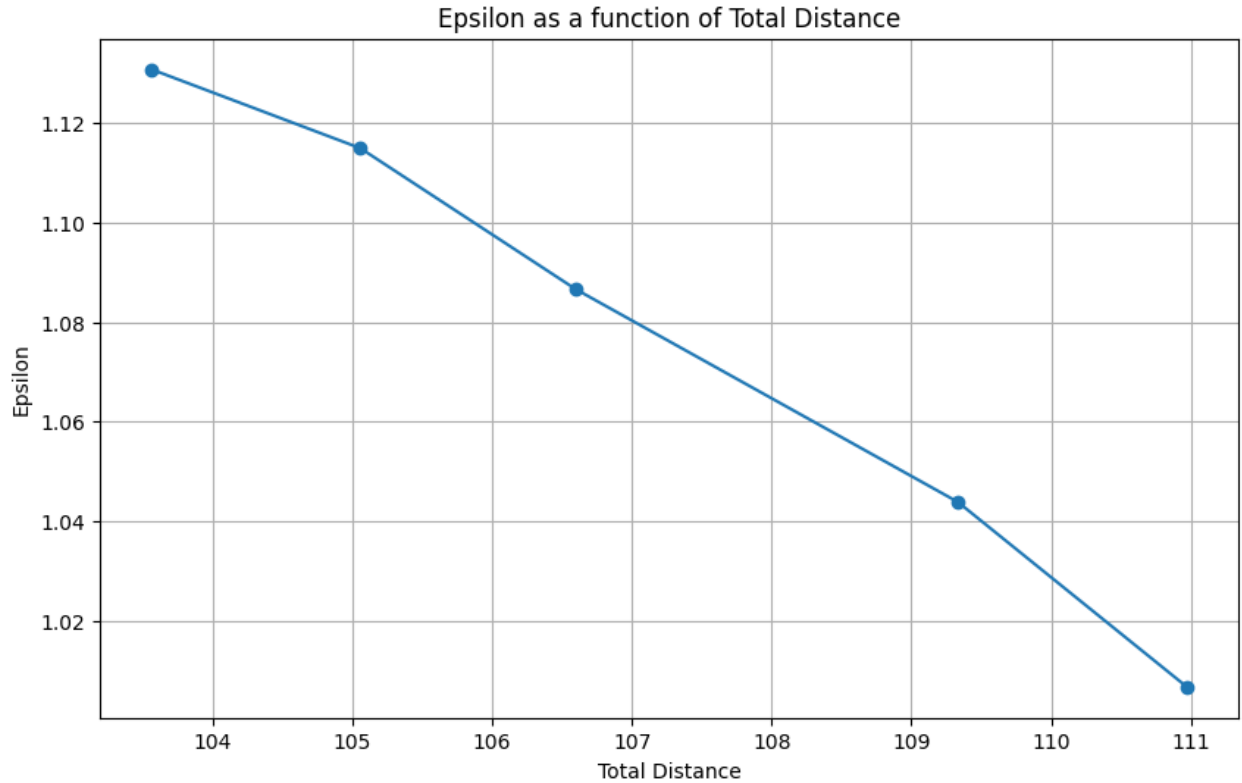


Figure 6: Solutions non-dominées du problème bi-objectif avec la distance totale en abscisses et le MinMax de charge de travail en ordonnées

4.2 Nouvelle disruption, nouveaux objectifs

Maintenant, la disruption est redéfinie en termes de nombre d'office déplacé. Ainsi, le problème initial de cette étape devient tri-objectif. Comme la disruption ne peut prendre que des valeurs entières allant de 0 à 4, nous représentons les solutions non-dominées comme les graphes de solutions non-dominées bi-objectives (distance et charge de travail), à nombre d'offices changés déterminée. Voici alors l'espace des solutions :

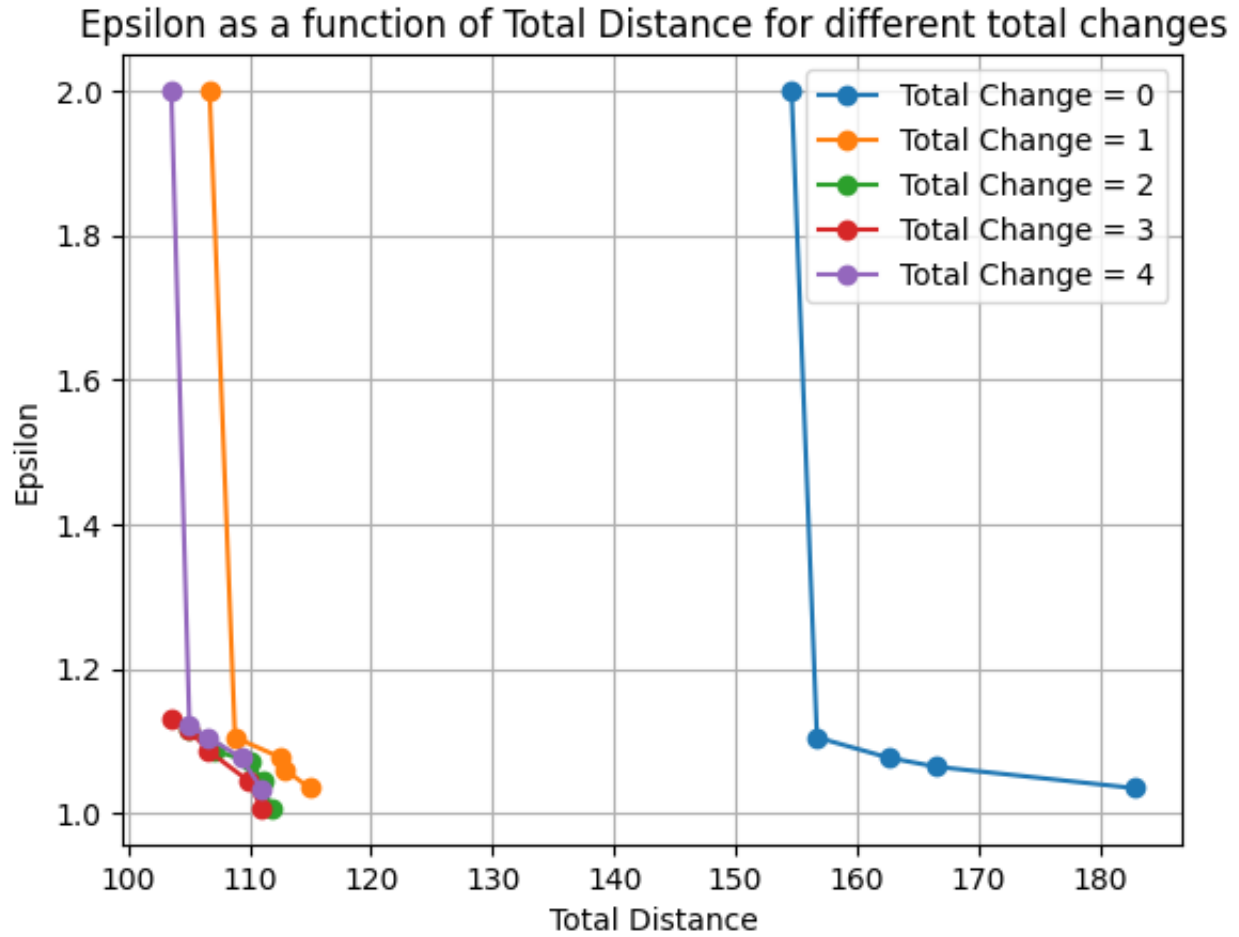


Figure 7: Solutions non-dominées du problème tri-objectif où epsilon représente le max de charge de travail pour un SR

5 Etape 4 : apprentissage des préférences

5.1 Modélisation du score

On modélise le score sur les 3 dimensions de la façon suivante (sous la forme UTA):

Le score est la somme des dimensions pondéré par un poids, le poids est une fonction linéaire par morceau sur chaque dimension, monotone, croissante et continue. qui vaut 0 pour la valeur minimale et 1 pour la valeur maximale. x étant un vecteur de taille 3, le score est donc donné par:

$$score(x) = \sum_{i=1}^3 \omega_i(x_i) \times x_i \quad (6)$$

avec en prenant ω_i^k la valeur de ω_i en x_i^k :

$$\omega_i(x_i) = \omega_i^k + (x_i - x_i^k) \times \frac{\omega_i^{k+1} - \omega_i^k}{x_i^{k+1} - x_i^k} \text{ avec } x_i^k \leq x_i \leq x_i^{k+1} \quad (7)$$

et :

$$x_i^k = min_i + k \frac{max_i - min_i}{L_i} \quad (8)$$

Les paramètres à déterminer sont donc les $(\omega_i^k)_{i=1,2,3; k \in [0, L_i]}$

Les valeurs minimales et maximales étant définis arbitrairement suivant les classements, on extrapole les fonction ω_i pour les valeurs inférieures et supérieures à x_i^0 et $x_i^{L_i}$ par les valeurs constante ω_i^0 et $\omega_i^{L_i}$.

5.2 Optimisation du score

à partir du classement effectué par l'agent, on approxime le score pour chaque SR classé de la façon suivante (on dispose de n SR classé avec leur scores):

$$H(x^t) = score(x^t) + \epsilon^t \quad (9)$$

avec ϵ^t une variable du problème d'optimisation.

On fixe un seuil $\delta = 1e^{-2}$ (pour avoir un ranking) et on ajoute les contraintes suivante:

$$\forall t \in [1, n - 1], H(x^t) \leq H(x^{t+1}) - \delta \quad (10)$$

et on minimise la somme des ϵ^t . la fonction objectif est donc:

$$\min\left(\sum_{t=1}^n \epsilon^t\right) \quad (11)$$

5.3 Résultats

On a donc utilisé la librairie gurobi pour résoudre ce problème d'optimisation.

On a utilisé les données de l'agent pour déterminer les paramètres ω_i^k . On a choisi de prendre $L_i = 5$ pour chaque dimension. On a donc 30 paramètres à déterminer. Les résultats sont données dans la figure 8.

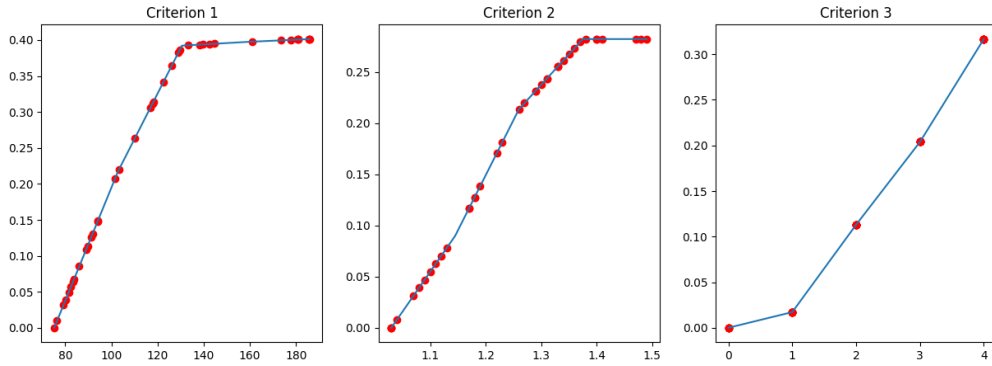


Figure 8: Valeurs des poids ω_i^k trouvés par optimisation

On a ensuite utilisé ces paramètres pour déterminer le classement des SR en comparant le score des solutions non-dominés. Les résultats sont donnés dans le tableau 1.

Index	Score	Distance	Max workload	Changed offices
1	0.314	114.94	1.03	1
2	0.322	112.92	1.06	1
3	0.331	108.81	1.1	1
4	0.333	112.62	1.08	1
5	0.387	111.79	1.01	2

Table 1: Les 5 premières solutions pour l'organisation des SR.