

Projet IA 2024/2025

Safeband

Compte rendu d'avancement

Décembre 2024



Maceo Duriez

maceo.duriez@student-cs.fr

Gaétan Jacquemin

gaetan.jacquemin@student-cs.fr

Ilann Amiaud-Plachy

ilann.amiaudplachy@student-cs.fr

Contents

1	Étape 1 : Modélisation du Problème d'Assignment	2
1.1	Formulation des modèles mono-objectifs	2
1.2	Résolution avec GUROBI	2
1.3	Exploration des solutions non-dominées	3
2	Étape 3 : Optimisation de la Localisation des Bureaux	5
2.1	Modélisation bi-objective	5
2.2	Nouvelle disruption, nouveaux objectifs	6
3	Etape 4 : apprentissage des préférences	6
3.1	Modélisation du score	6
3.2	Optimisation du score	7
3.3	Résultats	8

1 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.

1.1 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 :

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

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

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

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

1.2 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.

1.3 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 :

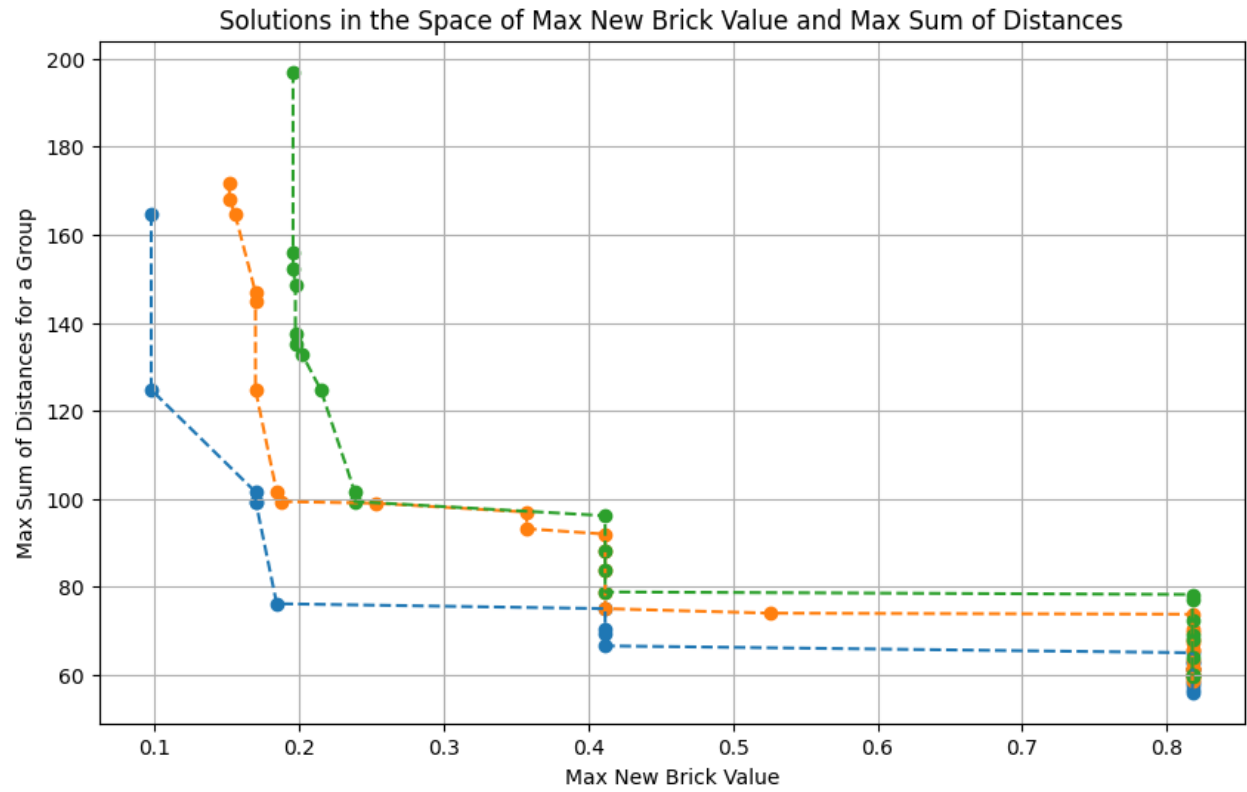


Figure 1: Solutions non-dominées pour des charges de travail selon 1. en bleu, 2. en orange et 3. en vert

2 Étape 3 : Optimisation de la Localisation des Bureaux

2.1 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 :

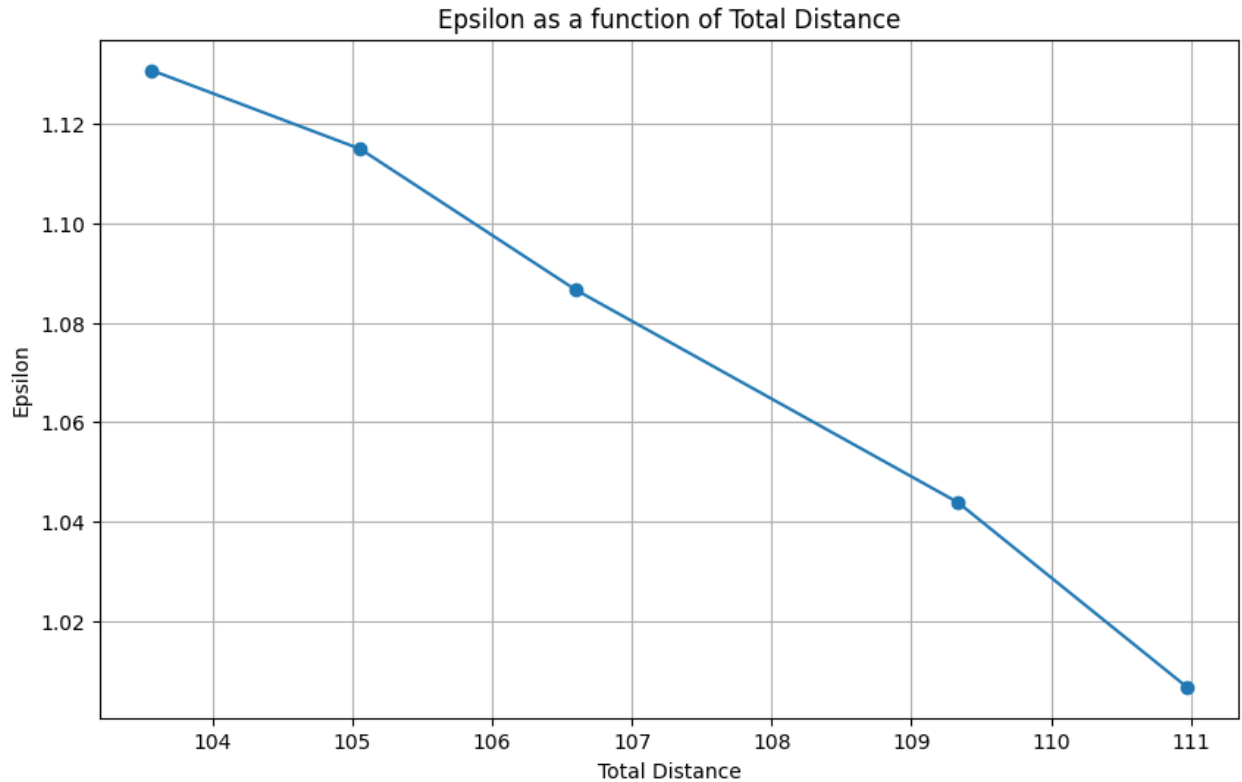


Figure 2: Solutions non-dominées du problème bi-objectif avec la distance totale en abscisses et le MinMax de charge de travail en ordonnées

2.2 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-objective (distance et charge de travail), à nombre d'offices changés déterminée. Voici alors l'espace des solutions :

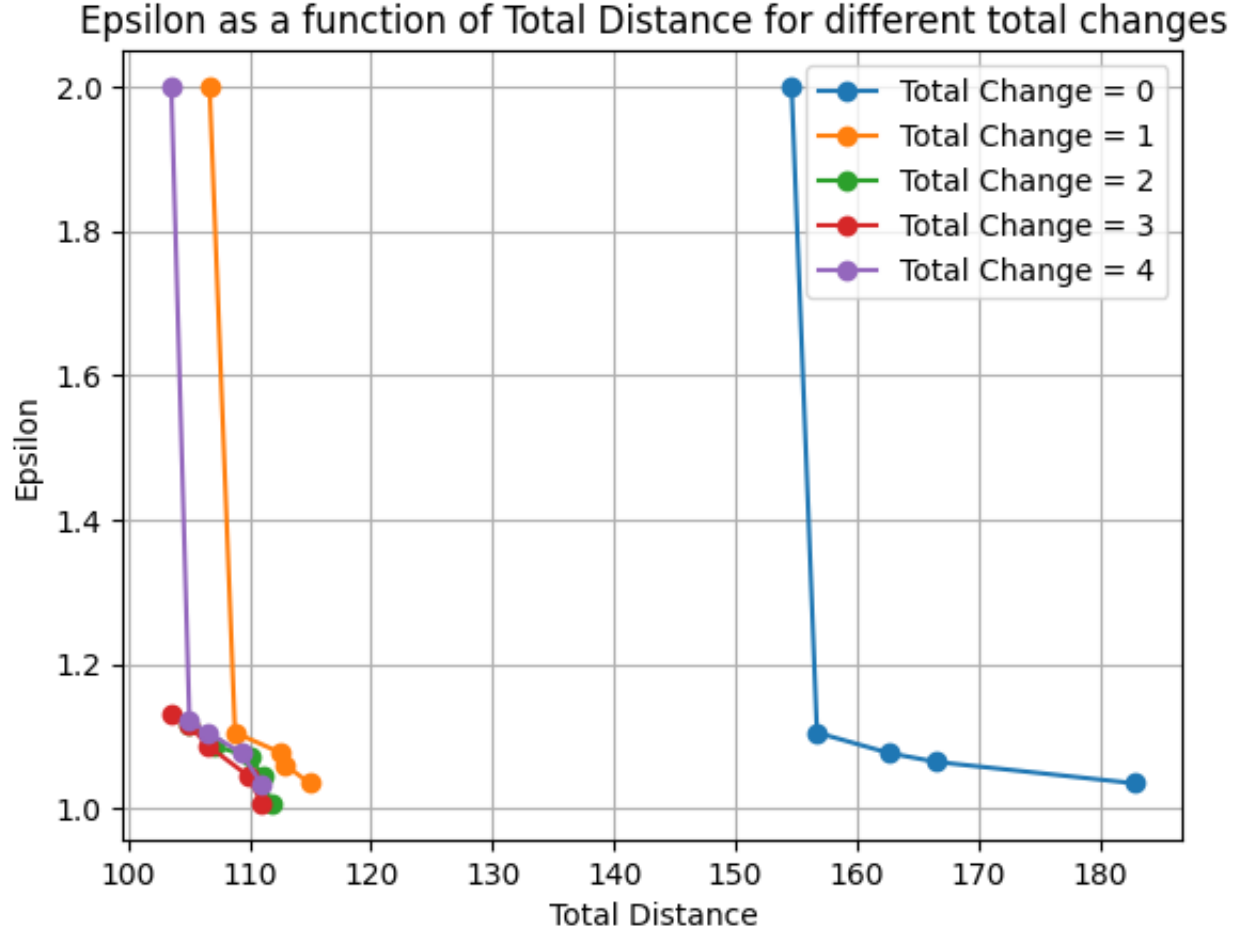


Figure 3: Solutions non-dominées du problème tri-objectif où epsilon représente le max de charge de travail pour un SR

3 Etape 4 : apprentissage des préférences

3.1 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:

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

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

$$\omega_i(x_i) = \omega_i^k + (x_i - x_i^k) \times \frac{\omega_i^{k+1} - \omega_i^k}{x_i^{k+1} - x_i^k} \text{ avec } x_i^k \leq x_i \leq x_i^{k+1} \quad (4)$$

et :

$$x_i^k = min_i + k \frac{max_i - min_i}{L_i} \quad (5)$$

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 $\omega_i^{L_i}$.

3.2 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):

$$H(x^t) = score(x^t) + \epsilon^t \quad (6)$$

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:

$$\forall t \in [1, n-1], H(x^t) \leq H(x^{t+1}) - \delta \quad (7)$$

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

$$\min \left(\sum_{t=1}^n \epsilon^t \right) \quad (8)$$

3.3 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 4.

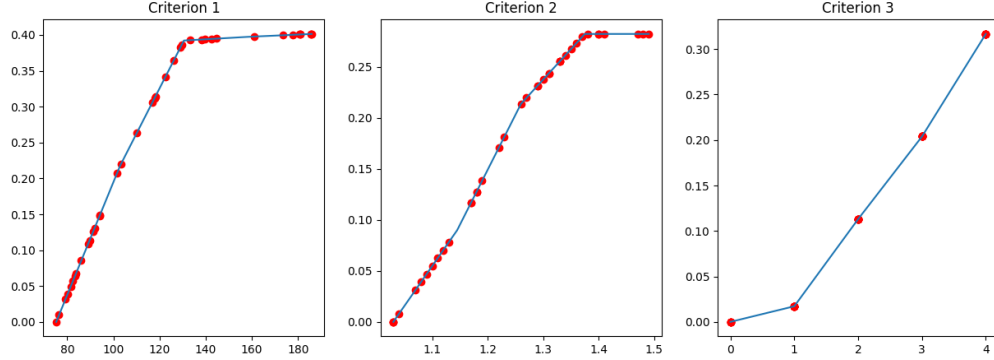


Figure 4: Valeurs des poids ω_i^k trouvés par optimisation

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.