Étude de couvertures de réseaux de métro, application de l'homologie persistante et proposition d'optimisation de l'algorithme.

Elowan ; 10381

June 8, 2025

Trouver les zones les moins biens desservies par un réseau de métro

Les principales étapes :

- *Convertir* les données géographiques en espace métrique ;
- Créer une suite de formes géométriques à partir de cet espace ;
- Trouver les trous de ces formes.

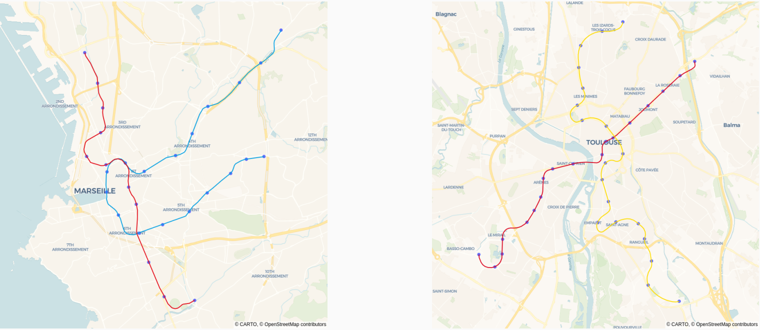


Figure 1: Lignes de métros (Ville A et Ville B)

Plus en détail, reconnaitre un trou : l'idée du lasso

Chercher à enrouler au lasso, puis reduire ces lassos sans jamais le rompre ou déchirer la forme.

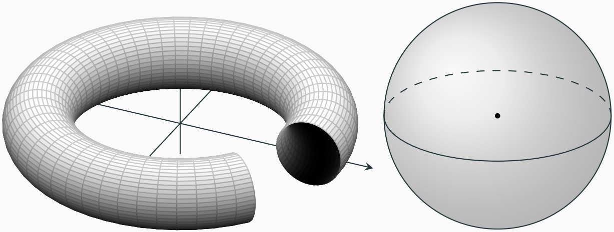


Figure 2: Un tore contenant deux trous et une sphère contenant 0 trou.

Définitions

Simplexe
Un simplexe σ de dimension k (ou k-simplexe) correspond à l'enveloppe convexe de $k+1$ vecteurs non inclus dans un sous-espace affine de dimension $k-1$.

Face
On dit que σ_i est une face de σ_j si et seulement si $\sigma_i \subset \sigma_j$ et la dimension de σ_i $\dim(\sigma_i)$ est égale à $\dim(\sigma_j) - 1$.

Complexe simplicial
Un ensemble de simplexes.

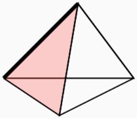


Figure 3: Exemple de complexe simplicial

Filtration
Suite croissante pour l'inclusion de complexes simpliciaux.

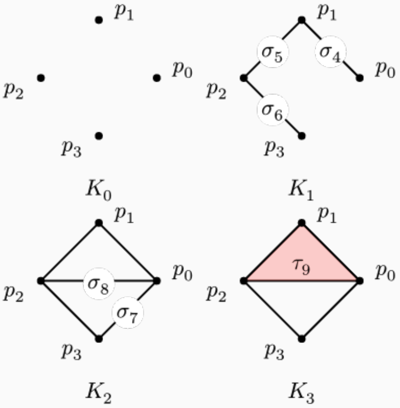


Figure 4: Exemple de filtration.

Classe d'homologie

Elle représente un trou en dimension *n*.

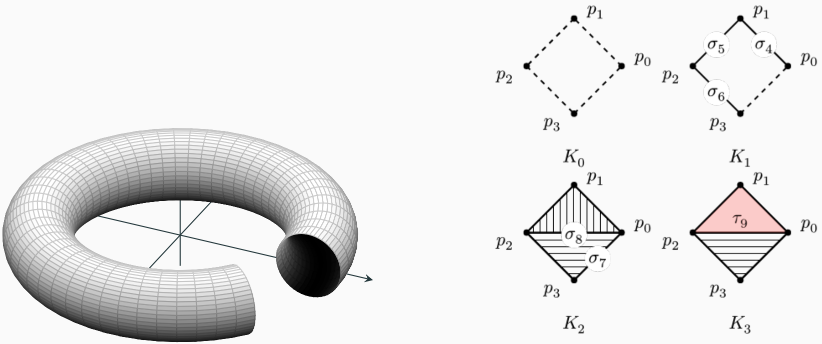


Figure 5: Le tore avec ces deux classes d'homologies et la filtration d'exemple avec en pointillé les classes d'homologies de dimension 0 et en hachuré celles de dimension 1.

- Construire une filtration à partir d’un ensemble de points ;
- Application de l’algorithme *standard* ;
- Récupération des classes d’homologies.

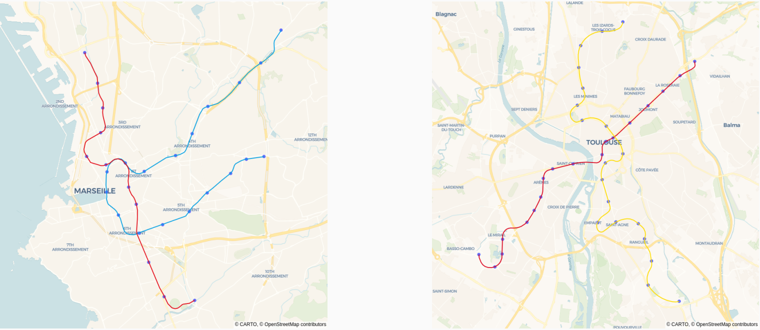


Figure 6: Lignes de métros (Ville A et Ville B)

Définition de la distance

Distance
On définit la distance d entre deux stations de metro x et y :

$$d(x, y) = \frac{1}{2}(min(t_{pied}(x, y), t_{voit}(x, y)) + min(t_{pied}(y, x), t_{voit}(y, x)))$$

Poids d'une station
La moyenne sur une semaine du temps d'attente en station.

Définition des complexes pondérés de Vietoris-Rips

Complexe Simplicial pondéré de Vietoris-Rips
On le définit au rang t, comme l'ensemble des simplexes $(\sigma_{i_0}, ..., \sigma_{i_k})$ sur les points tels que $\forall q \in [|0, k|], \sigma_q = [p_{i_0}, ..., p_{i_l}]$,

$$\begin{cases} \forall j \in [|0, l|], poids_{i_j} < t \\ \forall (j, k) \in [|0, l|]^2, d(p_{i_j}, p_{i_k}) + poids_{i_j} + poids_{i_k} < 2t \end{cases}$$

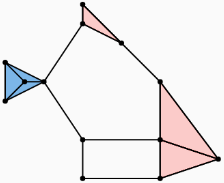


Figure 7: Création d'un complexe de Vietoris Rips au rang t, simplexes de dimension 0 et 1 en noir, dimension 2 en rouge et 3 en bleu.

Préparatif de l'algorithme (1/2): Ordre total sur les simplexes

Soient une filtration $K_0 \subset K_1 \subset ... \subset K_p$ et l'ensemble S de tous les simplexes apparaissant dans la filtration. On indice S de sorte que pour tout σ_i et σ_j de S:

$$\left.\begin{array}{l}\text{Si } \sigma_i \in K_{k_i} \text{ et } \sigma_j \in K_{k_j} \text{ avec } k_i < k_j \\ \text{Sinon si } \sigma_i \text{ est une face de } \sigma_j\end{array}\right\} \Rightarrow i < j$$

L'ordre total est déduit de cet indiçage.

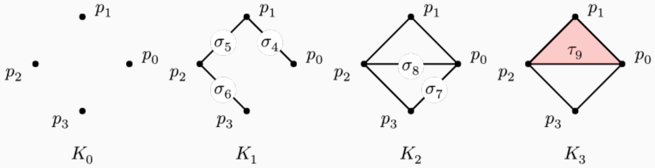
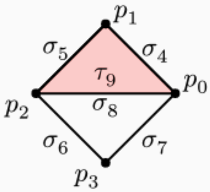


Figure 8: Indiçage de la filtration

Préparatif de l'algorithme (2/2): Matrice de bordure



	4	5	6	7	8	9
0	1			1	1	
1	1	1				
2		1	1		1	
3			1	1		
4						1
5						1
6						
7						
8						1
low	**1**	**2**	**3**	**3**	**2**	**8**

Figure 9: Exemple de matrice de bordure B associé à un ordre total

Application de l'algorithme

Entrée: $B \in M_n(\{0,1\})$ la matrice à reduire ($(C_i)_{i=1}^n$ ses colonnes)

Sortie : $\overline{B}$ matrice réduite

Pour *j=0 à n-1* **Faire**
    Tant que $\exists i < j$ *tel que low[j]* $=$ *low[i]* **Faire**
        $C_j \leftarrow (C_i + C_j) \bmod 2$
    Fin
Fin

Compréhension du résultat en sortie

	4	5	6	7	8	9
0	1			1	1	
1	1	1				
2		1	1		1	
3			1	1		
4						1
5						1
6						
7						
8						1
low	**1**	**2**	**3**	**3**	**2**	**8**

	4	5	6	7	8	9
0	1					
1	1	1				
2		1	1			
3			1			
4						1
5						1
6						
7						
8						1
low	**1**	**2**	**3**	**-1**	**-1**	**8**

Figure 10: Matrice B et réduite $\overline{B}$

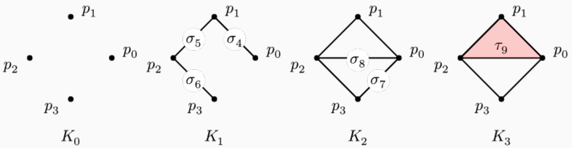


Figure 11: Filtration et utilisée

Optimisation

Remarques à l'initiative de la recherche d'optimisation :

- La matrice est creuse ;
- L'opération de somme de colonnes laisse invariante tous les simplexes de dimension différente.

Modifications apportées :

- Représentation de la matrice en liste d'adjacence (listes chaînées) ;
- Application de l'algorithme sur des matrices extraites et non la totale.

Algorithme optimisé

Entrée: $B \in M_n(\{0,1\})$ la matrice à reduire ($(C_i)_{i=1}^n$ ses colonnes)

Sortie : $\overline{B}$ matrice réduite

```
dims ← Tableau des simplexes où dims[i] contient la liste des simplexes de dimension i
Pour tout dimension d à considérer Faire
    Pour tout j ∈ dims[d] Faire
        Tant que ∃i < j dans dims[d] tel que low[j] = low[i] Faire
            C_j ← (C_i + C_j) mod 2
        Fin
    Fin
Fin
```

Résultats

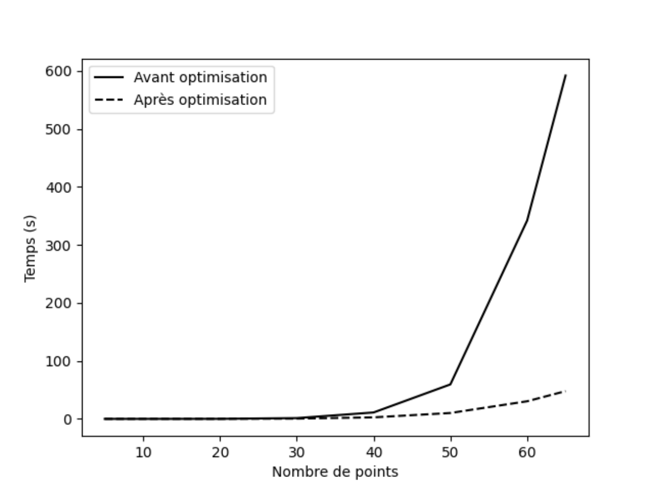


Figure 12: Ordonnée linéaire

Résultats

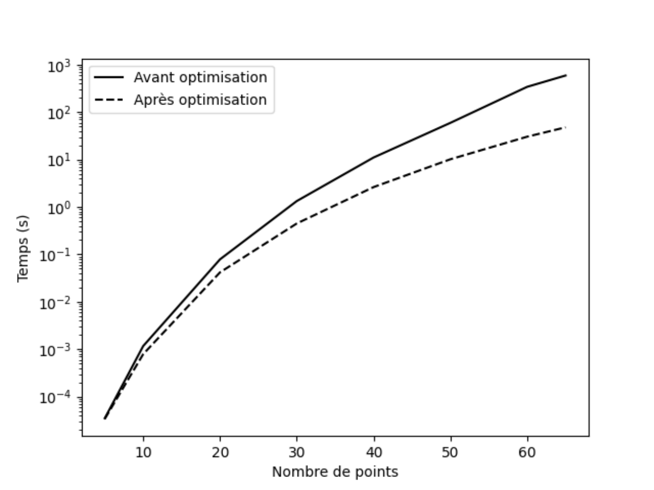


Figure 13: Ordonnée logarithmique

Résultats

Figure 14: Ville A

Figure 15: Ville B

Annexe : Définition

Complexe de chaînes
On définit un *complexe de chaînes* comme la donnée d'une suite

$$... \xrightarrow{\delta_{k+2}} C_{k+1} \xrightarrow{\delta_{k+1}} C_k \xrightarrow{\delta_k} C_{k-1} \xrightarrow{\delta_{k-1}} ...$$

Où chaque C_k est un groupe abélien libre qui a pour base les k-simplexes de X et δ_k est une morphisme de groupes tel que $\delta_k \circ \delta_{k+1} = 0$

On appelle δ_k un *opérateur de bords*.

Classes d'homologies H_k
On définit alors les *classes d'homologie de dimension k* comme le groupe de $Ker(\delta_k)$ quotienté par $Im(\delta_{k+1})$:

$$H_k = Ker(\delta_k)/Im(\delta_{k+1})$$

Annexe : Théorème des facteurs invariants

Il existe un unique ensemble $d_1, ..., d_p$ d'éléments de H_k définis à des inversibles près, tel que :

$$H_k \simeq \mathbb{Z}^\beta \bigoplus_{i=1}^{p} \mathbb{Z}/d_i\mathbb{Z}$$

Annexe : Diagrammes de persistance

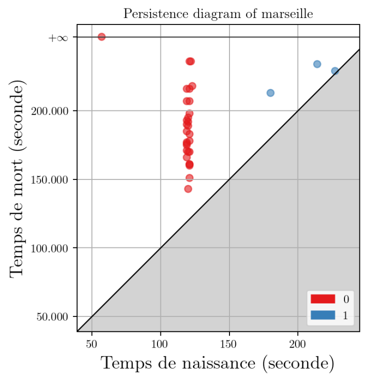


Figure 16: Diagramme de persistance de Marseille

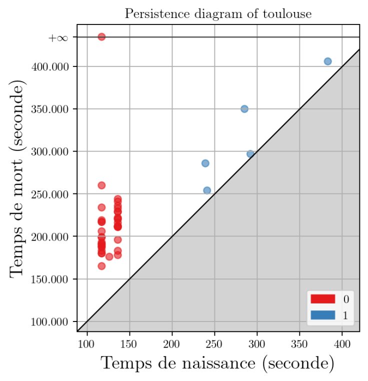


Figure 17: Diagramme de persistance de Toulouse

Annexe : Récupération des données

Pour le calcul des temps de trajet : apidocs.geoapify.com

Pour la récupération des stations et des temps d'attentes moyens : transport.data.gouv.fr

Annexe : Illustration preuve d'optimisation

	0	1	2	3	4	5	6	7	8	9
0					1			1	1	
1					1	1				
2						1	1		1	
3							1	1		
4										1
5										1
6										
7										
8										1
9										
low	-1	-1	-1	-1	**1**	**2**	**3**	**3**	**2**	**8**

Figure 18: Illustration des matrices extraites par dimension

Annexe : Preuve de correction 1/2

Soit $d \in [|0, |X| - 1|]$, considérons la matrice extraite :
$B_d = (B_{i,j})_{(i,j) \in I}$ telle que

$$I = \{(i,j) \in [|0, n-1|], dim(\sigma_i) = d \text{ et } dim(\sigma_j) = d+1\}$$

On note ϕ la correspondance entre les indices des deux matrices :
$(B_d)_{\phi(i,j)} = B_{i,j}$

Supposons que l'on exécute la ligne 3 de l'algorithme, alors $low(j) = low(i) = k$, on pose σ_i, σ_j et σ_k les simplexes associés. Donc σ_k est une face de σ_i et σ_j, donc par définition $dim(\sigma_k) + 1 = dim(\sigma_i) = dim(\sigma_j)$

Annexe : Preuve de correction 2/2

La ligne L_k ainsi que les deux colonnes C_i et C_j sont alors considérées dans B_d ($d = dim(\sigma_k)$). De plus, toutes les lignes ayant un coefficient non nul dans les colonnes C_i ou C_j le sont aussi puisqu'un coefficient non nul revient à être une face, donc de dimension d.

Ainsi l'opération de somme des colonnes $C_i + C_j$ dans B (ligne 3) est équivalent à celle de $C'_i + C'_j$ dans B_d avec $\phi(i, j) = (i', j')$.