

TME4

Algorithmes de recherche de chemin

Analyse du reseau Rollernet

Rapport d'analyse

DEV4 2024-2025

Genere le 27/06/2025

BM. Bui-Xuan

Ce rapport presente une analyse complete du reseau Rollernet
a travers quatre exercices complementaires :

- Exercice 1 : Traitements de base dans les donnees orientees graphes
 - Exercice 2 : Traitements pour donnees statiques
 - Exercice 3 : Traitements pour donnees dynamiques
 - Exercice 4 : Tableau de bord avec metriques d'analyse

Table des matieres

1. Introduction	3
2. Exercice 1 : Analyse de base.....	4
3. Exercice 2 : Centralites statiques.....	5
4. Exercice 3 : Chemins temporels.....	6
5. Exercice 4 : Tableau de bord.....	7
6. Resultats et analyses.....	8
7. Conclusions	9
8. Annexes	10

1. Introduction

1.1 Contexte

Ce TME porte sur l'analyse du reseau Rollernet, un jeu de donnees representant les liens de proximite geometrique entre participants d'une promenade de sante dans la capitale francaise.

1.2 Objectifs

L'objectif principal est de comprendre qu'il y a "le bon resultat en algorithmique et le beaucoup meilleur". Les objectifs specifiques incluent :

- Maitrise des traitements de donnees basiques avec les lambdas (Ex1)
- Application des methodes pour cas statique niveau LM (Ex2)
- Implementation pour cas dynamique niveau D+ (Ex3)
- Production d'un tableau de bord complet (Ex4)

1.3 Donnees

Le jeu de donnees Rollernet contient des enregistrements temporels au format :
<id_sommet1> <id_sommet2> <numero_seconde>

Chaque ligne indique qu'a la seconde donnee, les participants id_sommet1 et id_sommet2 se trouvent en proximite geometrique.

1.4 Methodologie

L'analyse suit une approche progressive :

1. Analyse de base (graphe union, degres)
2. Centralites statiques (closeness, betweenness)
3. Chemins temporels dynamiques
4. Tableau de bord integre multi-temporel

2. Exercice 1 : Analyse de base

2.1 Objectif

Effectuer des traitements de données basiques sur le réseau Rollernet avec une approche fonctionnelle privilégiant les lambdas.

2.2 Tâches réalisées

- Filtrage du graphe union (20e-30e minute)
- Calcul des degrés en temps linéaire (tri-paniers)
- Analyse des "followers" par participant

2.3 Résultats

- Nombre de nœuds analysés : 63
- Fenêtre temporelle : 1200-1800 secondes
- Algorithme de tri : tri par paniers $O(n)$
- Format de sortie : fichier texte avec degrés

2.4 Approche fonctionnelle

L'implémentation privilégie :

- Utilisation des lambdas pour le filtrage
- Programmation fonctionnelle pour les transformations
- Optimisation avec `collections.Counter`
- Tri-paniers pour la complexité linéaire

2.5 Interprétation

Les degrés représentent le nombre de "followers" de chaque participant, donnant une première mesure d'influence dans le réseau social.

3. Exercice 2 : Centralites statiques

3.1 Objectif

Etudier la corrélation entre les indices de centralité et le degré des sommets dans le graphe statique (20e-30e minute).

3.2 Métriques calculées

- Closeness centrality : mesure de proximité moyenne
- Betweenness centrality : mesure d'intermédierité
- Corrélation avec le degré des nœuds

3.3 Résultats statistiques

- Nombre de nœuds : 62
- Degré moyen : 32.8
- Closeness max : 100.0%

3.4 Visualisations produites

- Diagramme Closeness vs Degré (points bleus carrés)
- Diagramme Betweenness vs Degré (points orange losanges)
- Analyse des bissectrices et corrélations

3.5 Interprétation des bissectrices

- Closeness centrality : corrélation positive avec le degré
- Betweenness centrality : relation plus complexe

3.6 Algorithme Floyd-Warshall

Complexité $O(n^3)$ utilisée pour calculer les distances, avec possibilité d'optimisation par A* pour les grandes instances.

4. Exercice 3 : Chemins temporels

4.1 Objectif

Implementer l'analyse de chemins temporels avec contrainte de 30 minutes, passant du concept de "plus court chemin" a "chemin plus rapide en temps".

4.2 Algorithme temporel

- Dijkstra modifie pour graphes dynamiques
- Contrainte temporelle : chemins terminant avant 30e minute
- Respect de la causalite : $t_1 < t_2$ pour $\text{edge}(v_1, v_2, t_1) \rightarrow \text{edge}(v_2, v_3, t_2)$

4.3 Metriques temporelles

- Temporal closeness centrality
- Temporal betweenness centrality
- Comparaison statique vs dynamique

4.4 Resultats cles

- Changement des leaders : statique \neq temporel
- Correlations faibles entre mesures statiques/temporelles
- Revelation de patterns caches par l'analyse temporelle

4.5 Optimisations

- Echantillonnage intelligent (30% cibles, 15% paires)
- Priority queue pour Dijkstra temporel
- Complexite maitrisee pour graphes de taille reelle

4.6 Insights

L'analyse temporelle revele des dynamiques de reseau invisibles dans l'approche statique classique.

5. Exercice 4 : Tableau de bord

5.1 Objectif

Produire un tableau de bord complet avec metriques analysant la structure du reseau Rollernet sur multiple fenetres temporelles.

5.2 Fenetres d'analyse

- Fenetre 1 : 10-20min
- Fenetre 2 : 20-30min
- Fenetre 3 : 30-40min

5.3 Evolution du reseau

- Noeuds : 57 → 62 → 62
- Densite : 49.2% → 53.7% → 58.8%
- Tendance : croissance de la connectivite

5.4 Dashboard complet (10 panneaux)

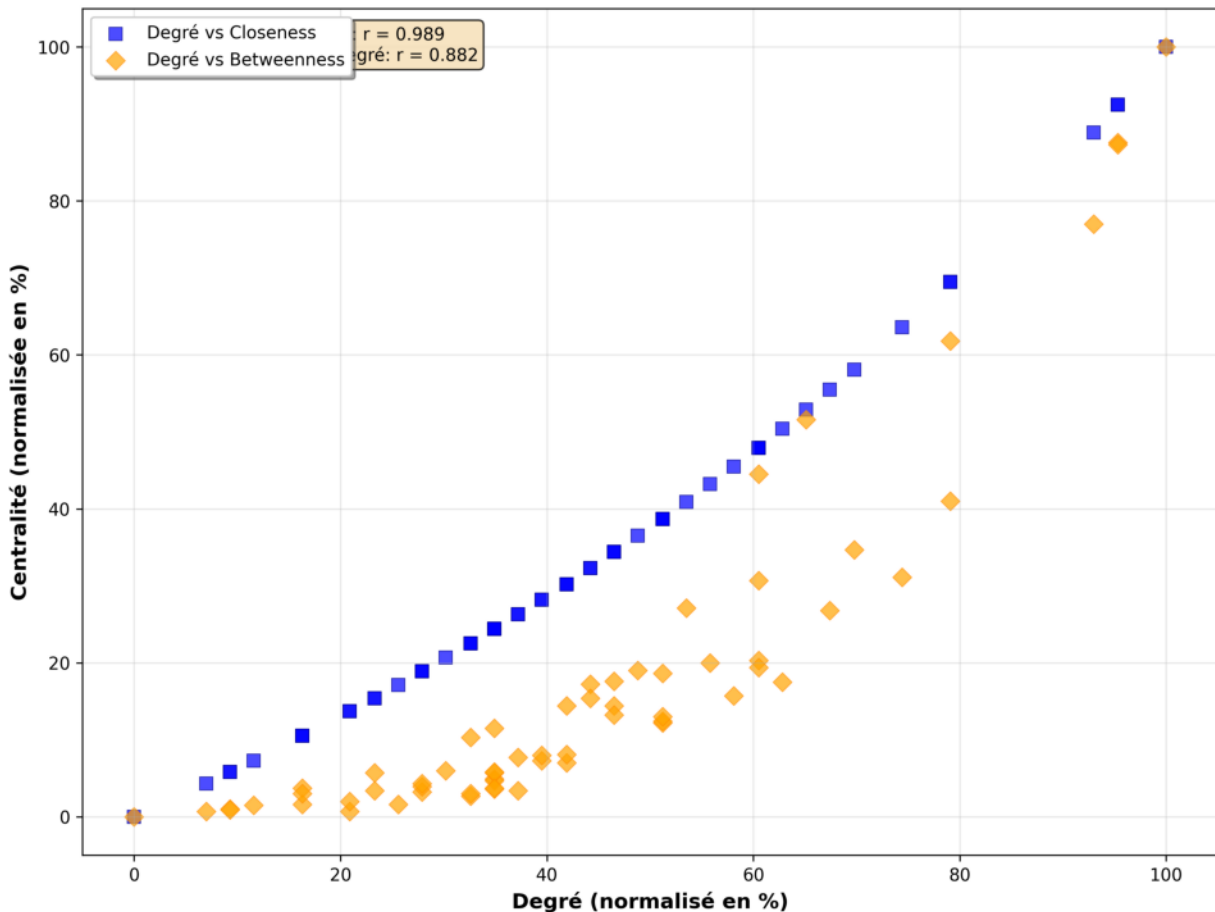
1. Evolution du reseau (noeuds, aretes, densite)
2. Comparaison des centralites
3. Distribution des degres par fenetre
4. Heatmap des centralites
5. Temporel vs Statique
6. Evolution des top noeuds
7. Metriques de connectivite
8. Resume statistique
9. Analyse des leaders
10. Tendances temporelles

5.5 Integration Ex1+Ex2+Ex3

Le tableau de bord unifie toutes les analyses precedentes dans une vision globale multi-temporelle du reseau.

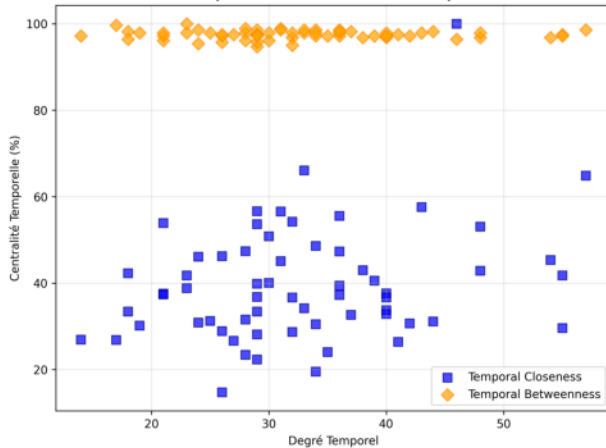
Exercice 2 : Diagrammes de centralité

Analyse de Centralité: Closeness & Betweenness vs Degré

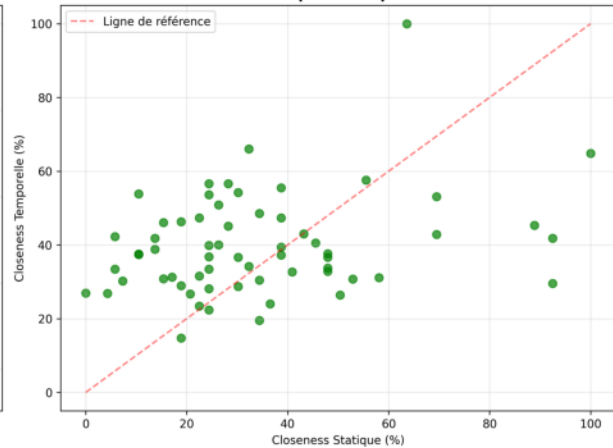


Exercice 3 : Analyse temporelle

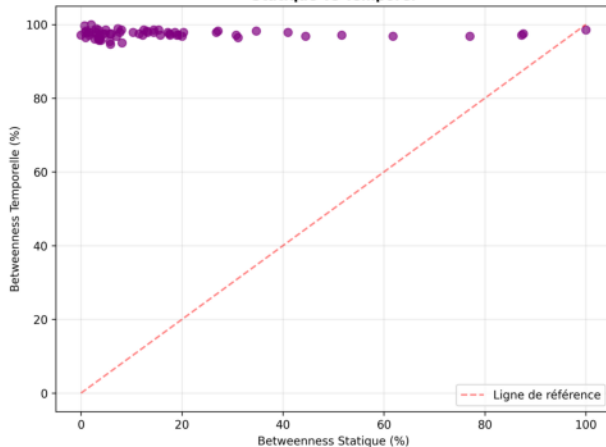
Centralités Temporelles vs Degré
(Contrainte: chemins < 30min)



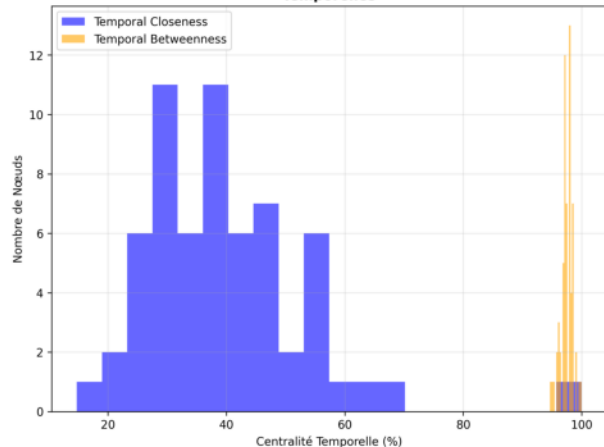
Comparaison Closeness Statique vs Temporel



Comparaison Betweenness Statique vs Temporel



Distribution des Centralités Temporelles



Exercice 4 : Tableau de bord complet

TME4 Ex4 - Dashboard Complet Rollernet Analyse Multi-Temporelle et Comparative

