

---

# Rapport de Projet : Algorithme de Floyd-Warshall

## Partie 2 : Exemple d'Application Réelle

---

### Groupe de projet :

1. Aya LAHOUEL
2. Meriem TIH
3. Zoubir SAHNOUN
4. Mohamed Hedi MOURALI
5. Mohamed Ilyes BEN SAID
6. Mohamed SAKHO

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>L'Algorithme de Floyd-Warshall</b>	<b>2</b>
2.1	Principe Général . . . . .	2
2.2	Fonctionnement Mathématique . . . . .	2
<b>3</b>	<b>Contexte du Problème : Suivi Épidémiologique</b>	<b>2</b>
3.1	La Modélisation par le Graphe . . . . .	3
3.2	L'Objectif de l'Algorithme . . . . .	3
<b>4</b>	<b>Les Données Construites</b>	<b>3</b>
<b>5</b>	<b>Résultats et Interprétation</b>	<b>4</b>
5.1	Cas d'étude : Du Patient 0 au Patient 19 . . . . .	4
5.2	Résultat de l'Algorithme . . . . .	4
5.3	Analyse . . . . .	4
<b>6</b>	<b>Conclusion</b>	<b>4</b>

# 1 Introduction

Ce rapport présente la seconde partie de notre projet consacré à l'algorithme de Floyd-Warshall. Après avoir implémenté l'algorithme et développé une interface de visualisation en Python capable de traiter les graphes de test, nous avons conçu un cas d'application concret pour démontrer son utilité.

L'objectif est double : d'une part, expliciter le fonctionnement théorique de cet algorithme de programmation dynamique, et d'autre part, prouver sa pertinence dans un contexte réel autre que les réseaux informatiques classiques. Nous avons choisi de modéliser la propagation d'un virus au sein d'une communauté.

## 2 L'Algorithme de Floyd-Warshall

Avant de présenter notre application, il convient de rappeler brièvement le principe de fonctionnement de l'algorithme utilisé.

### 2.1 Principe Général

L'algorithme de Floyd-Warshall est un algorithme d'analyse de graphes qui permet de trouver les plus courts chemins entre **tous les couples de sommets** dans un graphe orienté pondéré qui peut comporter des poids négatifs, à condition qu'il n'y ait pas de circuits absorbants.

### 2.2 Fonctionnement Mathématique

Soit un graphe  $G$  avec des sommets numérotés de 1 à  $N$ . L'algorithme construit une série de matrices  $D^{(k)}$ , où  $D_{i,j}^{(k)}$  représente la longueur du plus court chemin du sommet  $i$  au sommet  $j$  en n'utilisant comme sommets intermédiaires que les sommets de l'ensemble  $\{1, 2, \dots, k\}$ .

La relation de récurrence est la suivante :

$$D_{i,j}^{(k)} = \min \left( D_{i,j}^{(k-1)}, \quad D_{i,k}^{(k-1)} + D_{k,j}^{(k-1)} \right)$$

Cela signifie qu'à chaque étape  $k$ , pour chaque paire  $(i, j)$ , on vérifie si passer par le sommet intermédiaire  $k$  offre un raccourci par rapport au chemin connu précédemment.

- **Initialisation** ( $k = 0$ ) : La matrice est initialisée avec les poids des arêtes directes (et l'infini s'il n'y a pas d'arête).
- **Itération** : On répète le calcul pour  $k$  allant de 1 à  $N$ .
- **Complexité** : L'algorithme a une complexité temporelle de  $O(N^3)$ , ce qui le rend très efficace pour des graphes denses de taille moyenne.

## 3 Contexte du Problème : Suivi Épidémiologique

Pour notre exemple d'application, nous avons transposé ce problème mathématique de minimisation de distance à un problème de modélisation de risque sanitaire.

### 3.1 La Modélisation par le Graphe

Le graphe orienté et valué est construit selon les principes suivants :

- **Les Sommets (Nœuds 0 à 19)** : Chaque sommet représente une personne unique au sein de la communauté.
- **Les Arcs (Liens)** : Un arc orienté de A vers B signifie qu'il y a eu un contact physique ou une proximité suffisante pour permettre une transmission. L'orientation est importante car la chaîne de contamination suit une direction temporelle ou une susceptibilité (A peut contaminer B, mais B ne peut pas remonter le temps pour contaminer A).
- **Les Poids (Valuation)** : Le poids représente le "**Niveau de Résistance à la Contamination**" (ou l'inverse du risque).
  - **Poids faible (ex : 2)** : Contact très proche (famille, conjoint), le virus passe très facilement (résistance faible).
  - **Poids élevé (ex : 25)** : Contact distant ou protégé (collègue éloigné, masque), le virus a du mal à passer (résistance forte).

### 3.2 L'Objectif de l'Algorithme

Dans ce contexte, l'algorithme de Floyd-Warshall nous permet de calculer le "**Chemin de Moindre Résistance**" entre deux individus quelconques. Si une personne *X* est infectée ("Patient Zéro"), Floyd-Warshall nous permet d'identifier immédiatement quelle chaîne de contacts est la plus susceptible de transmettre le virus jusqu'à la personne *Y*, même s'ils ne se connaissent pas directement.

## 4 Les Données Construites

Nous avons construit un jeu de données de **20 sommets** et **49 arcs**. La structure sociale a été pensée pour être réaliste :

1. **Clusters Familiaux** : Des groupes de nœuds (ex : 0, 1, 2, 3) fortement connectés avec des poids très faibles (2), représentant des foyers.
2. **Liens Professionnels** : Des liens entre clusters (ex : nœuds 10 à 15) représentant un lieu de travail, avec des poids modérés.
3. **Ponts Sociaux** : Quelques individus (ex : 5, 6, 7) connectés à plusieurs groupes différents, agissant comme des vecteurs de transmission potentiels ("super-spreaders").

Voici un extrait du fichier `exemple.txt` utilisé par notre programme :

Listing 1 – Extrait du fichier de données

```
1 20
2 49
3 0 1 2      (Famille: risque eleve)
4 0 5 10     (Ami: risque moyen)
5 ...
6 10 15 20   (Travail: risque faible)
7 ...
```

## 5 Résultats et Interprétation

Nous avons exécuté notre programme Python sur ces données. L'interface a permis de générer la matrice des plus courts chemins. Nous nous concentrons ici sur un cas critique révélé par l'analyse.

### 5.1 Cas d'étude : Du Patient 0 au Patient 19

Nous cherchons à savoir comment le virus passerait de l'individu **0** (au début du graphe) à l'individu **19** (à la fin), qui appartiennent à des cercles sociaux totalement différents.

### 5.2 Résultat de l'Algorithme

L'algorithme a déterminé que le coût minimal de transmission est de **40**. Le chemin reconstruit est le suivant :

$$0 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 14 \rightarrow 19$$

### 5.3 Analyse

Ce résultat est particulièrement intéressant pour l'analyse épidémiologique :

- **Évitement des liens évidents** : L'algorithme n'a pas choisi de passer par les collègues de travail (Nœud 10), car bien que le lien existe, le poids (la résistance à l'infection) était trop élevé (20 – 25).
- **Rôle des ponts sociaux** : Le virus a emprunté un chemin via les nœuds 5, 6 et 7. Ces individus représentent des contacts "passerelles" qui relient des familles distinctes.
- **Propagation en cascade** : Bien que l'individu 0 ne connaisse pas l'individu 19, une chaîne de 7 intermédiaires suffit à créer un risque élevé de contamination cumulée.

## 6 Conclusion

Ce projet nous a permis de maîtriser l'algorithme de Floyd-Warshall non seulement dans sa théorie et son implémentation technique, mais aussi dans son utilité pratique.

En transposant le problème mathématique de "minimisation de la somme des poids" à un problème social de "maximisation du risque de contagion", nous avons démontré que cet algorithme est un outil puissant pour l'analyse de réseaux complexes, capable de révéler des connexions indirectes invisibles à l'œil nu.