

Rapport d'Analyse des Algorithmes de Plus Court Chemin

Lucie Grandet, Tristan Limousin, Antoine Groussard

4 avril 2025

1 Introduction

Dans le cadre de ce projet, nous avons étudié trois algorithmes classiques de recherche de plus court chemin dans un graphe : Dijkstra, Bellman-Ford et Floyd-Warshall. Ces algorithmes ont été implémentés en C# et testés sur le graphe construit dans le cadre du premier Rendu du 1er mars. Celui-ci représente les relations entre membres d'une association, où chaque membre est un nœud et chaque relation un lien non orienté. En rajoutant des poids allant de 1 à 5, ce nouveau graphe est plus adapté à celui représentant les lignes de métro de Paris.

2 Implémentation des algorithmes

Les trois algorithmes ont été implémentés dans un projet à part selon les spécifications classiques :

1. **Dijkstra** : Algorithme glouton qui trouve le plus court chemin depuis un nœud source vers tous les autres nœuds dans un graphe à poids positifs.
2. **Bellman-Ford** : Algorithme plus général qui gère les poids négatifs et détecte les cycles négatifs.
3. **Floyd-Warshall** : Algorithme qui calcule les plus courts chemins entre toutes les paires de nœuds.

Après avoir compiler ce projet dédié à l'étude comparative de ces trois algorithmes, on a :

```

Définir un noeud de départ
2
Définir un noeud d'arrivée
3
Que voulez-vous faire ?
1 - Dijkstra Chemin Plus court
2 - Dijkstra Distance
3 - BellmanFord Chemin Plus Court
4 - BellmanFord Distance
5 - Floyd-Warshall Distance
6 - Floyd-Warshall Chemin Plus Court
7 - étude comparative
7

Paires de noeuds la plus éloignée : 17 -> 25 (distance: 13)

=== Benchmark (1000 itérations) ===
Algorithmme      Moyenne (ms)   Min      Max      Écart-type
Dijkstra         0,0674         0,0481    0,6635   0,0368
Bellman-Ford     0,0502         0,0385    0,4011   0,0298
Floyd-Warshall   4,8566         4,2715    8,4975   0,6447

```

FIGURE 1 – Enter Caption

3 Résultats et Analyse

3.1 Performances comparées

Les résultats du benchmark sur 1000 itérations donnent les temps d'exécution moyens suivants :

TABLE 1 – Comparaison des performances des algorithmes

Algorithmme	Temps moyen (ms)	Écart-type
Dijkstra	0.0674	0.0368
Bellman-Ford	0.0502	0.0298
Floyd-Warshall	4.8566	0.6447

3.2 Analyse des résultats

1. Complexité algorithmique :

- Dijkstra : $O((n + m) * \log(n))$ pour un graphe de n noeuds et m liens.
- Bellman-Ford : $O(n * m)$ pour un graphe de n noeuds et m liens.
- Floyd-Warshall : $O(n^3)$

2. Rapidité d'exécution :

- Floyd-Warshall est significativement plus lent car il calcule les chemins pour toutes les paires de nœuds, ce qui explique sa complexité cubique.
- Dijkstra et Bellman-Ford ont des performances similaires sur ce graphe, avec un léger avantage pour Bellman-Ford dans notre implémentation.

3. Pertinence pour notre cas d'application :

- Pour une recherche ponctuelle de plus court chemin entre deux nœuds, Dijkstra est le plus adapté pour notre graphe, ne contenant que des poids positifs (des distances).
- Bellman-Ford offre des garanties similaires, bien que légèrement plus rapide dans notre test (dû à l'implémentation). Cependant, sa complexité théorique le rend moins scalable sur des graphes denses.
- Floyd-Warshall n'est pertinent que si on a besoin de calculer tous les plus courts chemins à l'avance. En effet, son calcul exhaustif de tous les chemins représente un gaspillage de ressources pour une requête unique.

3.3 Justification des choix

Dans le contexte d'une application de livraison entre un client et un cuisinier :

1. **Dijkstra** serait l'algorithme privilégié car :
 - Tous les poids sont positifs (distances réelles)
 - Il offre les meilleures performances pour une requête unique
 - Et il est facile à implémenter avec une file de priorité
2. **Bellman-Ford** aurait pu être utilisé comme une alternative simple, si le graphe utilisé était plus petit.
3. **Floyd-Warshall** serait trop coûteux pour une utilisation ponctuelle, mais pourrait être intéressant si on précalcule tous les chemins pour des requêtes ultérieures rapides.

4 Conclusion

Cette étude comparative montre que pour notre cas d'application spécifique (recherche de plus court chemin entre deux points dans un graphe), l'algorithme de Dijkstra est le plus approprié. Il combine une bonne complexité algorithmique avec de bonnes performances pratiques. Bellman-Ford offre une alternative valable avec une implémentation potentiellement plus simple. Floyd-Warshall, s'avère trop coûteux pour une utilisation ponctuelle dans notre contexte.

Ces résultats confirment les attentes théoriques et démontrent l'importance de choisir l'algorithme en fonction des besoins spécifiques de l'application.