

Graphe et Réseau

Algorithme de Dijkstra bidirectionnel

Prof: Jean-François Hêche

Assistant: Thibaud Franchetti

Étudiant : Nelson Jeanrenaud

Introduction

L'objectif de ce laboratoire est d'implémenter les algorithmes de Dijkstra et Dijkstra bidirectionnel dans leur version orientée pour le calcul de plus court chemin entre deux sommets d'un réseau. Et de documenter les performances des deux algorithmes.

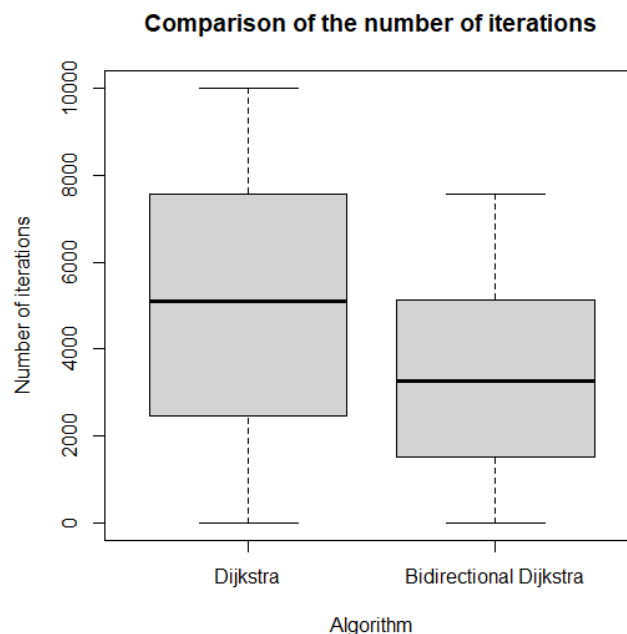
Les classes de lecture de graphes ont été fournies en début de laboratoire.

Comparaison des performances

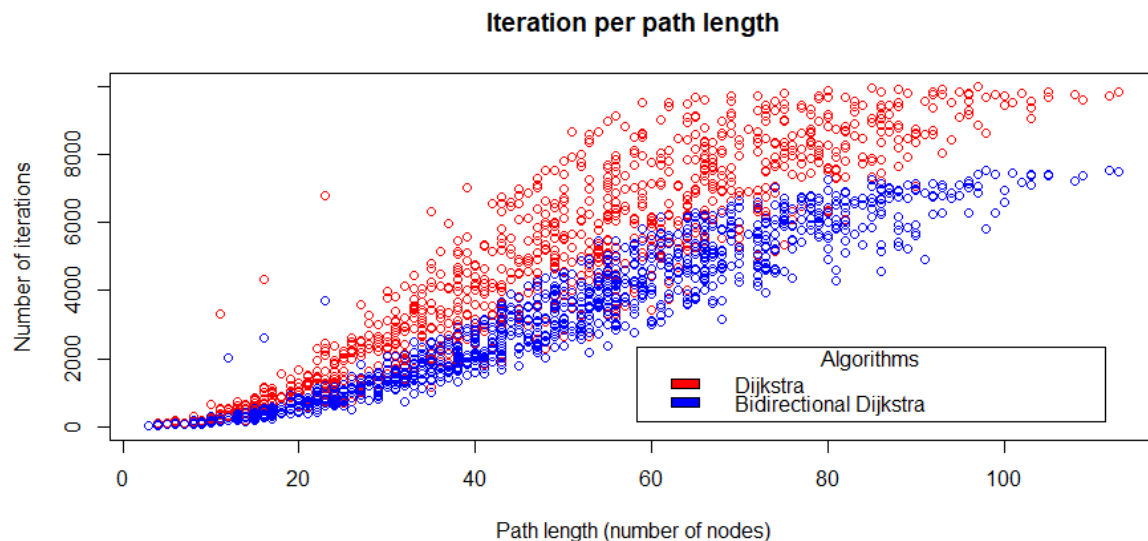
Méthodologie

Pour mesurer les performances des deux algorithmes je génère 1000 couples de sommets (source, destination) aléatoirement et tente de trouver le plus court chemin entre chacun de ces couples avec les deux algorithmes. Ensuite nous enregistrons l'état de fin des algorithmes pour chaque recherche (nombre d'itérations, longueur du chemin, poids du chemin). Les résultats suivants ont été obtenus sur un graphe de 1000 sommets.

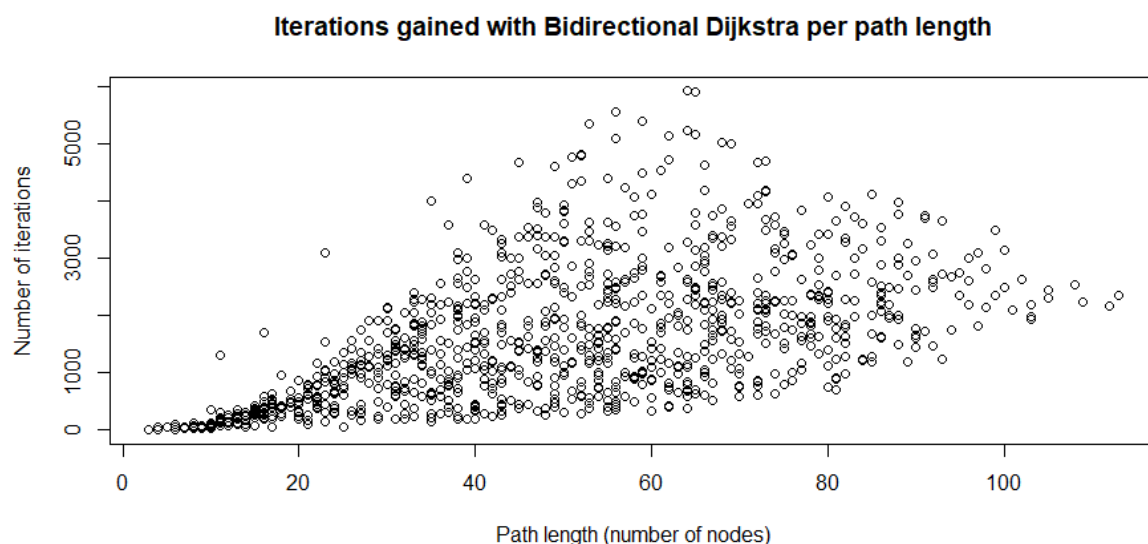
Résultats



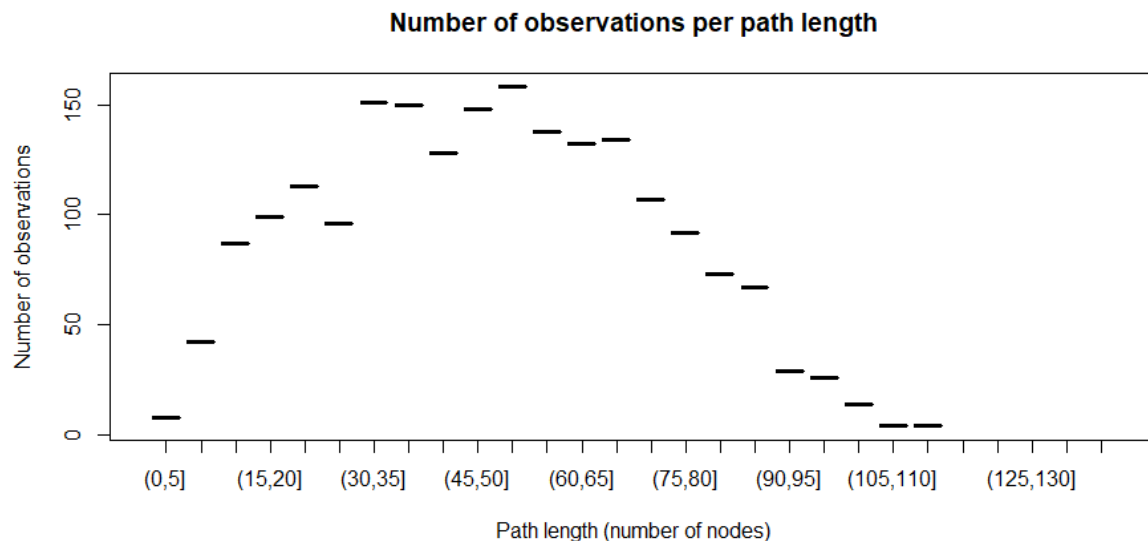
Premièrement, on peut remarquer une amélioration du nombre d'itérations avec l'algorithme de Dijkstra bidirectionnel. On passe d'une moyenne de 5009 itérations avec Dijkstra à 3366 itérations, soit une amélioration de 1643 itérations (réduction de 32.8%).



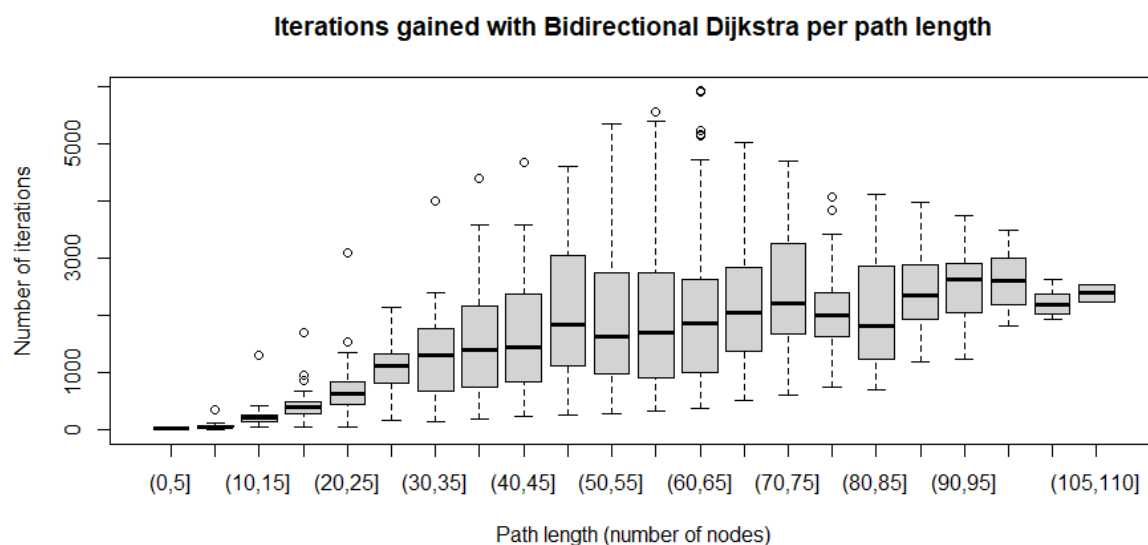
Assez logiquement, on peut observer que l'amélioration obtenue avec la version bidirectionnelle est plus conséquente quand le plus court chemin est assez long. Cependant la variance est assez grande et on peut remarquer un chevauchement entre les deux nuages de points. Cela est dû au fait que pour certaines combinaisons de sommets les arborescences en avant et en arrière se rejoignent en même temps que la seule arborescence de Dijkstra simple rejoint le sommet destination. Par la même pensée on peut imaginer des cas où Dijkstra simple trouve une solution en moins d'itérations.



Pour ce test nous n'avons pas obtenu ce cas théorique. Mais nous pouvons remarquer que les gains en itérations semblent atteindre leur pic à environ 60 sommets de longueur. J'attribue ça au manque de données de grandes longueurs de chemin et la variance des résultats.

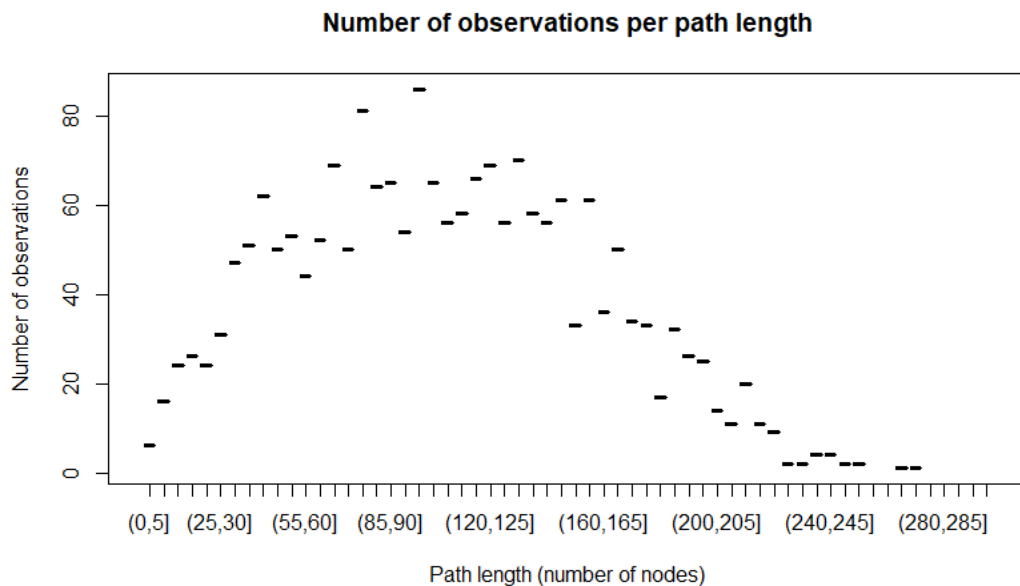


La distributions des observations est unimodale

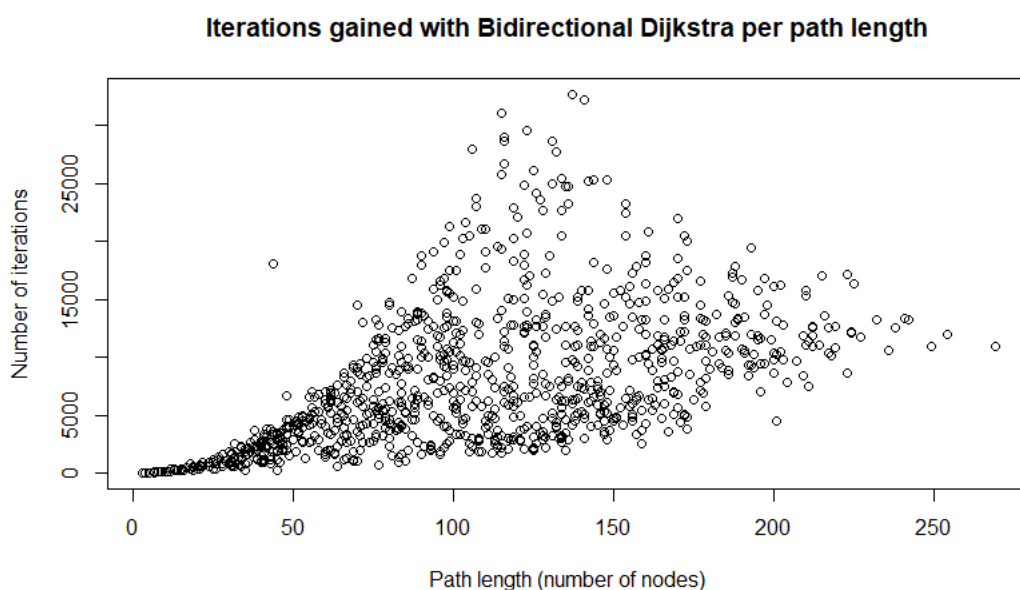


Boxplot des observations en les regroupants tout les 5 sommets dans le plus court chemin

Effectivement, La majorités des observations sont dans l'intervall $[30,65]$. On remarque également que l'écart-type est très grand pour ces valeurs "centrales". Cela explique facilement la distribution obtenue en figure 3. Et en observant les boxplots de la figure 5 on peut voir que la médiane augmente bien avec la longueur du chemin trouvé.

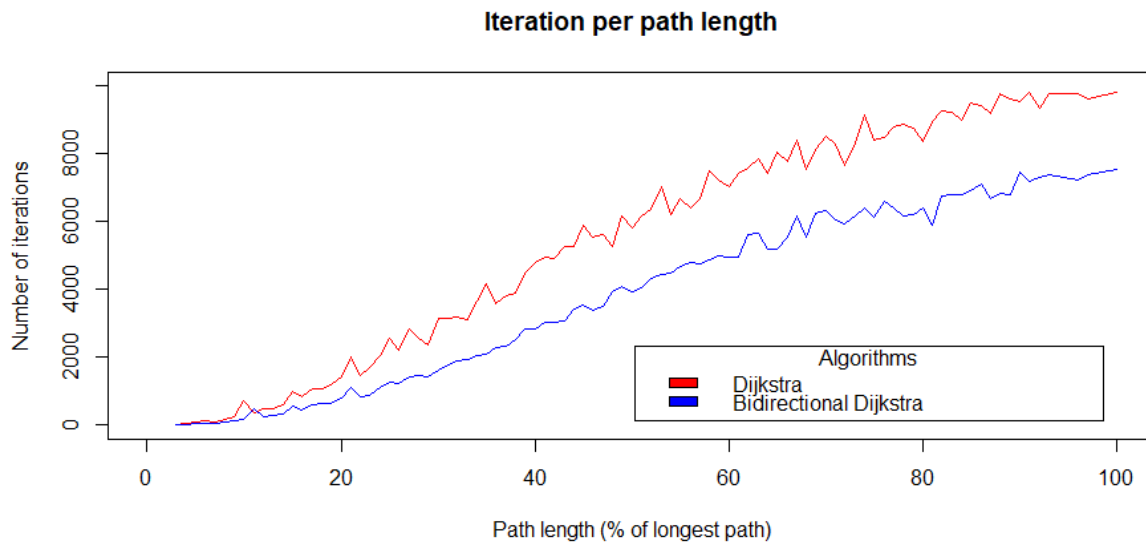


Distribution des observations du test sur le graphe de 5000 sommets



Itérations gagnées en utilisant le graphe de 5000 sommets

En testant les algorithmes avec le graphe de 5000 sommets ont obtenu des distributions de formes identiques. On peut en conclure que les observations avec un grand nombre d'itération au milieu de la distribution sont dû à la variance et non aux performances des algorithmes par rapport à cette longueur de chemin.



Moyenne du nombre d'itérations par rapport à la taille du chemin trouvé

Pour illustrer cette amélioration, ce graphique montre la moyenne du nombre d'itération des deux algorithmes. Les données sont regroupées par le pourcentage arrondi du plus long chemin traité lors du test.

Conclusion

Dijkstra bidirectionnel offre effectivement un gain non négligeable en nombre d'itération par rapport à la version simple. En moyenne le nombre d'itération est réduit de 32.8% mais ce gain augmente avec la longueur du plus court chemin. Dans des graphes relativement petits, les coûts en complexité spatiale de dijkstra bidirectionnel ne vaudra pas, dans beaucoup de cas, le gain en performance.

On a pu également observer que ces gains ne sont pas une garentie absolue et que sur certaines configuration de graphe; sommet de départ; sommet d'arrivée, Dijkstra simple peut offrir de meilleures performances.