

GRE Labo2 - Algorithme de Dijkstra bidirectionnel

Auteur: Valentin Kaelin

Date: 15.05.2022

Contexte

Pour réaliser la comparaison entre la version classique de l'algorithme de Dijkstra et sa version bidirectionnelle, 1000 couples de sommets (s à t) ont été générés aléatoirement avec la seed 20220404. Par la suite, les plus courts chemins pour chaque couple ont été calculés (avec les deux versions de l'algorithme) sur le graphe à 10'000 sommets du fichier `R10000_1.txt`. Le nombre de sommets traités par chaque version ont été stockés afin de pouvoir analyser les résultats.

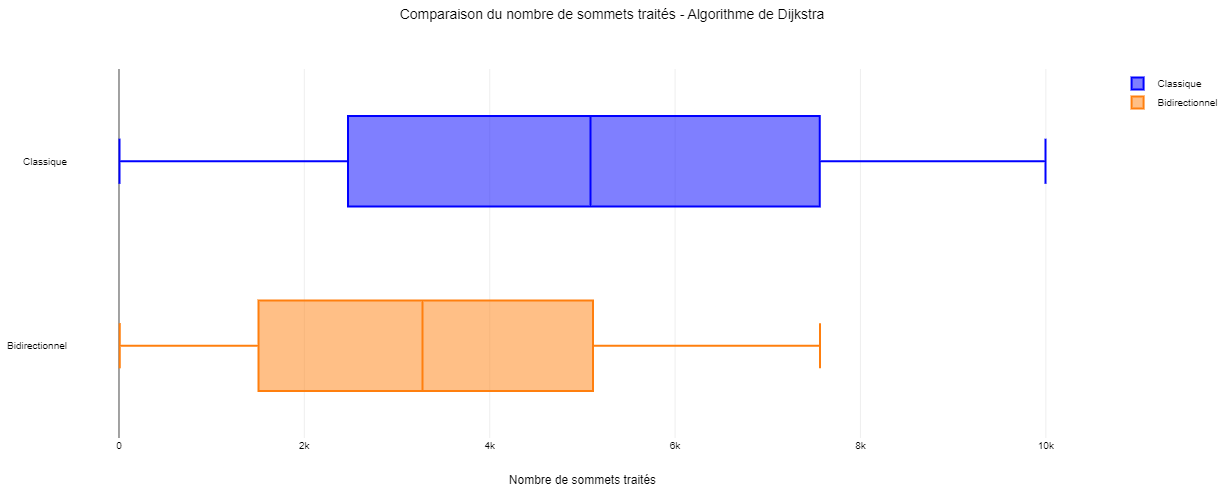
Résultats

Un fichier `.csv` a été généré avec les 1000 couples générés aléatoirement. Voici un extrait de son contenu:

#	Source	Destination	Nb sommets traités classique	Nb sommets traités bidirectionnel
1	7611	5959	5892	5070
2	5024	9036	3261	2684
3	1569	5469	7599	6225
4	3363	7911	198	120
5	6946	4726	3127	2163
6	9083	3706	6391	5558
7	8743	8774	7972	5649
8	190	2756	1019	698
9	245	2721	7828	3992
10	8293	1964	8973	6400
...
997	3945	9007	4094	2910
998	9113	6972	680	335
999	589	522	7503	5544
1000	7777	1353	2293	1151

Pour ne pas surcharger le rapport, le fichier complet est disponible à [l'adresse suivante](#) en cas de besoin.

Par la suite, un boxplot a été généré grâce au site statskingdom.com:



Ce site nous met également diverses statistiques intéressantes à disposition:

	Classique	Bidirectionnel
Taille des datasets	1000	1000
Minimum	5	7
Maximum	9997	7563
Q1	2472	1506.5
Q3	7560	5113.5
Médiane	5088.5	3274
Moyenne	5007.835	3365.418

Analyse

Grâce au boxplot, la différence nous saute aux yeux directement. La version bidirectionnelle traite bien moins de sommets que la version classique (ouf!). Les écarts entre les moyennes et les médianes sont assez faibles, ce qui nous indique qu'il n'y a pas trop d'outliers venant fausser la moyenne.

Nous pouvons donc calculer le pourcentage d'amélioration de la version bidirectionnelle:

```
100 - 3274 * 100 / 5088.5 = -35.66% d'amélioration
```

Une autre remarque que nous pouvons faire grâce au tableau est la suivante: l'algorithme classique, bien que moins performant, a tout de même un minimum plus bas que la version bidirectionnelle. Cela est dû aux conditions d'arrêts différentes entre les deux versions. Nous pouvons donc supposer que la version classique peut être plus performante lorsque le plus court chemin est très court.

Afin d'essayer de confirmer cette théorie, j'ai filtré les 1000 résultats en gardant uniquement les chemins de longueur ≤ 6 (valeur arbitraire pour avoir quand même quelques résultats).

Source	Destination	Nb sommets traités classique	Nb sommets traités bidirectionnel
98	3821	22	12
260	187	92	46
442	1664	5	7
547	6651	90	46

Ce petit test nous remontre qu'en réalité la version classique n'est que très rarement plus efficace sur un graphe de cette envergure, même lorsque les chemins sont courts.

Pour être encore plus sûr, j'ai affiché les résultats plus performants avec la version classique que la version bidirectionnelle et ils se comptent au nombre de 2 sur les 1000 (0.2% du temps):

Source	Destination	Nb sommets traités classique	Nb sommets traités bidirectionnel	Plus court chemin
1664	8920	5	7	[1664, 9250, 8920]
4143	2386	47	48	[4143, 7606, 5077, 8310, 8763, 2386]

Dans les deux cas, le plus court chemin est court comme nous avons pu le remarquer précédemment.

Conclusion

Pour conclure, cette expérience nous a bien démontré que la version bidirectionnelle est plus efficace que la version classique en terme de nombre traitements de sommets. Il existe cependant des cas pour lesquels la version classique est plus efficace mais ceux-ci sont très rares (plus courts chemins de longueur très courte). Et mêmes dans ces cas-ci, la différences est très légère.

Nous pouvons donc terminer en indiquant qu'utiliser la version bidirectionnelle de Dijkstra, malgré une implémentation un peu plus compliquée, est la bonne chose à faire si les performances sont un critère important dans le problème à résoudre.