



INSA Toulouse
156 Avenue de Rangueil,
31400 Toulouse

Rapport tests Bureau d'étude Graphe

Année universitaire : 2014-2015
Groupe : 3 MIC C
Binôme : AHAMADA Abdillah
CABANNES Jordy

Introduction

Au travers de ce rapport nous allons tester les performances de nos deux algorithmes de recherche du plus court chemin que sont Dijkstra Standard et A-Star.

A-Star est une extension de Dijkstra Standard, il se base sur le coût entre deux sommets mais également sur une estimation entre le sommet courant et la destination. Quand à Dijkstra Standard il réalise ses calculs seulement sur le coût entre l'origine et le lieu courant.

Nous pourrions voir quel algorithme est le plus performant et dire si oui ou non A-Star est plus efficace que Dijkstra Standard.

Pour cela nous effectuerons tout d'abord des test de validité, c'est-à-dire que nous verrons si les algorithmes sont capables de trouver le plus court chemin, que ce soit en temps ou en distance. Nous pourrions comparer les coûts des chemins ainsi que le nombre de sommets visités au cours de l'exécution des algorithmes.

Puis nous effectuerons des test de performances et nous comparerons les temps d'exécution des algorithmes sur des chemins identiques.

A) Tests de validité

Au travers de ces différents tests nous allons nous assurer que les algorithmes de Dijkstra Standard et de A-Star trouvent bien le plus court chemin que ce soit en distance ou en temps. Nous allons ainsi nous assurer de leur bon fonctionnement et de leur validité.

Tout d'abord nous allons reprendre les tests proposés par le sujet du bureau d'étude, à savoir les chemins sur les cartes de l'INSA, de Paris et de Midi-Pyrénées.

Ensuite nous allons tester nos algorithmes sur un trajet assez long Toulouse → Bordeaux, nous étudierons un trajet ABC puis les trajets AB et BC. Nous testerons également sur un trajet dans un certain sens puis en sens inverse.

Enfin nous allons étudier des cas particuliers tels que le plus court chemin pour un noeud d'origine et de destination identique, pour deux noeuds qui ne sont pas connexes et un chemin où le noeud de destination n'appartient pas à la carte.

1) Test du sujet

a) Chemin de l'INSA

Noeud de départ : 2

Noeud d'arrivée : 139

Résultat attendu : d'après le sujet, en temps nous devons obtenir 2,235 minutes.

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	1171	10	43
	Temps (en minutes)	2,2348	10	58
Dijkstra Standard	Distance (en mètres)	1171	11	63
	Temps (en minutes)	2,2348	10	61

Nous remarquons que pour Dijkstra Standard et A-Star en temps nous retrouvons bien les 2,235 minutes attendues. Nos algorithmes de calcul du plus court chemin en temps semblent donc correct.

b) Chemin de Paris

Noeud de départ : 268

Noeud d'arrivée : 418

Résultat attendu : d'après le sujet, en temps nous devons obtenir 2,863 minutes.

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	2218	50	254
	Temps (en minutes)	2,8632002	59	491
Dijkstra Standard	Distance (en mètres)	2218	31	536
	Temps (en minutes)	2,8632002	50	537

Une fois de plus les résultats des deux algorithmes en mode temps correspondent à ce qui était affiché dans le sujet.

c) Chemin de Midi-Pyrénées

Noeud de départ : 119963

Noeud d'arrivée : 96676

Résultat : d'après le sujet, en temps nous devons obtenir un peu plus de 200 minutes.

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	291351	926	92193
	Temps (en minutes)	200,69861	1099	122474

Dijkstra Standard	Distance (en mètres)	291351	464	140077
	Temps (en minutes)	200,69861	801	141927

Les résultats pour Dijkstra Standard et A-Star en temps sont égaux et proches de 200 minutes.

Après cette série de trois tests nous pouvons en conclure que nos algorithmes de calcul du plus court chemin en temps semblent fonctionner comme il était attendu. Étant donné que nous nous appuyons sur le même raisonnement dans les algorithmes de calcul du plus court chemin en distance, nous pouvons dire également que les résultats obtenus en distance seront sans doute aussi les meilleurs possibles. Nos algorithmes de calcul du plus court chemin en distance seront sans doute eux aussi corrects. De plus nous remarquons que pour chaque test le nombre de sommets marqués par l'algorithme A-Star est bien moins important que celui de Standard. Cela prouve bien que A-Star est plus efficace, il permet de trouver le plus court chemin plus en minimisant le nombre de sommets visités.

2) Autres tests

a) Long trajet

Carte : France

Noeud de départ : 145668 (Toulouse)

Noeud d'arrivée : 2228923 (Bordeaux)

Résultat : D'après Google Maps en temps nous devrions être proche de 142 min et en distance 246 km.

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	286257,0	1039	125107
	Temps (en minutes)	155,8048	1644	132989
Dijkstra Standard	Distance (en mètres)	286257,0	708	471739
	Temps (en minutes)	155,8048	1480	482567

On ne retrouve pas exactement les mêmes résultats que Google Maps, cela peut s'expliquer par le fait que nos cartes sont moins à jour que celles utilisées par Google Maps. On a un écart d'environ 40 km et de 12min. Nos algorithmes sont assez performants et on trouve le même résultat que l'on utilise A-Star ou Dijkstra Standard.

[b\) Trajet ABC](#)

Nous allons tester nos algorithmes sur un trajet $A \rightarrow C$, puis nous prendrons un noeud B sur le plus court chemin entre A et C. De là nous testerons sur les trajets $A \rightarrow B$ puis $B \rightarrow C$ pour voir si l'on retombe sur le même plus court chemin que celui de A vers C.

Carte : Midi-Pyrénées

- [Trajet A \$\rightarrow\$ C](#)

Noeud de départ A : 127298

Noeud d'arrivée C : 36615

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	254764	853	89444
	Temps (en minutes)	172,49806	1236	117709
Dijkstra Standard	Distance (en mètres)	254764	478	143979
	Temps (en minutes)	172,49806	893	143401

- [Trajet A \$\rightarrow\$ B](#)

Noeud de départ A : 127298

Noeud d'arrivée B : 125189

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance	10842,0	16	27

	(en mètres)			
	Temps (en minutes)	7.92	15	35
Dijkstra Standard	Distance (en mètres)	10842,0	24	204
	Temps (en minutes)	7,92	26	171

- [Trajet B → C](#)

Noeud de départ B : 125189

Noeud d'arrivée C : 36615

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	243922	863	88109
	Temps (en minutes)	164.57	1229	110736
Dijkstra Standard	Distance (en mètres)	243922	537	143979
	Temps (en minutes)	164.57	941	143401

Si l'on additionne les coûts en temps et en distance des trajets $A \rightarrow B$ et $B \rightarrow C$ on retrouve les coûts du trajet $A \rightarrow C$. Cependant on constate que le nombre de sommets marqués et le nombre max d'éléments sont plus importants quand on décompose le trajet $A \rightarrow C$.

[c\) Trajet en sens inverse](#)

Carte : Midi-Pyrénées

Noeud de départ : 127298

Noeud d'arrivée : 36615

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	254470	902	49461
	Temps (en minutes)	170,81822	1217	103474
Dijkstra Standard	Distance (en mètres)	254470	383	129945
	Temps (en minutes)	170,81822	951	132489

En sens inverse on trouve des coûts moins importants cela s'explique sûrement par des routes à sens unique qui ne peuvent être empruntées que de C vers A. Une fois de plus on remarque que l'algorithme A-Star est plus efficace que l'algorithme Standard, beaucoup moins de sommets ont été marqués et le même chemin a été trouvé.

3) Tests particuliers

a) Trajet entre deux noeuds non connexes

Carte : Morbihan

Noeud de départ : 19004 (Morbihan île)

Noeud d'arrivée : 5441 (Morbihan en France)

Résultat attendu : Le coût doit être infini

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	INFINITY	55	1446
	Temps (en minutes)	INFINITY	79	1446
Dijkstra Standard	Distance (en mètres)	INFINITY	47	1446
	Temps (en minutes)	INFINITY	73	1446

À la fin de ces tests tous les sommets de l'île ont été marqués et une fois que le tas a été vidé, les algorithmes se sont arrêtés. Ils ont tous affiché un coût infini et le message « le tas est vide il n'existe pas de chemin ».

b) Même noeud de départ et d'arrivée

Carte : INSA

Noeud de départ : 2

Noeud d'arrivée : 2

Résultat attendu : Tous les algorithmes doivent afficher un coût de 0.

	Mode	Coût du PCC	Nombre max d'éléments dans le tas	Nombre de sommets marqués
A-Star	Distance (en mètres)	0	1	1
	Temps (en minutes)	0	1	1
Dijkstra Standard	Distance (en mètres)	0	1	1
	Temps (en minutes)	0	1	1

Comme c'était attendu les algorithmes donnent tous un coût de 0 et seul le sommet n°2 a été marqué, aucun de ses successeurs n'a été marqué avant que le programme ne se termine.

c) Destination pas sur carte

Carte : INSA

Noeud de départ : 2

Noeud d'arrivée : 2456373

L'algorithme s'arrête, le message « l'origine ou la destination ne se trouve pas sur le graphe » s'affiche.

4)Conclusion

Nous voyons donc bien au travers de ces tests que A-Star est plus performant dans tous les cas que Dijkstra Standard, il trouve le plus court chemin en marquant moins de sommets. De plus nos algorithmes traitent de nombreux cas d'erreurs.

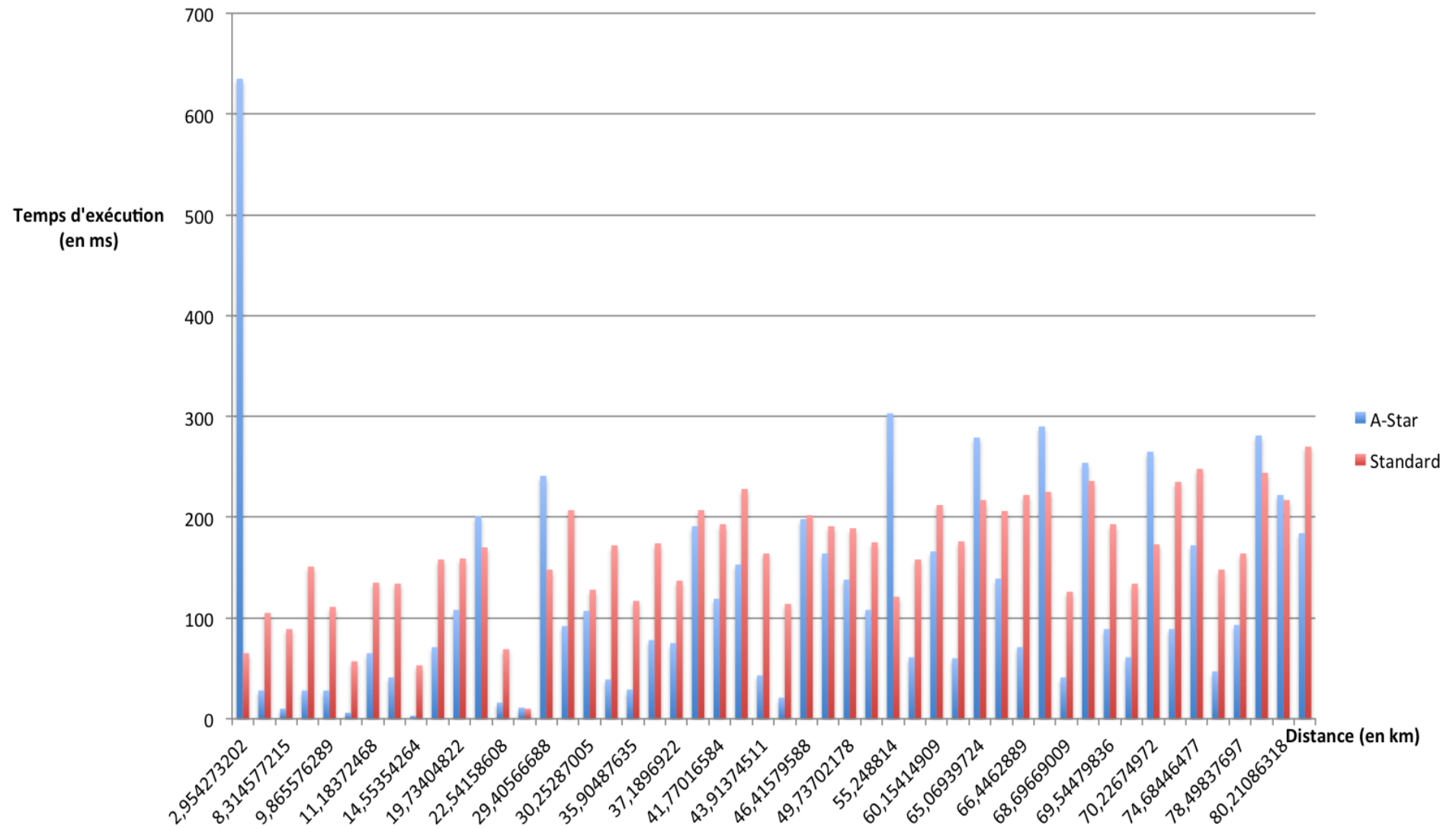
B) Tests de performances

L'objectif de cette partie est de comparer en terme de temps de calcul, l'algorithme standard de Dijkstra et l'algorithme de A-Star.

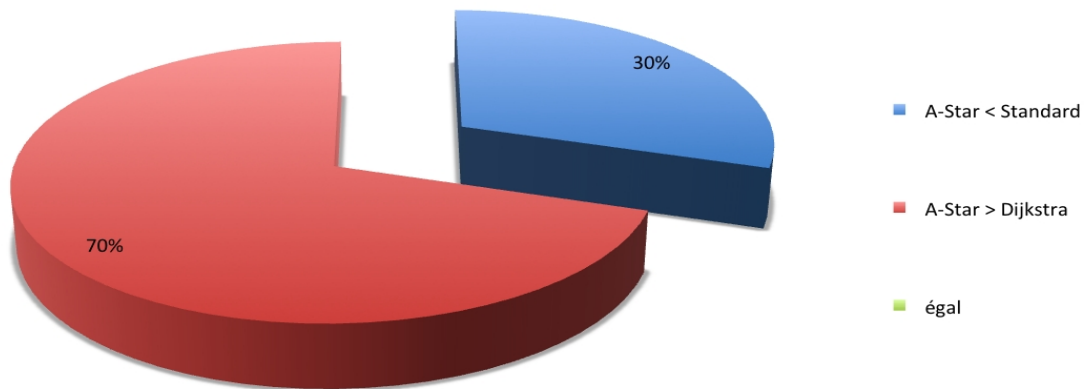
Pour cela nous faisons 100 tests automatiques qui consistent à choisir aléatoirement deux sommets sur la carte choisie, on lance les deux algorithmes sur les deux mêmes sommets et on mesure le temps de calcul.

Premier test de performances de calculs effectué sur la carte : Midi-Pyrénées.

Temps d'exécution des algorithmes en fonction de la distance



Pourcentages des comparaisons des temps d'exécution



Sur ce graphique on visualise en rouge les tests où A-Star est plus performant que Dijkstra Standard, et en bleu les tests où Dijkstra Standard est plus performant que A-Star.

Quelques chiffres clés :

Temps moyen A* : 188 ms

Temps moyen Dijkstra * : 207 ms

Écart max entre A* et Dijkstra si A-Star plus performant : 570 ms

Écart max entre Dijkstra et A* si Dijkstra plus performant : 174 ms

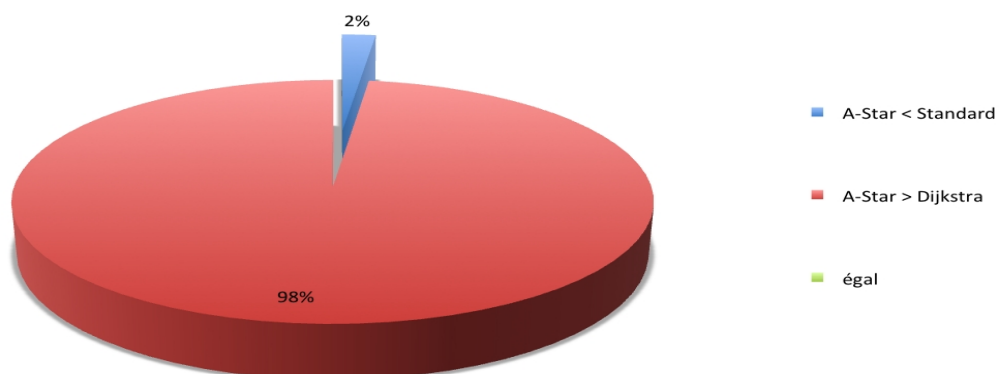
Nombre moyen de noeuds parcourus avec A* : 42799

Nombre moyen de noeuds parcourus Dijkstra : 74988

Second test de performances de calculs effectué sur la carte : Morbihan.

Pourcentages des comparaisons des temps d'exécution

11



Quelques chiffres clés :

Temps moyen A* : 14 ms

Temps moyen Dijkstra * : 70 ms

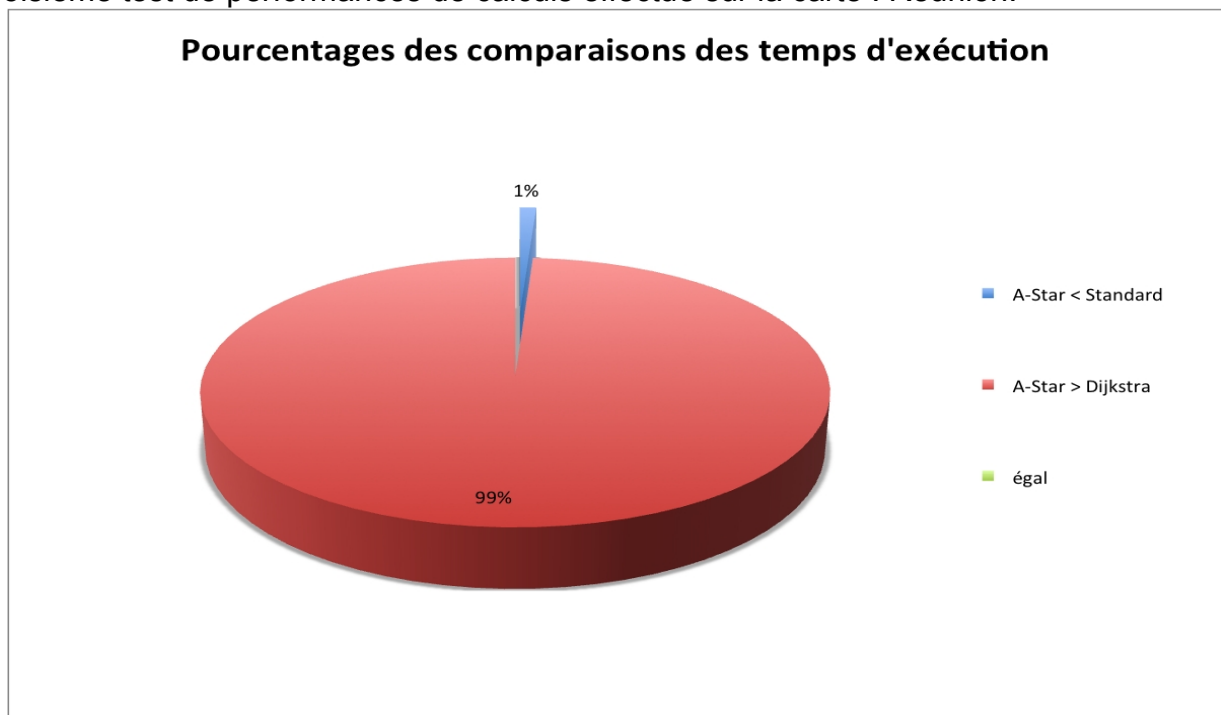
Écart max entre A* et Dijkstra si si A-Star plus performant : 127 ms

Écart max entre Dijkstra et A* si Dijkstra plus performant : 15ms

Nombre moyen de noeuds parcourus avec A* : 5797

Nombre moyen de noeuds parcourus Dijkstra : 6430

Troisième test de performances de calculs effectué sur la carte : Réunion.



Quelques chiffres clés :

Temps moyen A* : 21 ms

Temps moyen Dijkstra * : 88 ms

Écart max entre A* et Dijkstra si si A-Star plus performant : 99 ms

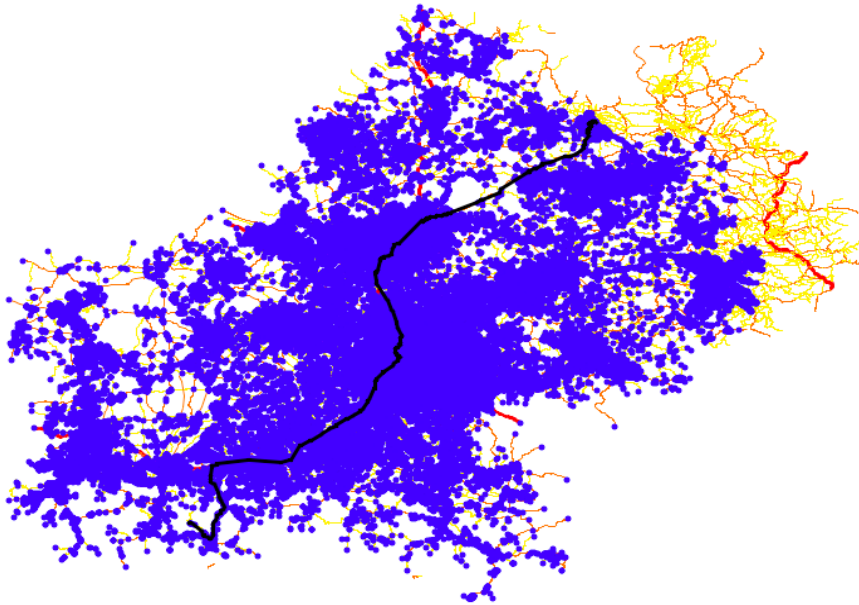
Écart max entre Dijkstra et A* si Dijkstra plus performant : 1 ms

Nombre moyen de noeuds parcourus avec A* : 7404

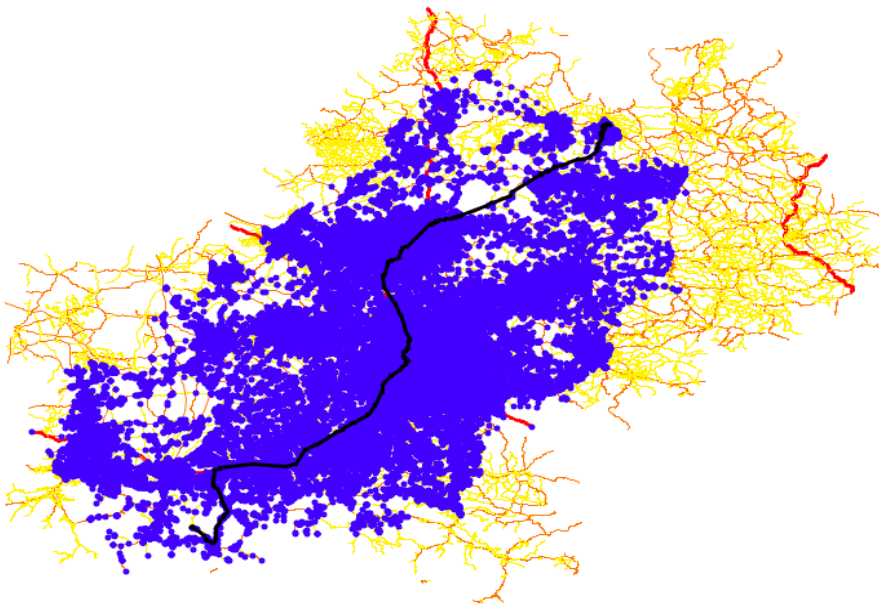
Nombre moyen de noeuds parcourus Dijkstra : 9701

Conclusion

Sur la carte Midi-Pyrénées, sur la carte de Morbihan et sur la carte de la Réunion A* est de façon générale bien plus rapide que Dijkstra. En effet grâce à l'heuristique qui permet de guider l'algorithme (distance restante pour se rapprocher de la destination à vol d'oiseau), A* marque beaucoup de moins de noeuds lors de son exécution, ce qui logiquement diminue son temps d'exécution.



Voici une capture d'écran de l'exécution de l'algorithme de Dijkstra Standard sur Midi-Pyrénées en mode temps



Voici une capture d'écran de l'exécution de l'algorithme A-Star sur Midi-Pyrénées en

mode temps.

On voit bien que A-Star marque moins de sommet que Dijkstra Standard, ce qui explique qu'il est plus efficace sur une carte telle que Midi-Pyrénées.