

BUREAU D'ÉTUDE GRAPHES

---

## Analyse de performances

---

3 MIC E

Guilhem CICHOCKI  
Alexis GIRARDI

# Introduction

Le but du présent document est de présenter les résultats obtenus lors des campagnes de tests que nous avons menées. L'objectif est de prouver l'exactitude ainsi que la performance des deux algorithmes de plus court chemin que nous avons implémentés (Dijkstra et A\*). C'est pourquoi il sera présenté dans une première section les résultats des tests unitaires, visant à démontrer l'exactitude des algorithmes puis dans une seconde section nous montrerons des résultats de performance de nos algorithmes.

## 1 Tests unitaires

L'objectif principal de ces tests unitaires est de montrer que les deux algorithmes que nous avons implémentés donnent effectivement le plus court chemin comme nous le dit la théorie. De plus ils visent à tester le comportement de nos algorithmes dans des cas anormaux et vérifier qu'ils répondent correctement, sans planter. Pour comparer l'exactitude de nos résultats sur les plus courts chemins, nous les compareront aux résultats donnés par Google Maps (qui servira de référence).

### 1.1 Test 1 : Trajet Albertville-Toulon

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
France	27211	792890

Sortie attendue : Selon Google Maps : en distance 418km pour 4h33, en temps 469km pour 4h26

Sortie réelle :

Algorithme	Mode	Coût distance	Coût temps
Dijkstra	Distance	372,542 km	4 h 52 min 30s
A*	Distance	372,542 km	4 h 52 min 30s
Dijkstra	Temps	462,438 km	3 h 45 min 17s
A*	Temps	462,438 km	3 h 45 min 17s

Les résultats diffèrent quelque peu de Google Maps, mais on considère qu'il est beaucoup plus précis que nous. Il doit posséder une heuristique supplémentaire qui lui évite d'emprunter des petites routes ou des cols (non pris en compte par notre heuristique "vol d'oiseau"). De plus son calcul du temps est plus fin. Cependant les résultats sont convenables et nos algorithmes passent les tests. On remarquera qu'en mode temps le plus court chemin retenu est celui passant par l'autoroute A7, contrairement au mode temps qui coupe par les Alpes du sud<sup>1</sup>. On note aussi que nos deux algorithmes donnent le même résultat (ce qui est normal), par la suite on ne donnera plus qu'un seul résultat.

### 1.2 Test 2 : Trajet du sujet en Midi-Pyrénées

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
Midip	119963	96676

Sortie attendue : "Un peu plus de 200 mins (soit environ 3h 20mins)" (*cf. sujet*) en temps et 291.351km selon des tests des années précédentes en distance.

Sortie réelle :

Mode	Coût distance	Coût temps
Distance	291,351km	4 h 42 min 29s
Temps	319,791km	3 h 20 min 41s

Le test est conforme aux attentes, il est validé.

---

1. Voir en annexe C une visualisation graphique du tracé

### 1.3 Test 3 : Sommets non-connexes (France - Corse)

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
France	1953477	2243692

Sortie attendue : Message d'erreur : "Sommet de destination inatteignable"

Sortie réelle : Message d'erreur : "Sommet de destination innatégnable"

Dans ce cas particulier, nos algorithmes ne plantent pas et relèvent le cas "sommet inatteignable".

### 1.4 Test 4 : Trajet nul

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
Réunion	100	100

Sortie attendue : Coût nul en distance et en temps dans les 2 modes.

Sortie réelle :

Mode	Coût distance	Coût temps
Distance	0,0km	0s
Temps	0,0km	0s

Le test est validé.

### 1.5 Test 5 : Sommets inexistant (1/2)

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
Midip	-1	$x$

Sortie attendue : Erreur sommet inexistant

Sortie réelle : Erreur sommet inexistant  
Test validé

### 1.6 Test 5 : Sommets inexistant (2/2)

Entrée :

Carte	Sommet de départ	Sommet d'arrivée
Midip	1000	150828

Sortie attendue : Erreur sommet inexistant

Sortie réelle : Erreur sommet inexistant

Cette batterie de tests unitaires a permis de montrer l'exactitude de nos algorithmes ainsi que leurs réponses adéquates dans des cas anormaux. Tout ceci nous garanti le fonctionnement de nos algorithmes.<sup>2</sup> Étudions maintenant les performances de ces derniers.

---

2. Dans la quasi-totalité des cas : *non-zero bug program*

## 2 Tests de performances

Cette campagne de tests nous permet de comparer les performances des deux algorithmes de plus court chemin que nous avons implémentés à savoir Dijkstra et A\*. Pour ce faire nous avons lancé un grand nombre de tests, pour avoir un échantillon suffisamment représentatif, en choisissant aléatoirement<sup>3</sup> deux sommets sur le graphe. Puis nous lançons nos deux algorithmes entre ces deux sommets. Enfin un script va venir lire les résultats produits par ces tests pour en faire une statistique.

**Note sur les erreurs rencontrées lors de la campagne de tests :** nous désignerons dans ce qui suit par "erreur" tous les cas où la recherche de plus court chemin n'a pu se dérouler dans des conditions normales car le sommet de destination était inatteignable. Ceci s'explique par le fait que les graphes mis à disposition disposent de composantes fortement connexes. En effet les cartes utilisées sont issues d'OpenStreetMap<sup>4</sup> qui sont réalisées par des contributeurs, certains sommets ne sont pas reliés au réseau routier comme des parkings par exemple.

### 2.1 Tests sur la carte de France

Voici les résultats obtenus sur la carte de France (7773 tests lancés dont 292 ont été retirés car relevant d'une erreur soit 3.7% d'erreur, au final les résultats se basent sur un  $n$ -échantillon de 7481 tests) :

Algorithme	Nombre maximal moyen d'éléments dans le tas	Nombre de sommets visités moyen	Temps d'exécution moyen (s)
Dijkstra	0.86	496.59	1.14
A*	0.91	136.20	0.32

TABLE 1 – Résultats de performances sur la carte de France, données par 100km de trajet

Ou encore plus graphiquement :

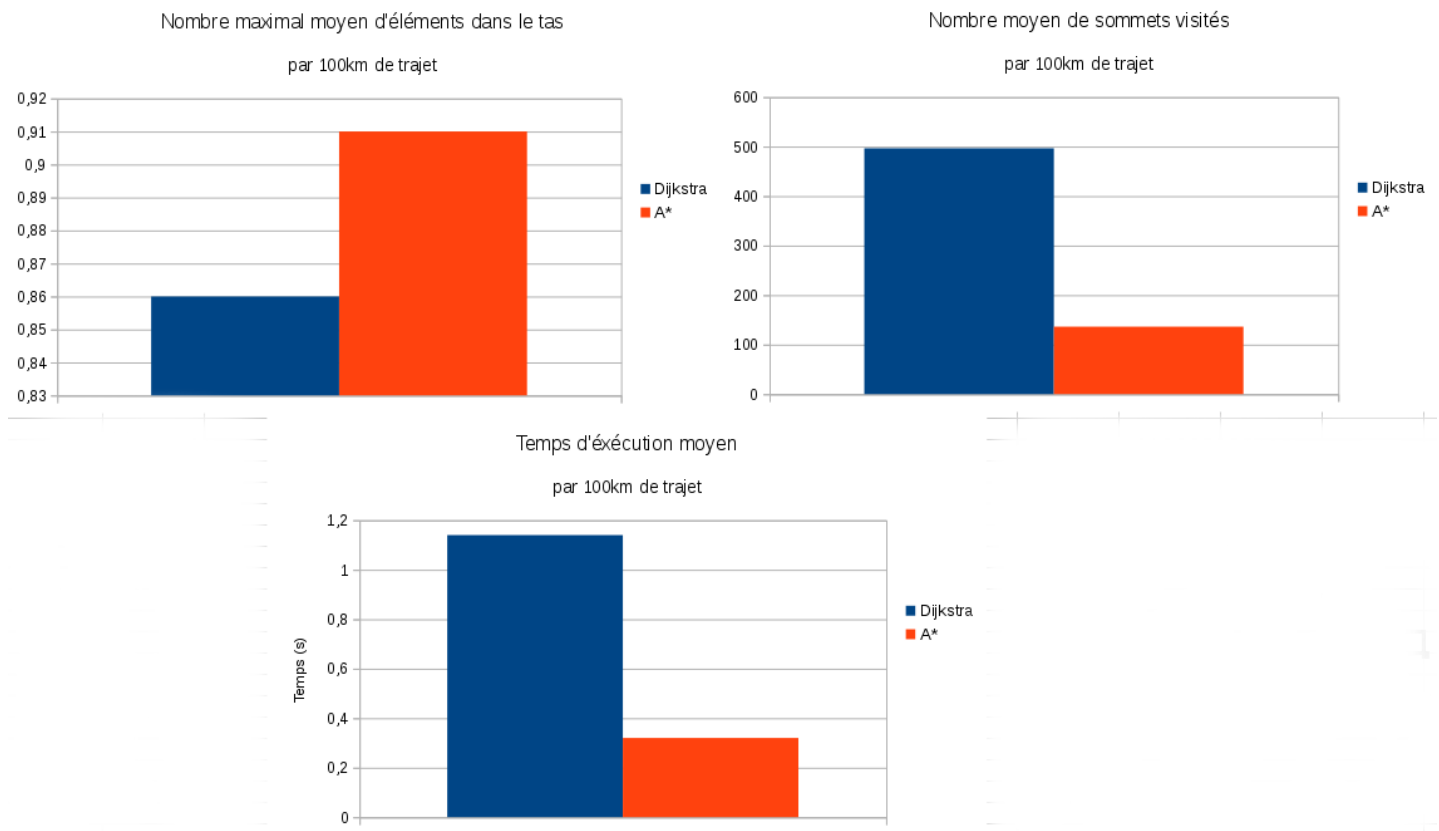


FIGURE 1 – Représentation graphique des résultats de performances sur la carte de France

3. Selon un générateur de loi uniforme

4. <http://www.openstreetmap.org>

On constate la très nette supériorité d'A\* que ce soit en terme de sommets visités ou bien de temps d'exécution, bien qu'il ait une certaine propension à mettre plus d'éléments dans le tas. En terme de temps d'exécution A\* est 3.5 fois plus performant que Dijkstra.

Ceci s'explique par le fait qu'A\* est une extension de Dijkstra conçue pour être plus rapide, grâce à l'utilisation de l'heuristique qui évite à A\* de partir trop loin dans la mauvaise direction. De plus, sur la carte de France la densité de sommets est répartie de façon homogène sur toute la surface de la carte, ce qui rend notre heuristique "distance la plus courte à vol d'oiseau" très performante.

## 2.2 Tests sur la carte de la Réunion

Voici les résultats obtenus sur la carte de l'île de la Réunion (6296 tests lancés dont 100 ont été retirés car relevant d'une erreur soit 1.58% d'erreur, au final les résultats se basent sur un  $n$ -échantillon de 6196 tests) :

Algorithme	Nombre maximal moyen d'éléments dans le tas	Nombre de sommets visités moyen	Temps d'exécution moyen (s)
Dijkstra	0.56	40.79	0.056
A*	0.61	26.06	0.053

TABLE 2 – Résultats de performances sur la carte de France, données par 100km de trajet

Sous forme graphique :

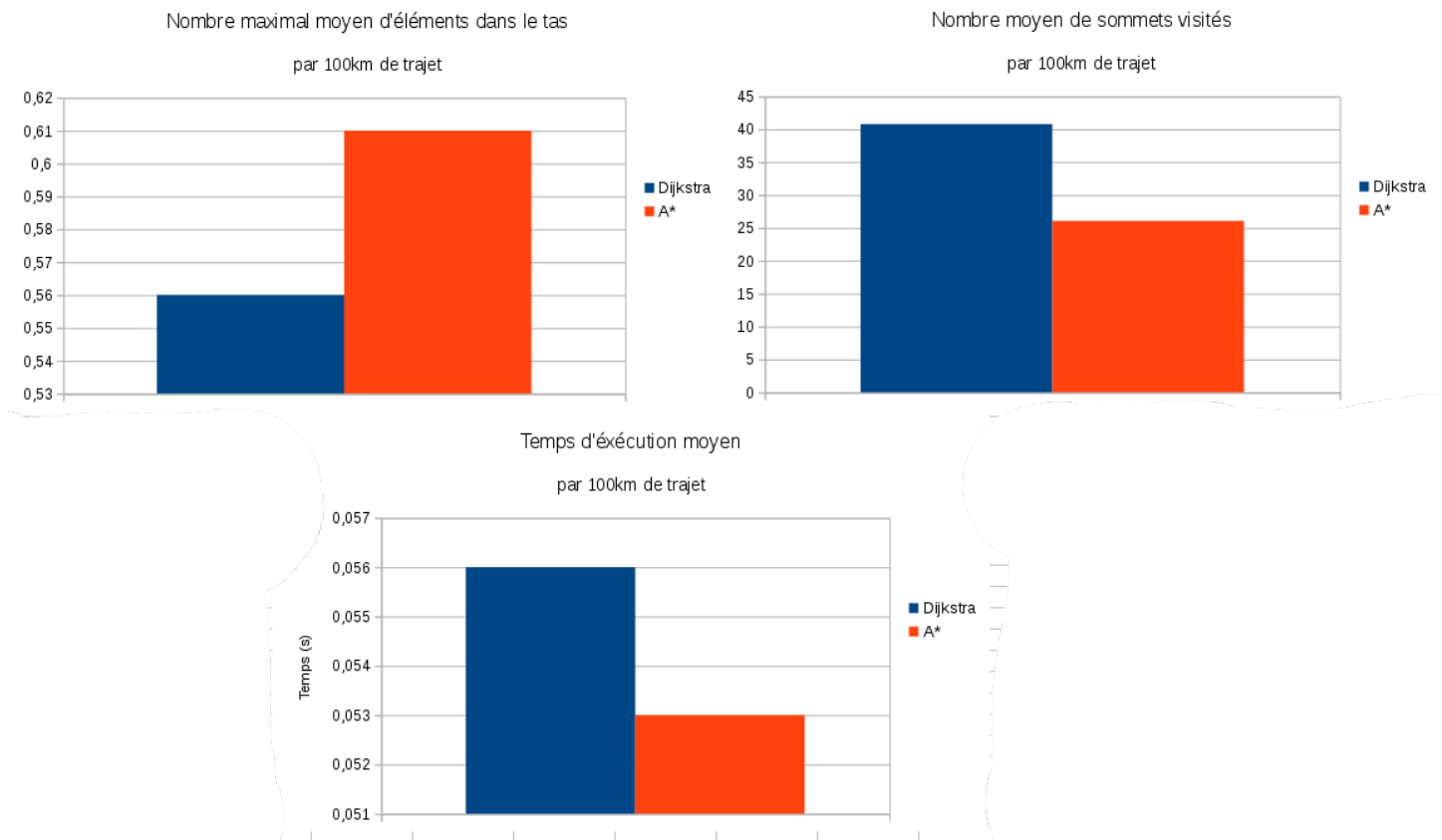


FIGURE 2 – Représentation graphiques des résultats de performances sur la carte de France

Ici encore A\* reste plus performant, mais sa suprématie ne saute plus autant aux yeux. En temps d'exécution, A\* est meilleur de 1.05 fois Dijkstra. Cette baisse d'écart s'explique par le fait que sur l'île de la Réunion les routes sont réparties sur la côte. Ainsi pour aller d'un bout à l'autre de l'île, A\* avec son heuristique aura tendance à vouloir passer par le diamètre de l'île, mais il n'existe aucune route de ce type, il perdra donc du temps.

## A Machines utilisées pour les tests

### A.1 Tests unitaires et tests sur la carte de France

- Processeur : i5-4440 QuadCore @ 3.10GHz
- RAM : 8Go DDR3

### A.2 Tests sur la carte de la Réunion

- Processeur : i7-2600 8-Core @ 3.40GHz
- RAM : 16Go

## B Sources utilisées

- ANALYSE DE PERFORMANCES : BE GRAPHS, *ORFILA*, *OUEDRAOGO*, 2013

## C Tracés de test

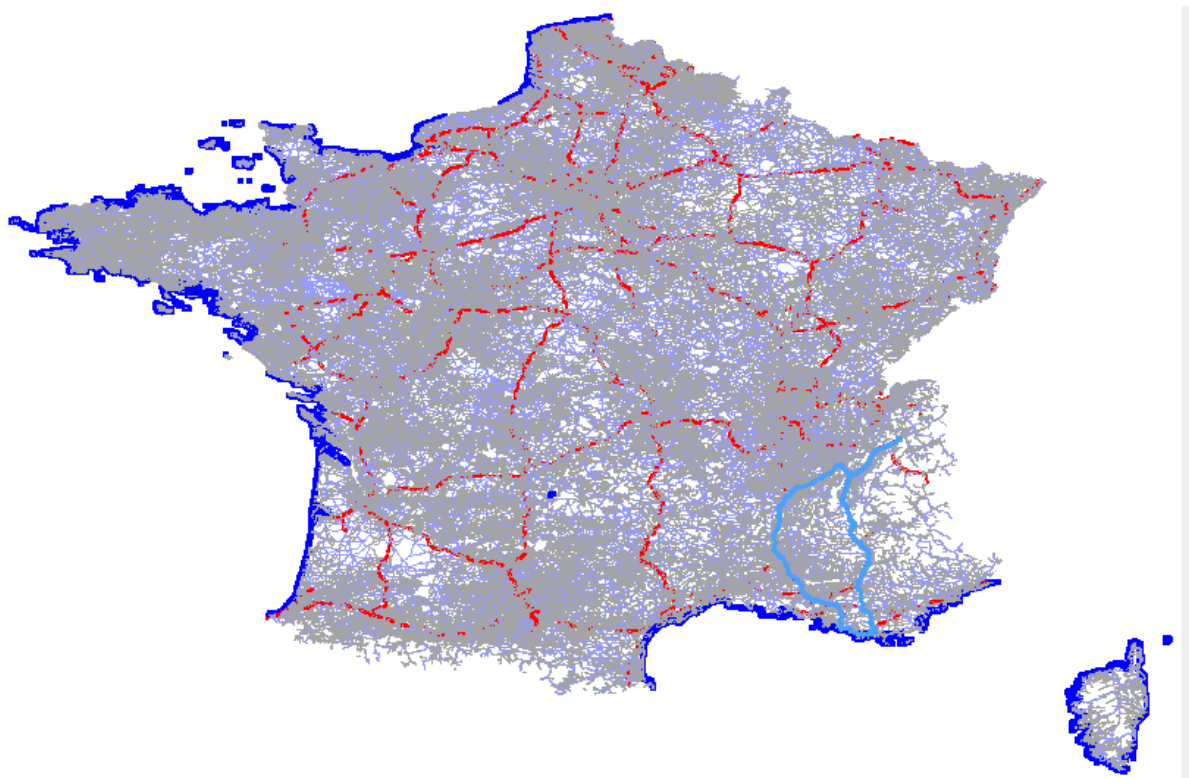


FIGURE 3 – Trajet Albertville-Toulon en temps et en distance