

BE graphes

*Application à l'itinéraire le plus court
entre deux lieux géographiques*

Poulin Lancelot ; Gantet Jérémie 3MIC-IR groupe C

Plan

- I. Introduction et contextualisation
- II. Tests de validité
- III. Tests de performances
- IV. Problème ouvert
- V. Conclusion

Introduction et contexte de développement

- Coder algorithme de **Dijkstra** et sa variante **A*** en Java


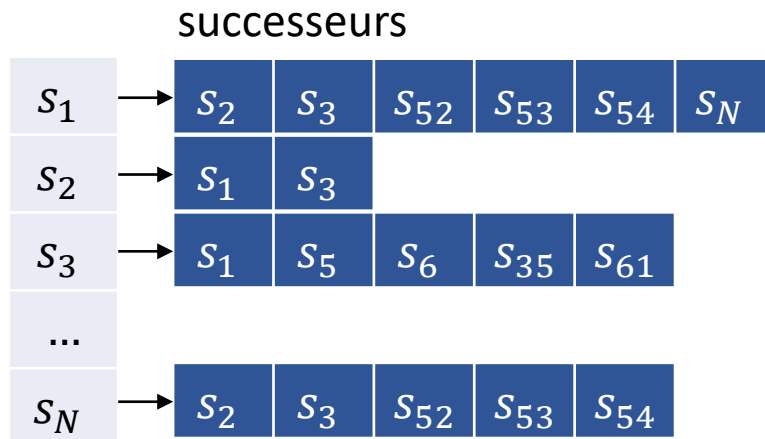
 **GitHub** <https://github.com/LancelotPoulin/BEGraphes>

tableau de liste d'adjacences



Cartes au format .mapgr

<https://data.typeface.fr/insa>

List of graph and maps files (75 maps available)

Map ID	Region name	Graph file	Mapsforge file	Compressed archive
0x800	Fractal [v6] 100770 nodes	fractal.mapgr 28/02/2018	27.87 MB	fractal.tgz 19/03/2018
0x801	Fractal-Spiral [v5] 100770 nodes	fractal-spiral.mapgr 28/02/2018	27.87 MB	fractal-spiral.tgz 19/03/2018
0x850	Carre [v6] 25 nodes	carre.mapgr 28/02/2018	1.65 KB	carre.tgz 19/03/2018
0x851	Carre-Dense [v6] 360000 nodes	carre-dense.mapgr 28/02/2018	30.49 MB	carre-dense.tgz 19/03/2018
BE	Belgique [v6] 1036329 nodes	belgium.mapgr 19/03/2018	42.14 MB	belgium.mapfg 06/01/2018
BJ	Bénin [v6] 130281 nodes	benin.mapgr 19/03/2018	5.90 MB	benin.mapfg 04/01/2018
			22.27 MB	benin.tgz 19/03/2018

Interface graphique



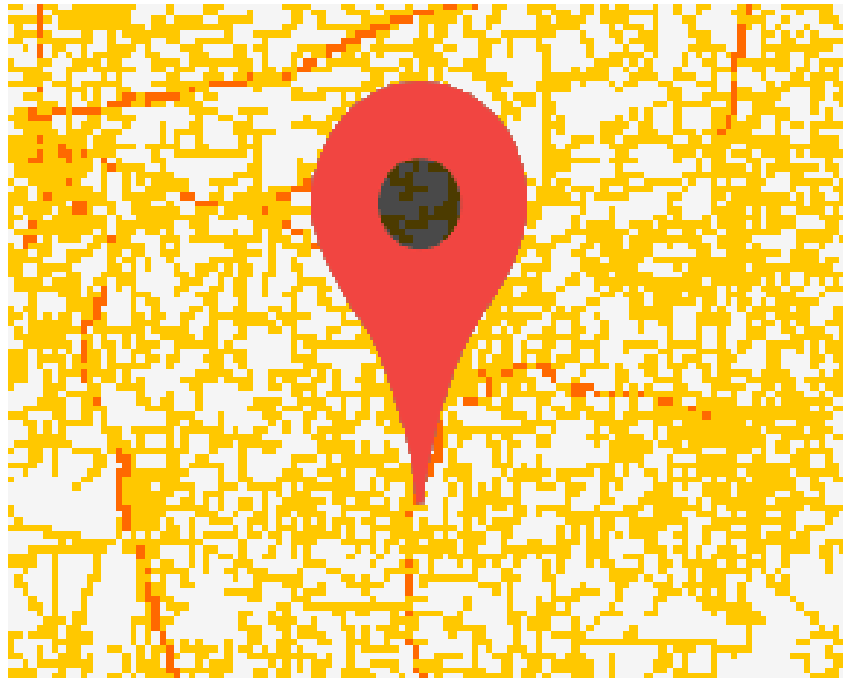
Plan

- I. Introduction et contextualisation
- II. Tests de validité
- III. Tests de performances
- IV. Problème ouvert
- V. Conclusion

Tests de validité

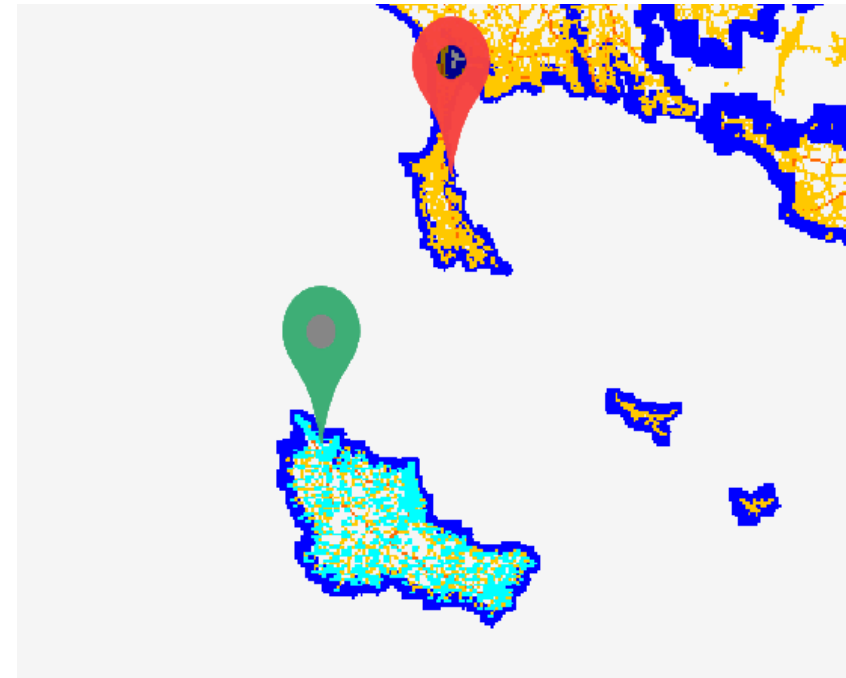
bretagne.mapgr

Chemin unique sommet



Found a path from node #418389 to node #418389, 0,0000 kilometer...

Chemin infaisable

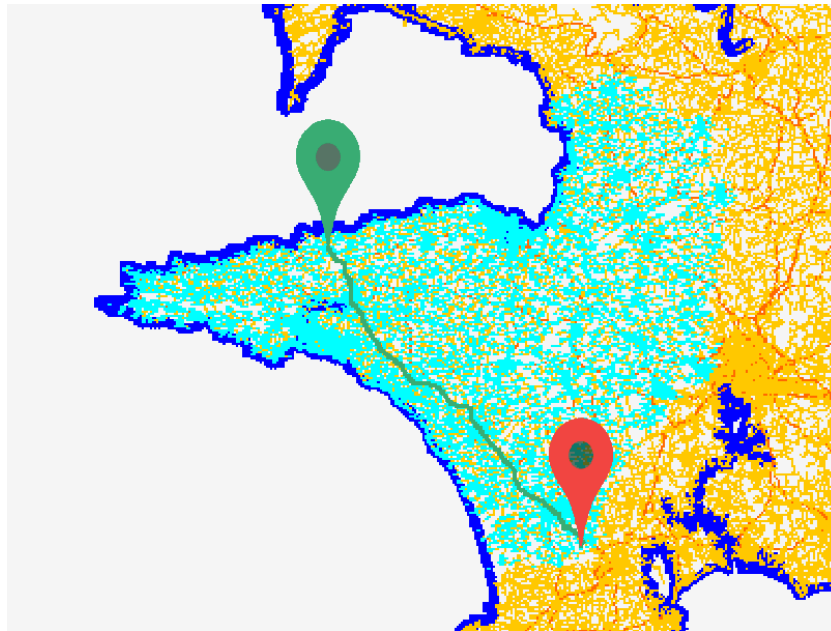


No path found from node #285243 to node #116069 in 0 seconds.

Tests de validité

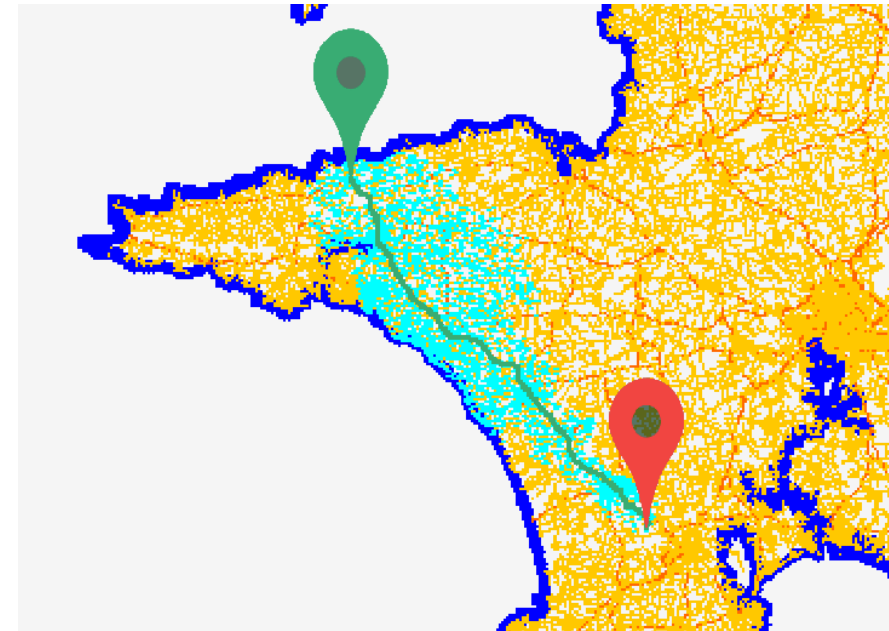
bretagne.mapgr

Dijkstra



Path from #273416 to #598348
Length = 32,723 kilometers, Duration=36 minutes, 50 seconds.

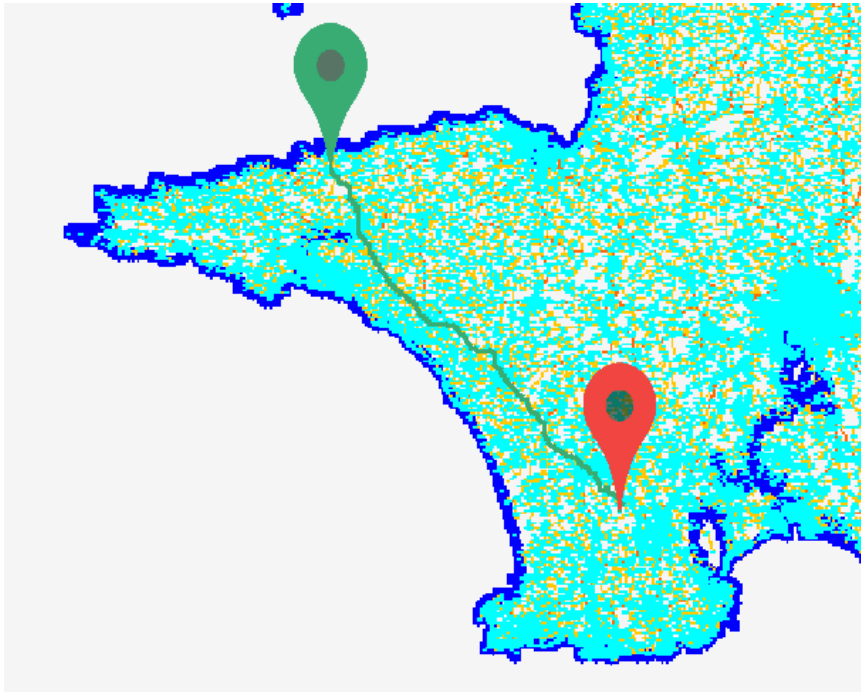
A Star



Path from #273416 to #598348
Length = 32,723 kilometers, Duration=36 minutes, 50 seconds.

Test d'optimalité

Avec oracle : Bellman-Ford



Path from #273416 to #598348
Length = 32,723 kilometers, Duration=36 minutes, 50 seconds.

Sans oracle

Vérification de :

$$\text{Coût}_{\text{distance}}(\text{plus court chemin en } \textit{distance}) \\ = \inf (\text{Coût}_{\text{distance}}(\textit{autres chemins}))$$

$$\text{Coût}_{\text{temps}}(\text{plus court chemin en } \textit{temps}) \\ = \inf (\text{Coût}_{\text{temps}}(\textit{autres chemins}))$$

Plan

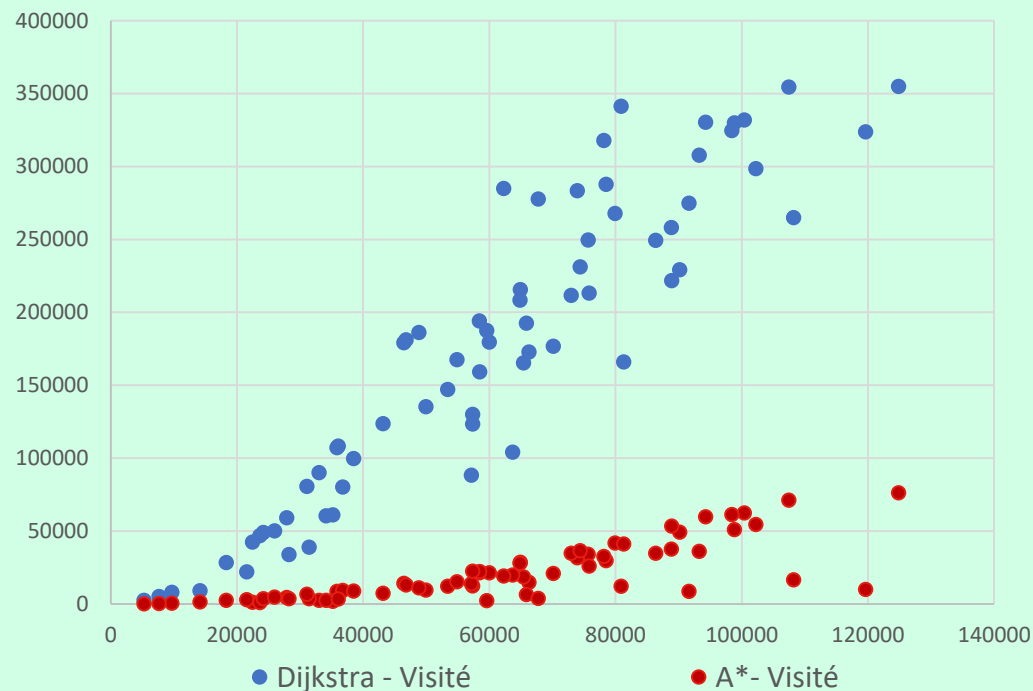
- I. Introduction et contextualisation
- II. Tests de validité
- III. Tests de performances
- IV. Problème ouvert
- V. Conclusion

Carte non routière

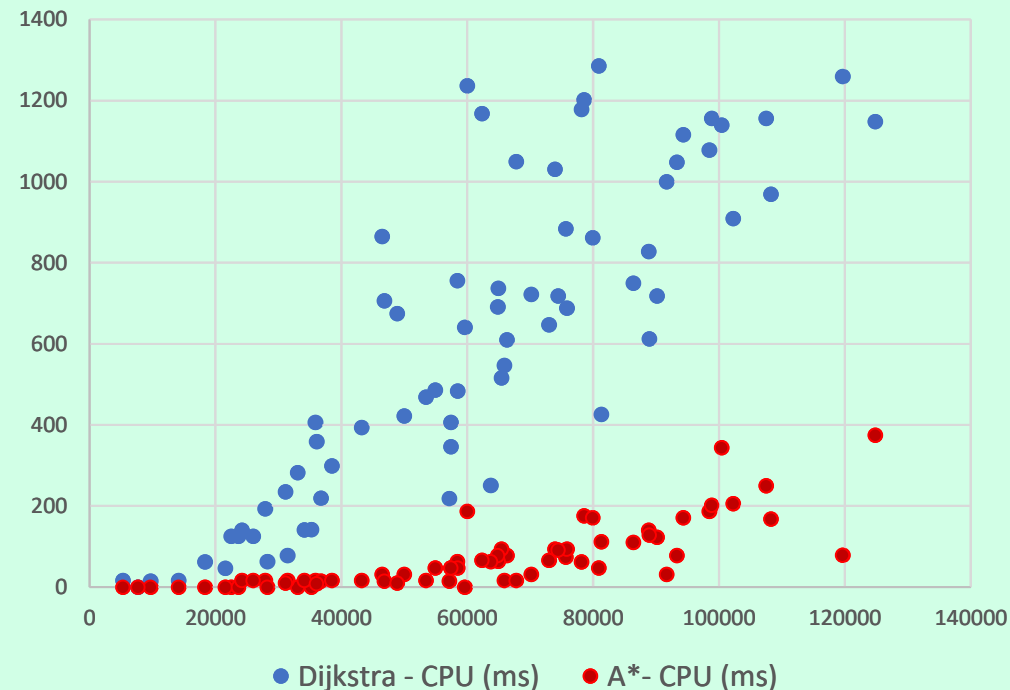
Distance

carre-dense.mapgr

Nombre de **sommets visités** selon la longueur **en m** du chemin le plus court



temps d'exécution selon la longueur **en m** du chemin le plus court

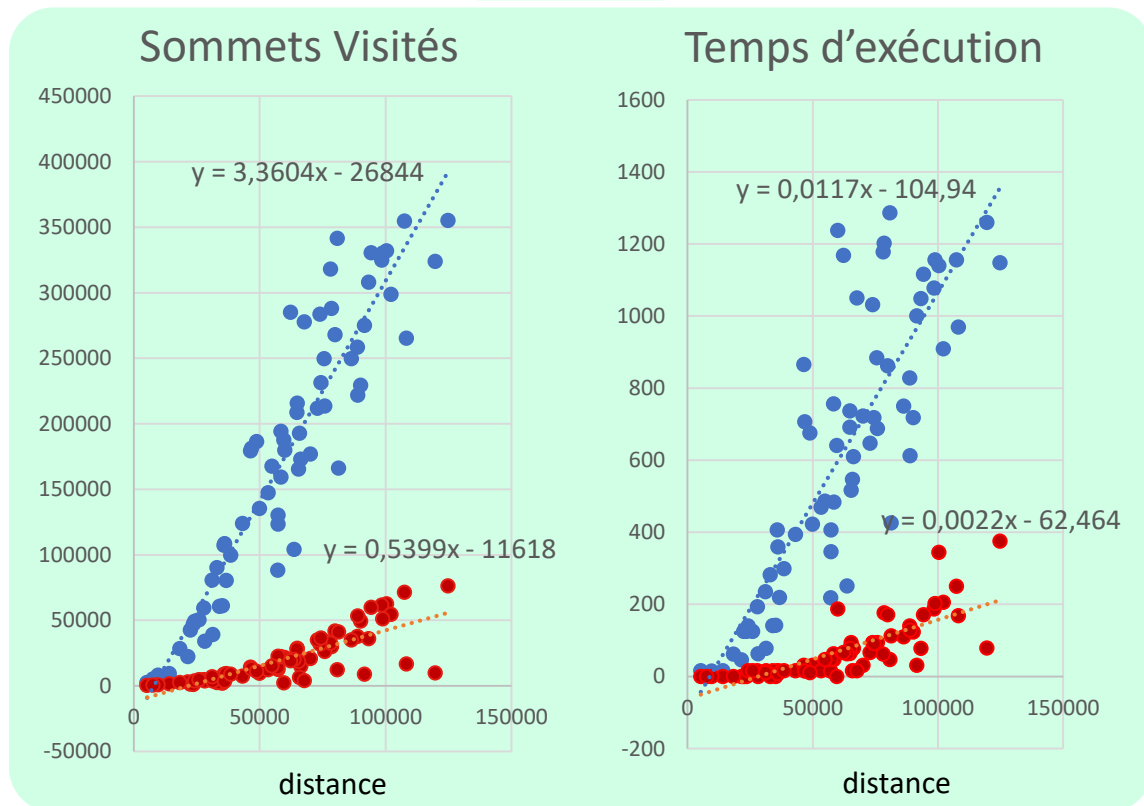


Carte non routière - comparaison

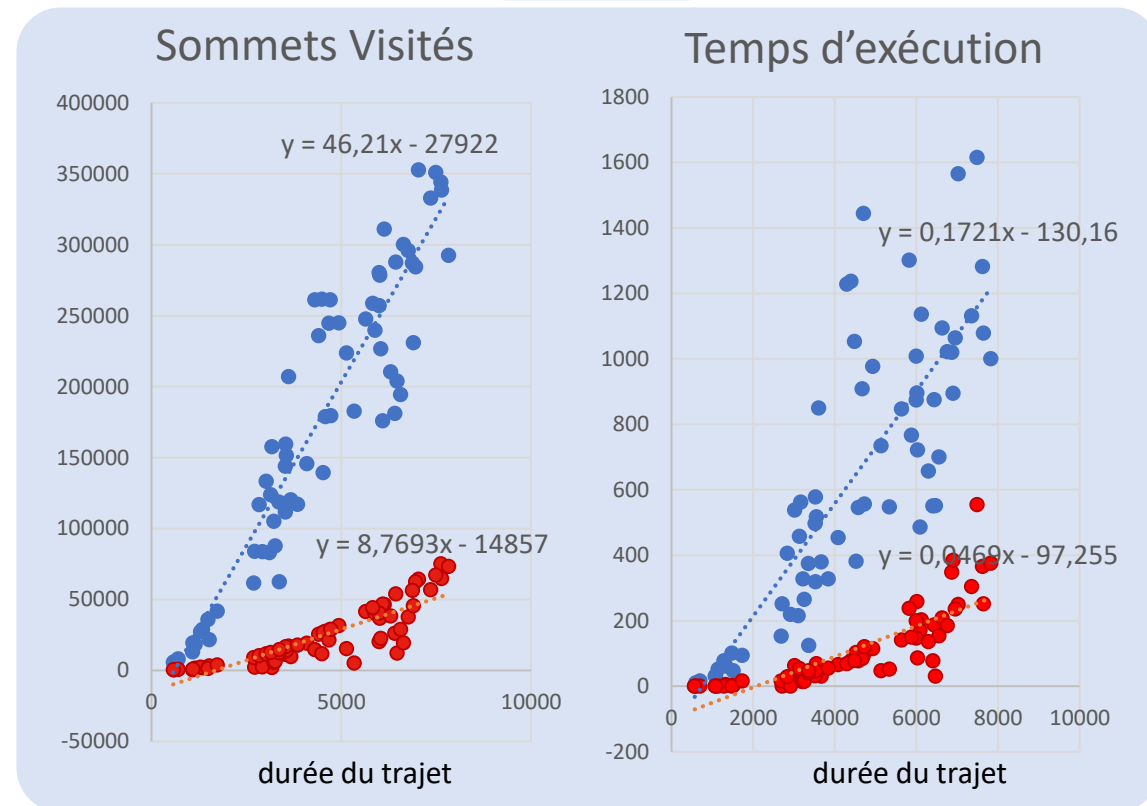
carre-dense.mapgr

Distance

Temps



■ Dijkstra



■ A*

Carte non routière - comparaison

carre-dense.mapgr

Distance

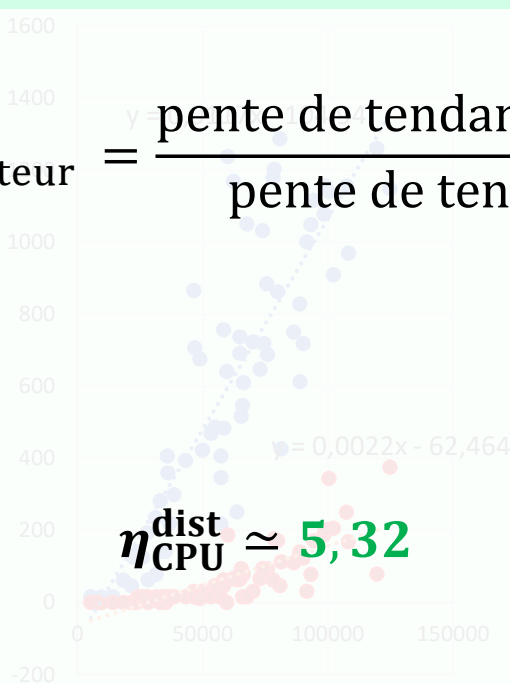
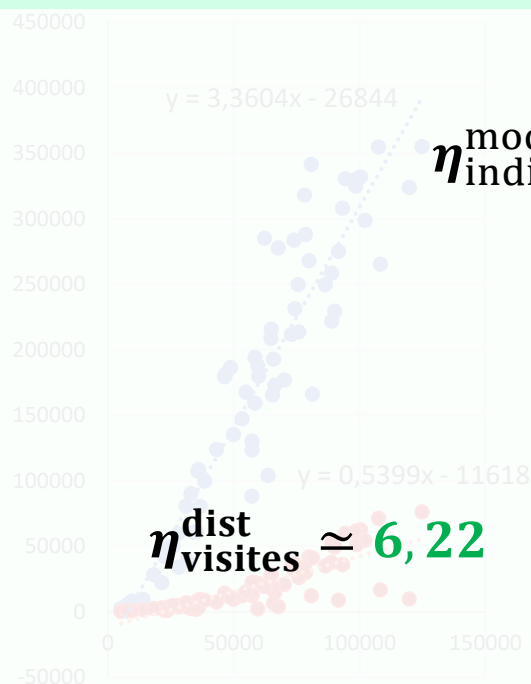
Temps

Sommets Visités

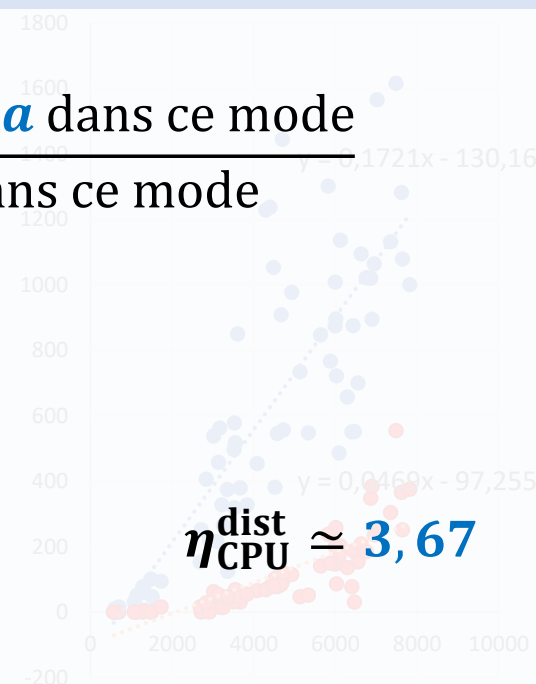
Temps d'exécution

Sommets Visités

Temps d'exécution



■ Dijkstra



■ A*

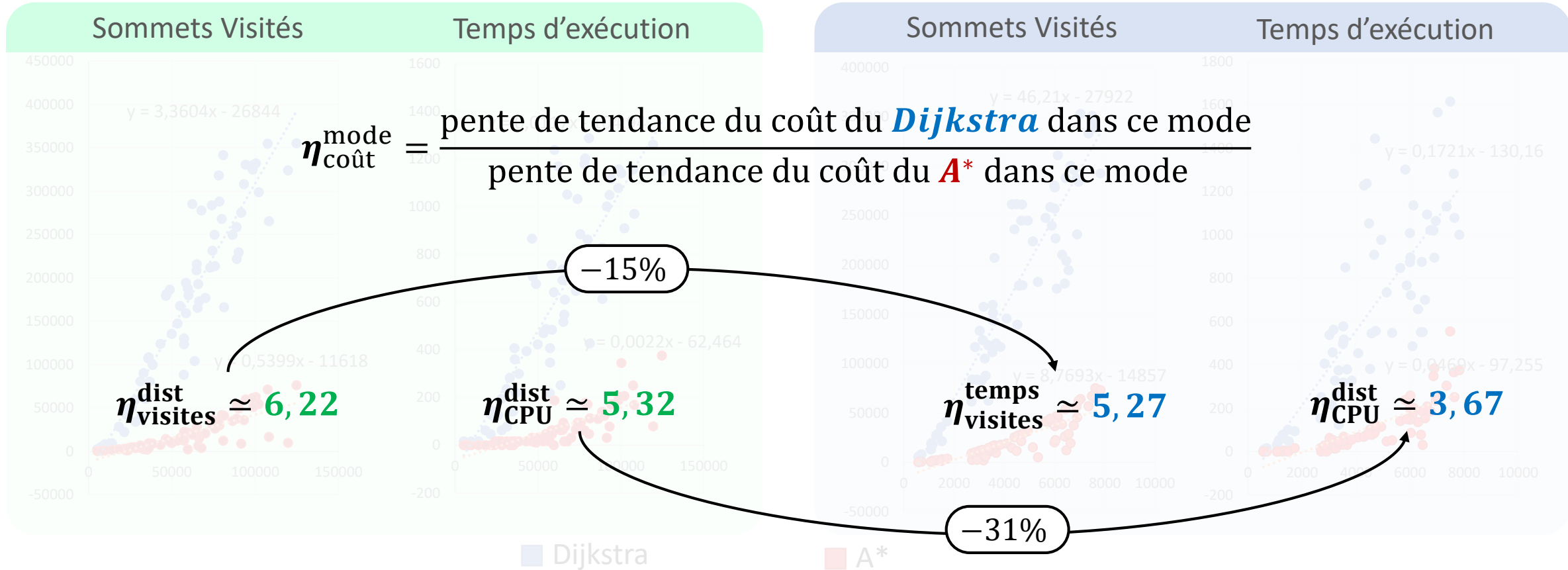
$\eta_{\text{indicateur}}^{\text{mode}} = \frac{\text{pente de tendance de l'indicateur en Dijkstra dans ce mode}}{\text{pente de tendance de l'indicateur en A* dans ce mode}}$

Carte non routière - comparaison

carre-dense.mapgr

Distance

Temps



Plan

- I. Introduction et contextualisation
- II. Tests de validité
- III. Tests de performances
- IV. Problème ouvert
- V. Conclusion

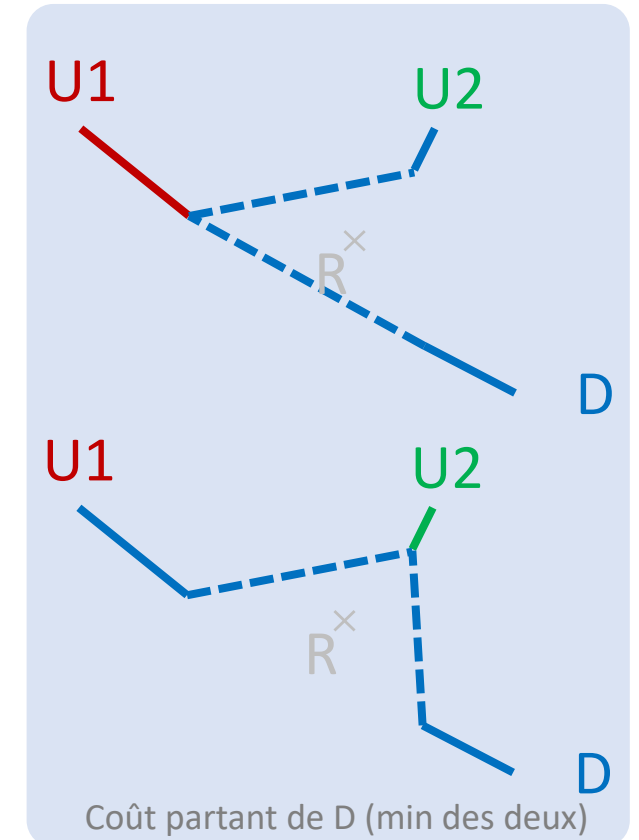
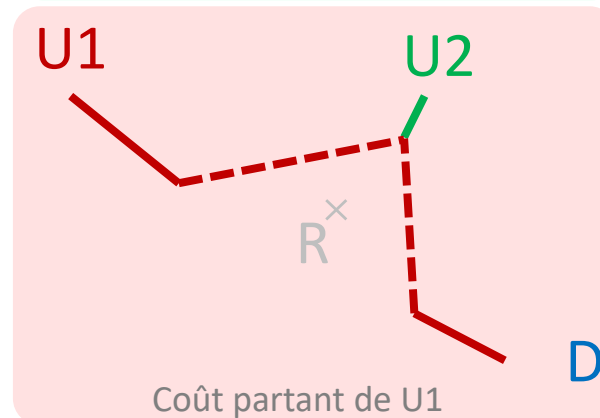
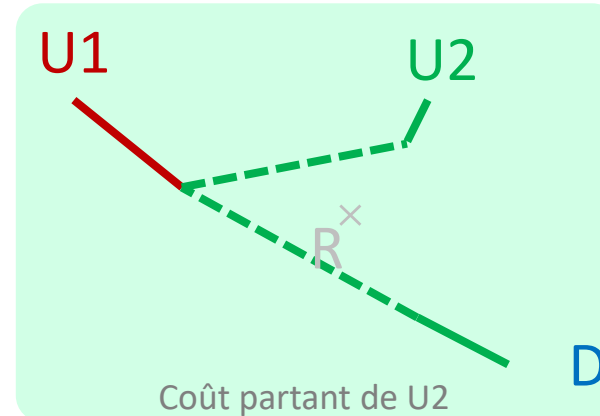
Problème ouvert : convoiturage

Solution proposée

- Lancer 3 variantes de A* dirigées vers leur destination en passant par le sommet courant de l'autre automobiliste
- Mettre à jour l'itération au coût minimal
- Arrêter lorsque un sommet R est marqué par les trois algorithmes

En fin d'algorithme, on a bien :

$$\left. \begin{array}{l} \min(U_2R + RD) \\ \min(U_1R + RD) \\ \min(U_1R + U_2R) \end{array} \right\} \Rightarrow \min(U_1R + U_2R + RD)$$

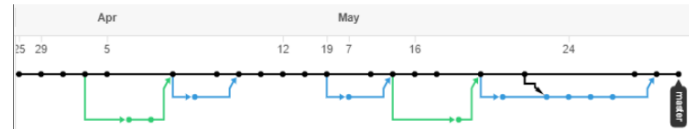
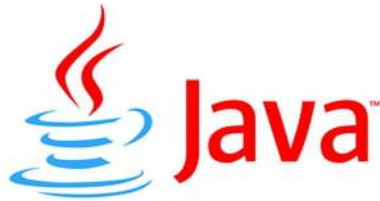


--- Distance à vol d'oiseau

Plan

- I. Introduction et contextualisation
- II. Tests de validité
- III. Tests de performances
- IV. Problème ouvert
- V. Conclusion

Conclusion

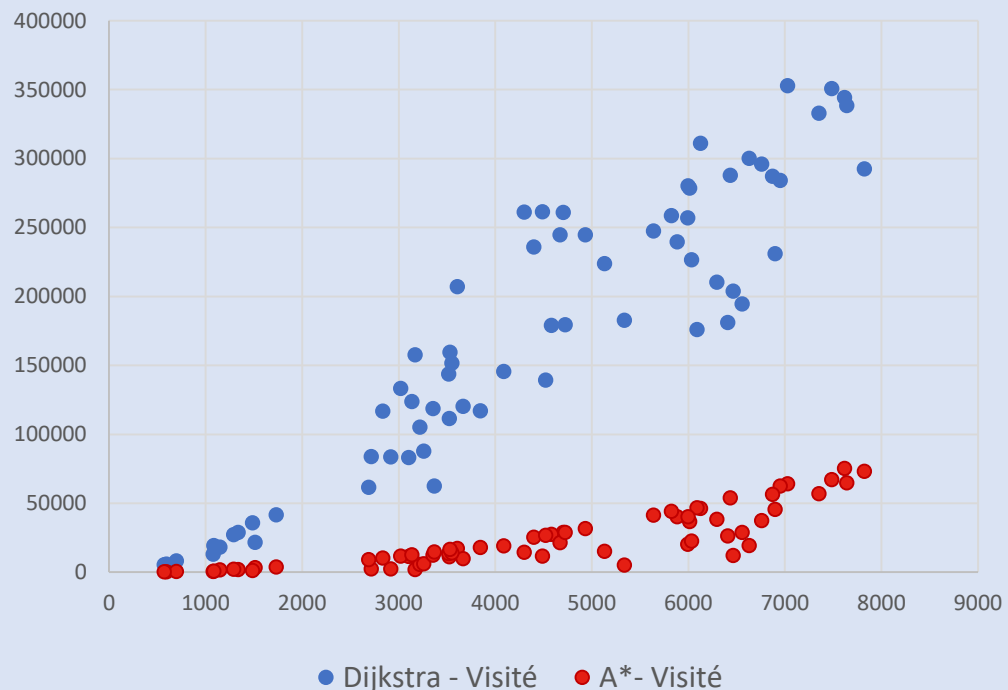


Carte non routière

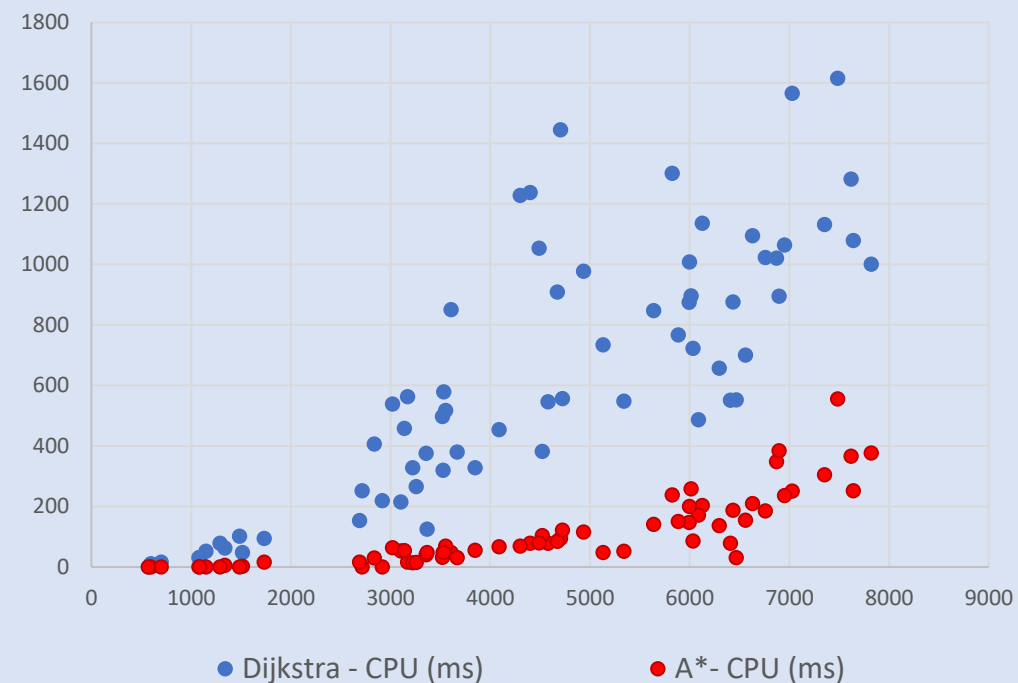
Temps

carre-dense.mapgr

Nombre de **sommets visités** selon la longueur **en s** du chemin le plus court

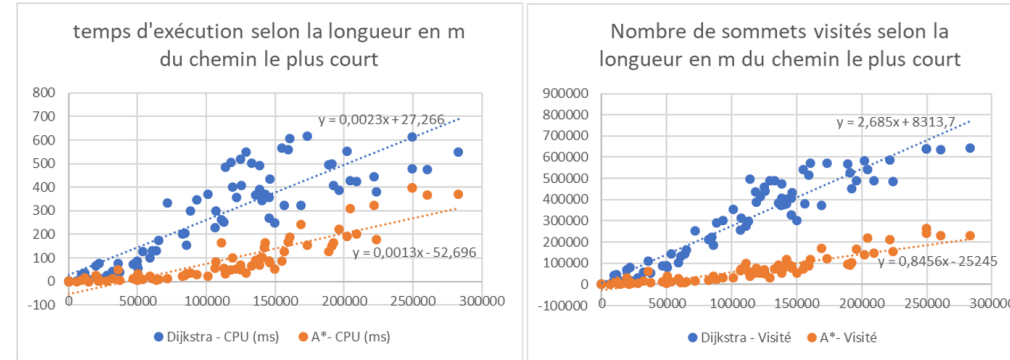


temps d'exécution selon la longueur **en s** du chemin le plus court

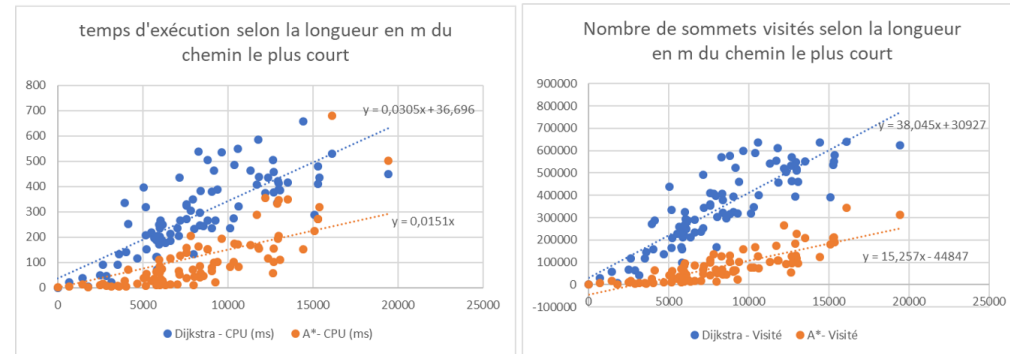


Carte routière - comparaison

bretagne.mapgr



$$\eta_{\text{CPU}}^{\text{dist}} = \frac{0,0023}{0,0013} \approx 1,77 \quad \text{et} \quad \eta_{\text{visites}}^{\text{dist}} = \frac{2,285}{0,8456} \approx 3,18$$



$$\eta_{\text{CPU}}^{\text{temps}} = \frac{0,0305}{0,0151} \approx 2,01 \quad \text{et} \quad \eta_{\text{visites}}^{\text{temps}} = \frac{2,285}{0,8456} \approx 2,49$$

Problème ouvert

Sommet **marqué** par Dijkstra



$\text{coût}_{\text{sommet}} = \text{distance minimale avec l'origine}$

Solution proposée

- Lancer un Dijkstra à partir de U1, U2, et D
- Arrêter la recherche dès qu'un sommet R est marqué par les trois algorithmes

Au sommet R, on a :

$\min(U_1R) \ \& \ \min(U_2R) \ \& \ \min(DR)$

Or nous voulons :

$\min(U_1R + U_2R + DR)$

