

Soutenance be graphe

NOZAHIC Morvan PELOUS Thomas 3 MIC - E





Plan:

- Implémentation
- Tests de validité
- Tests de Performances
- Problème ouvert: Covoiturage

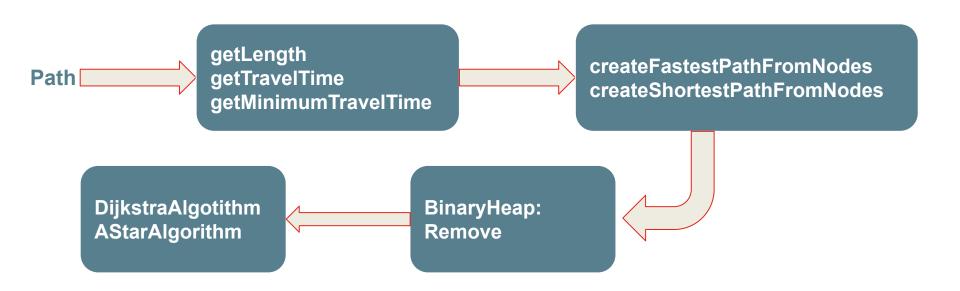
Conclusion





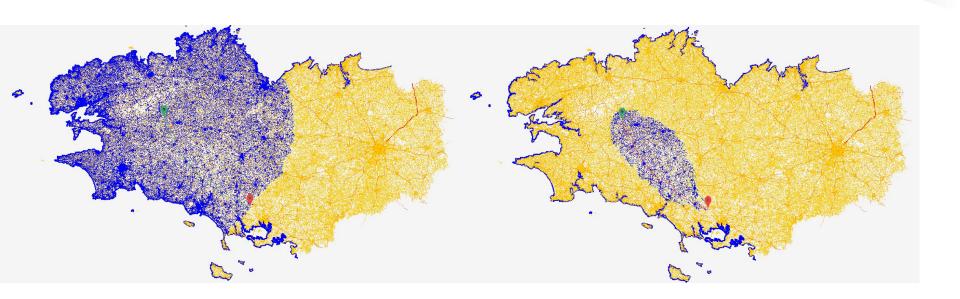
Implémentation

Algorithmes préliminaires:





Implémentation







Tests de validité

Avec oracle: Algorithme de Bellman-Ford

Sans oracle:

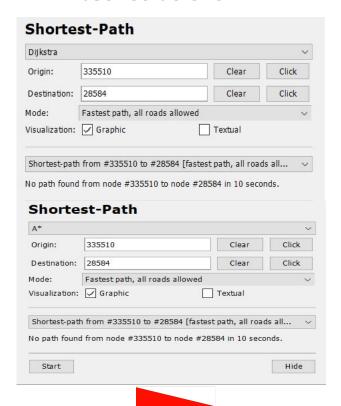
- 1. Absence de chemin
- 2. coût nul
- 3. Carte simple





Tests de validité

1. Absence de chemin



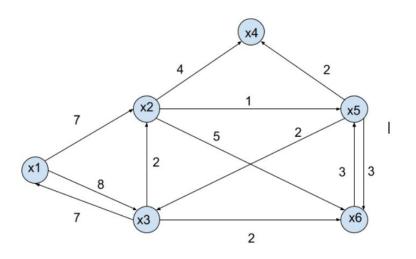
2. Coût nul

Dijkstra			`				
Origin:	335510 Clear		Click				
Destination:	335510	Clear	Click				
Mode:	Fastest path, all roads allowed						
Visualization:	✓ Graphic Textual						
Shortest-path	from #335510 to #3355	10 [fastest path, all ro	ads a 🔻				
Found a path	from node #335510 to no	ode #335510, 0.0000 n	ninutes i				
			Hide				
Start							
			niue				
Shorte	st-Path		nide				
Shorte	st-Path		Hide				
	st-Path	Clear	Click				
A*		Clear	,				
A* Origin: Destination:	335510	Clear	Click				
A* Origin: Destination: Mode:	335510 335510 Fastest path, all roads a	Clear	Click				
A* Origin: Destination: Mode: Visualization:	335510 335510 Fastest path, all roads a	Clear	Click				
A* Origin: Destination: Mode: Visualization:	335510 335510 Fastest path, all roads a	Clear	Click				
A* Origin: Destination: Mode: Visualization: Shortest-patt	335510 335510 Fastest path, all roads a	Clear llowed Textual 510 [fastest path, all ro	Click Click				
A* Origin: Destination: Mode: Visualization: Shortest-patt	335510 335510 Fastest path, all roads a Graphic from #335510 to #3355	Clear llowed Textual 510 [fastest path, all ro	Click Click				





3. Carte simple

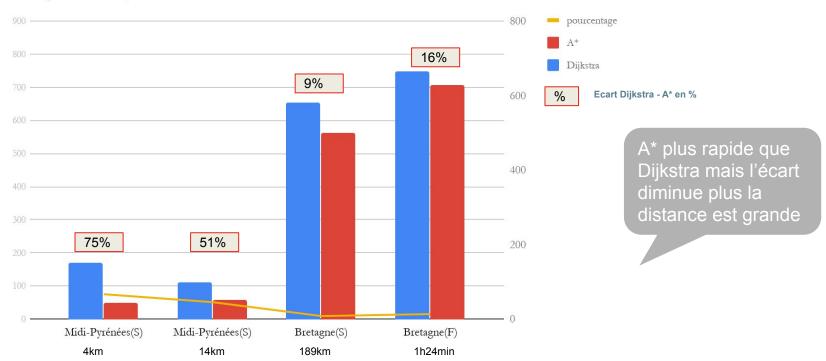


	x1	x2	х3	х4	x5	х6
x1	ı	7,x1	8,x1	10,x5	8,x2	10,x3
x2	10,x3	-	3,x5	3,x5	1,x2	4,x5
х3	7,x3	2,x3	1	5,x5	3,x2	2,x3
x4	8	8	8	-	8	8
х5	9,x3	4,x3	2,x5	2,x5	-	3,x5
х6	12,x3	7,x3	5,x5	5,x5	3,x6	-



Tests de Performances

Comparaison Dijkstra - A*

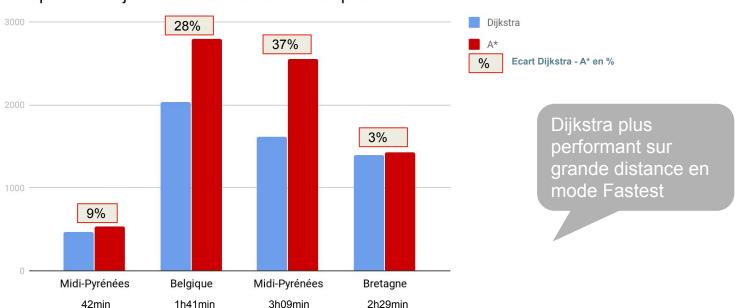




Tests de Performances

Quelques tests sur de plus longues distances:

Comparaison Dijkstra - A* : mode Fastest uniquement

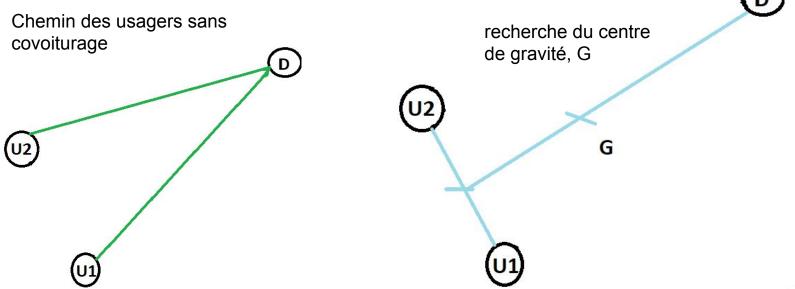




Problème ouvert: Covoiturage

Trouver le centre de gravité du triangle formé par les deux usagers, U1 et U2, et la destination, D

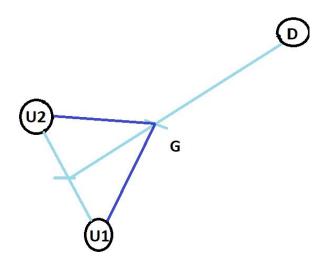
La méthode nécessite 7 algorithme de Dijkstra ou A*





Problème ouvert: Covoiturage

Chemin à partir du centre de gravité, G



Comparaison à la solution sans covoiturage

D

G



Conclusion

Implémentation d'algorithmes de plus court chemin

Mise en place de tests de validité et performances

• Etude d'un problème ouvert pour mettre en application nos algorithmes



MERCI POUR VOTRE ATTENTION