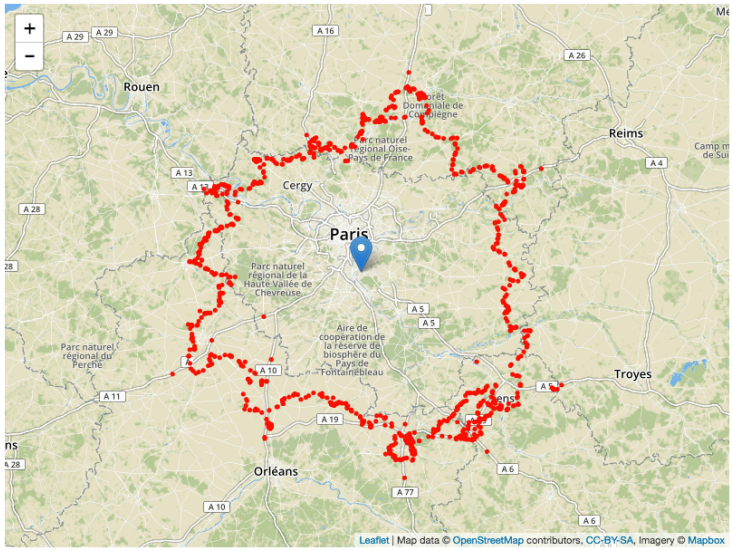
Shortest Path Trees and Reach in Road Networks

Louis Abraham & Sayuli Drouard

Our subject: shortest paths

- Given a starting point, where do ***t* hours** on a **shortest path** bring you ?
- What if your **destination** is at least ***t'* hours** away ?

- Study of the notion of **reach** (option B)



Method for Part I

- Given a starting point, where do ***t* hours** on a **shortest path** bring you ?

} Dijkstra implemented with binary

- What if your **destination** is at least ***t'* hours** away ?

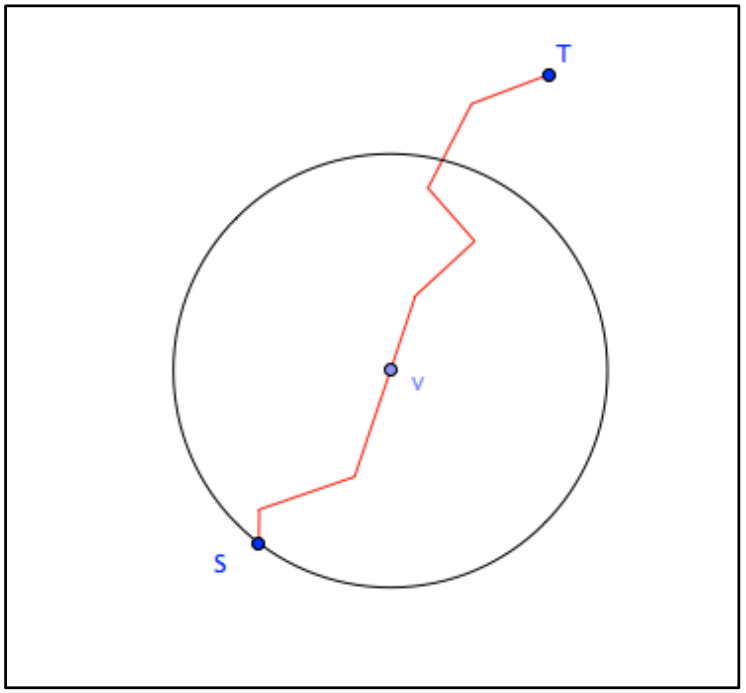
} Compute shortest paths using Dijkstra, then select vertices further away than *t'*, and backtrack to a *t* distance

Optimization

- **Dijkstra stop**
- **Dijkstra memoization**
- **Distance sorting**
- **Online Dijkstra**

Part II (Option B)

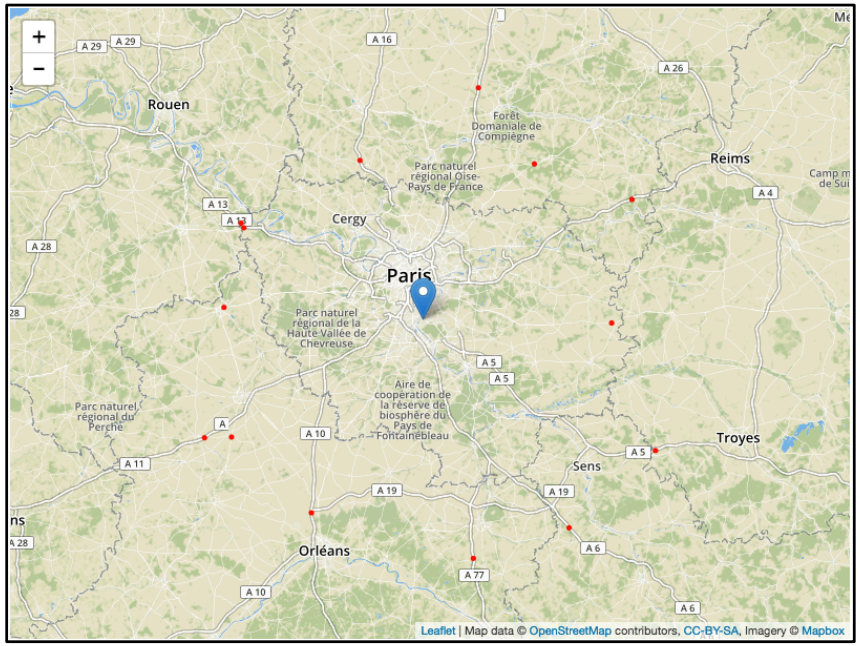
The notion of reach

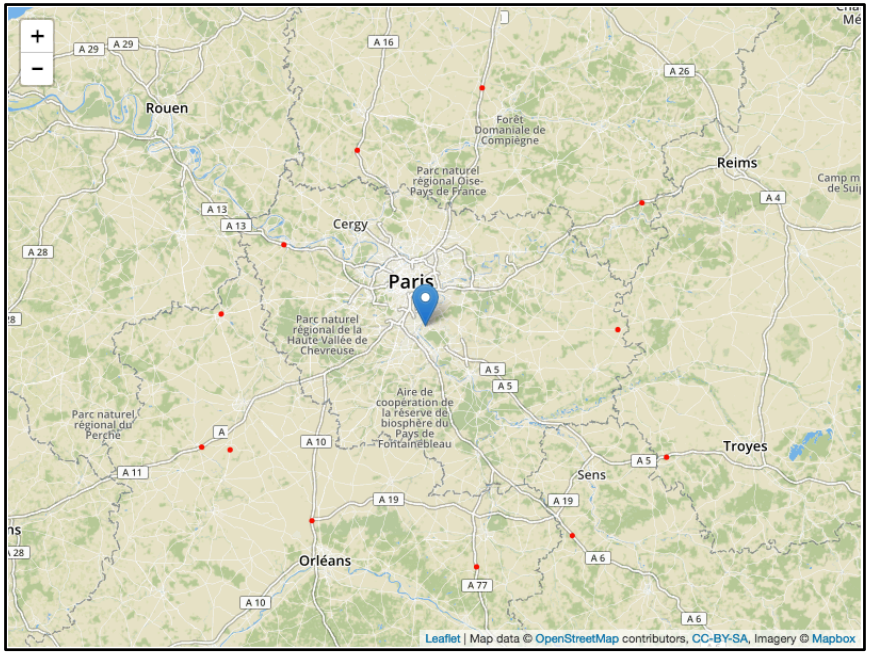


Part II (Option B)

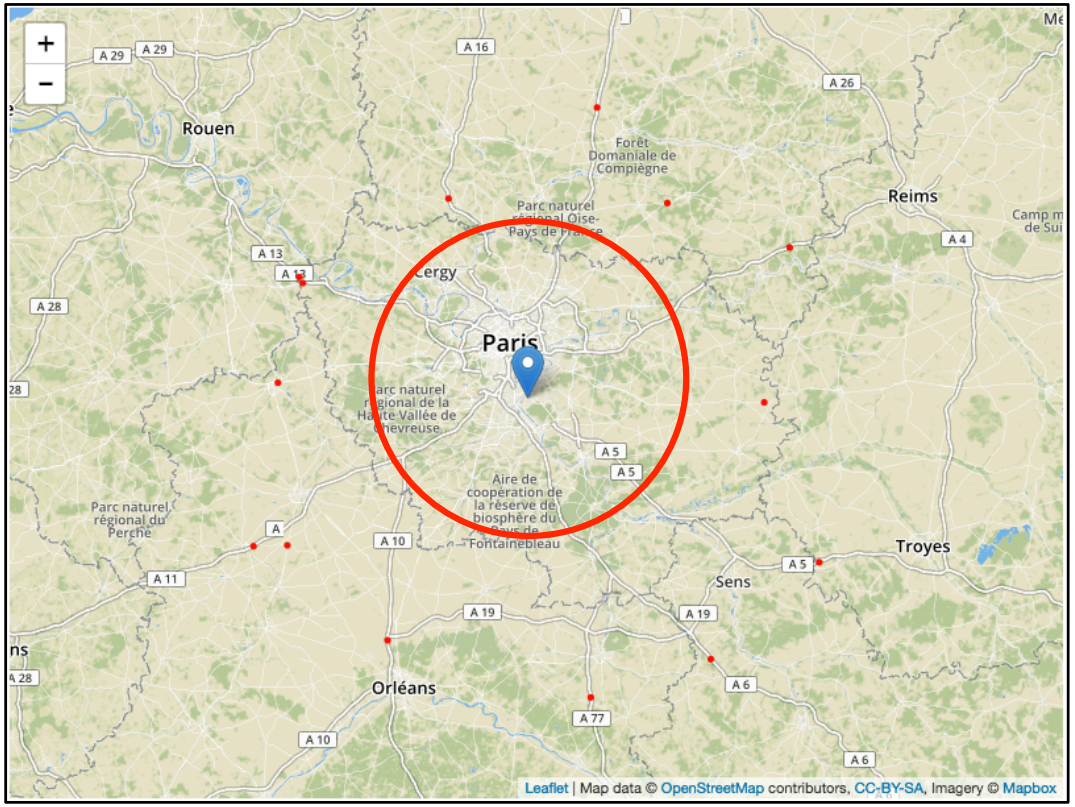
T_{out} = set of points from problem (1.3)

S_{in} : same definition as T_{out} with the converse graph





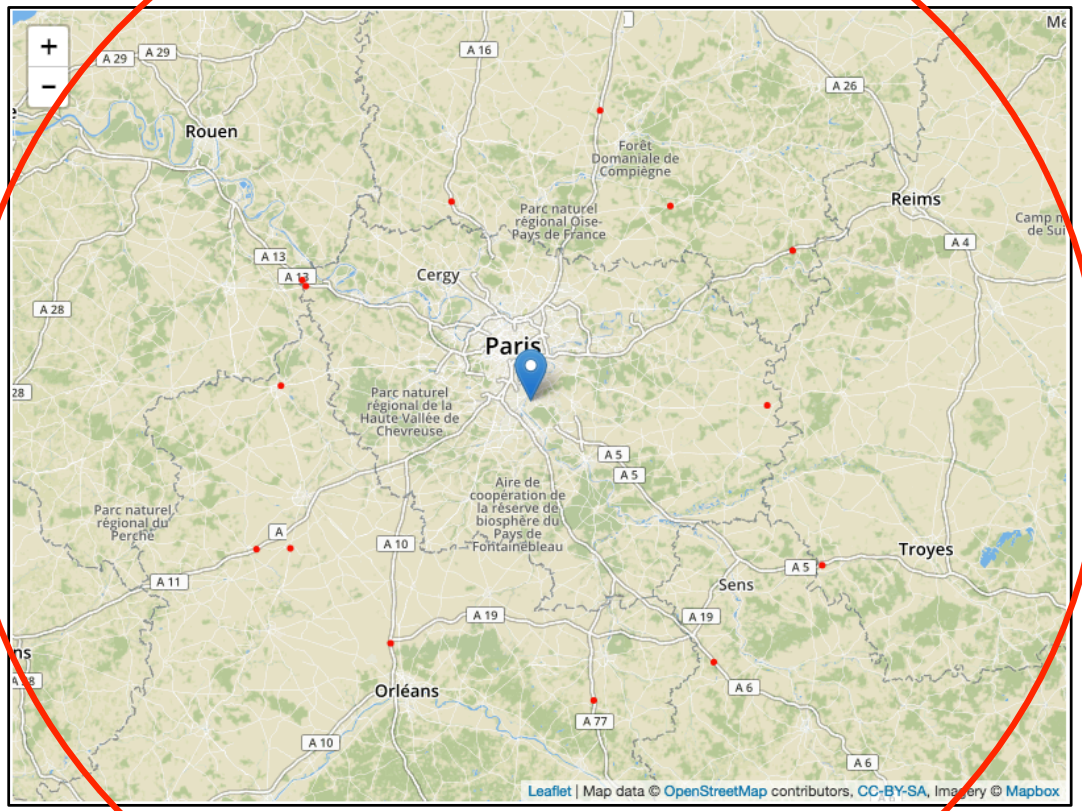
Part II (Option B)



If reach(v) < 1 hour :

NO shortest path joins S_{in} and T_{out} through v

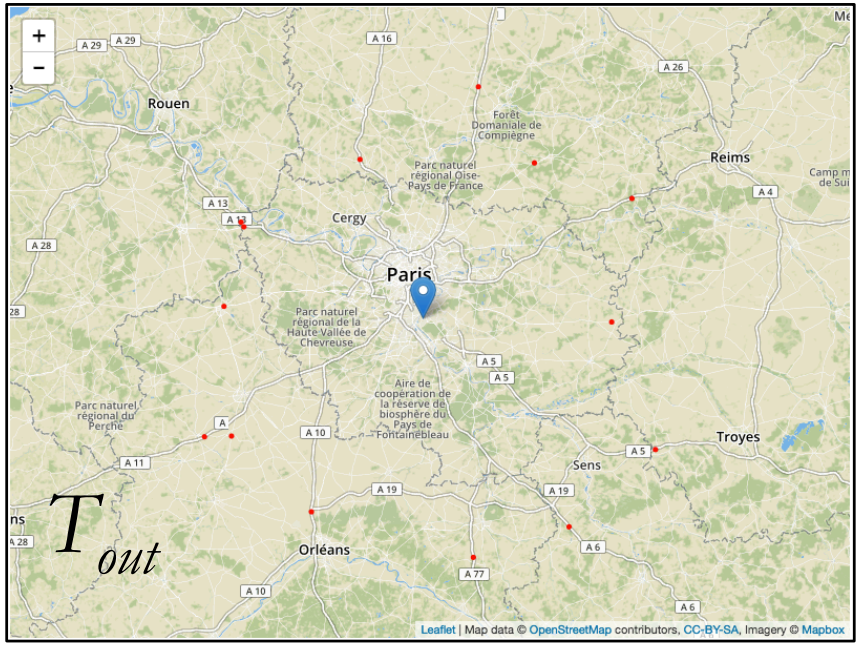
Part II (Option B)

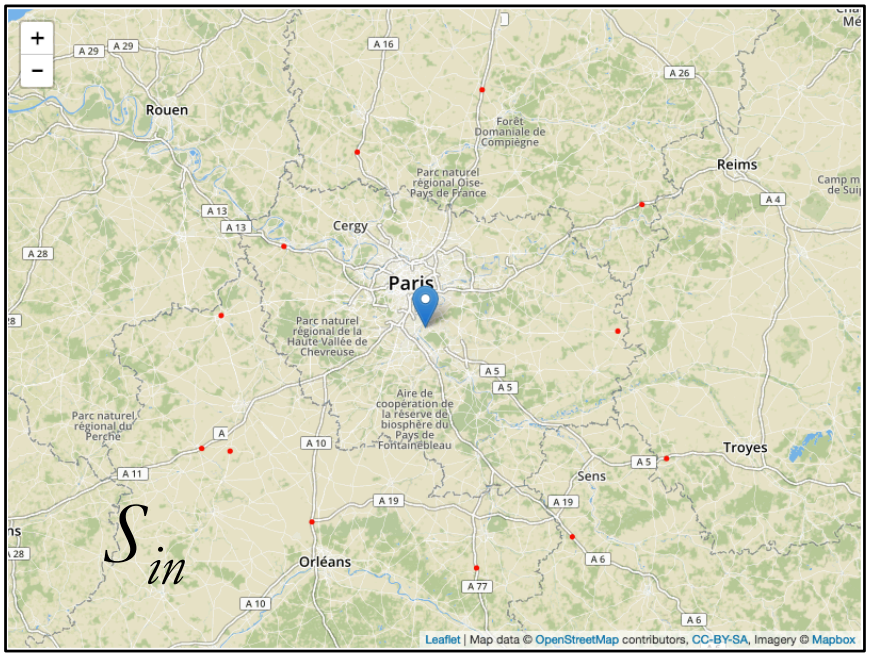


If reach(v) ≥ 2 hours :

There MUST be a shortest path joining S_{in} and T_{out} through v

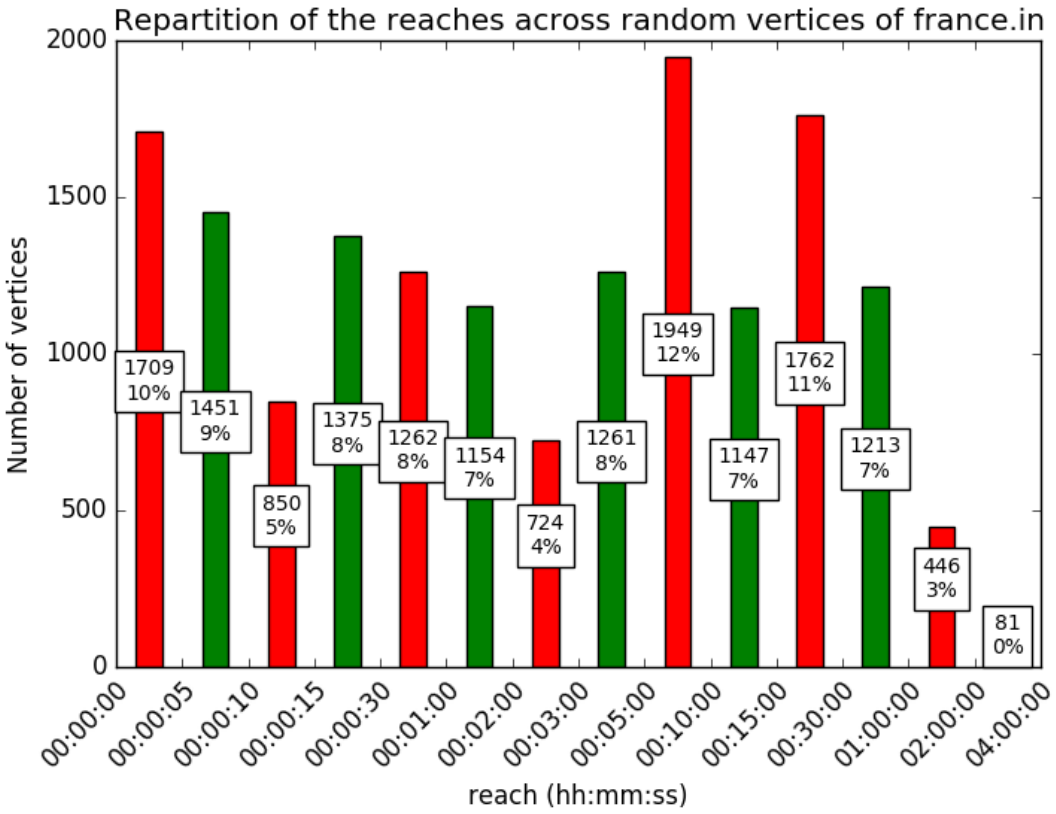
Part II (Option B)



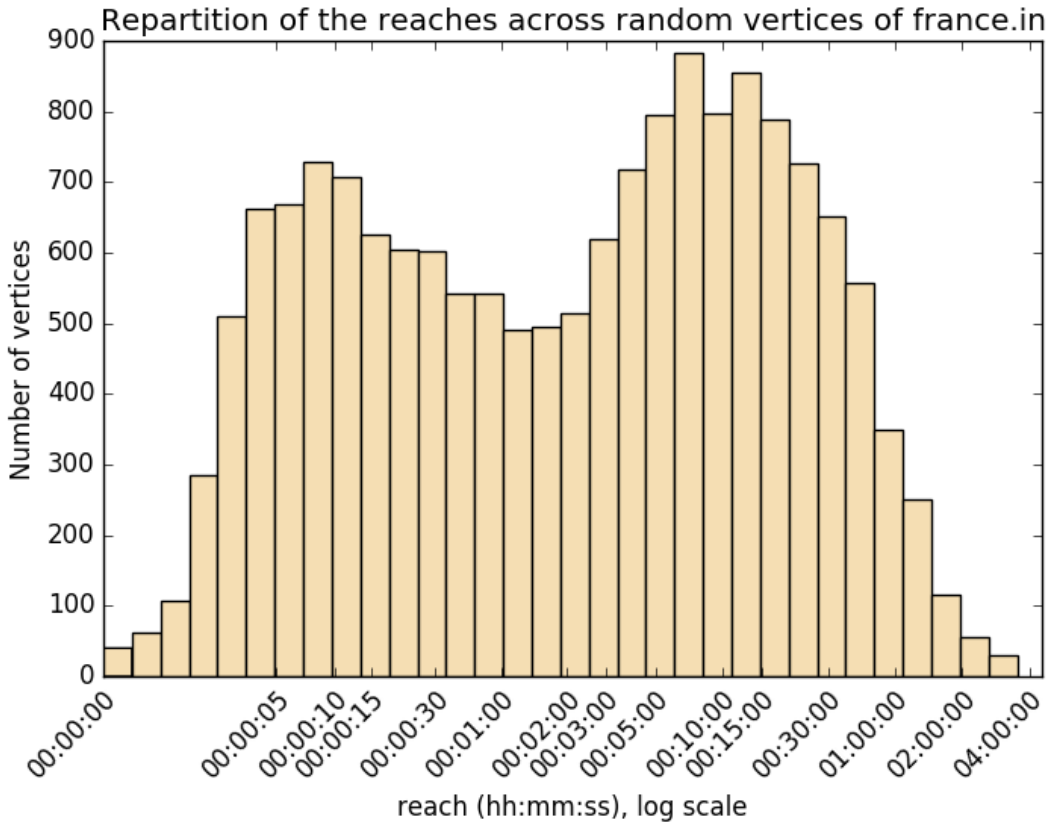


The algorithm: estimating **reach(*v*)**

Reach in France



Reach in France



Thank you.