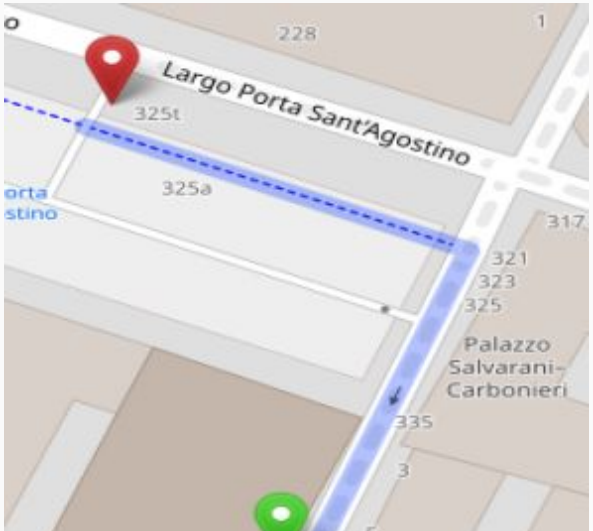
A Case Study on Problematic Routes between Two Points in the Map of Modena City

Bilel Arfaoui
Ingegneria Informatica Unimore

The problem in routing

An issue arose while utilizing the APOC-provided A* algorithm to determine the optimal route between two points in our graph during route optimization investigation.

Expected Output



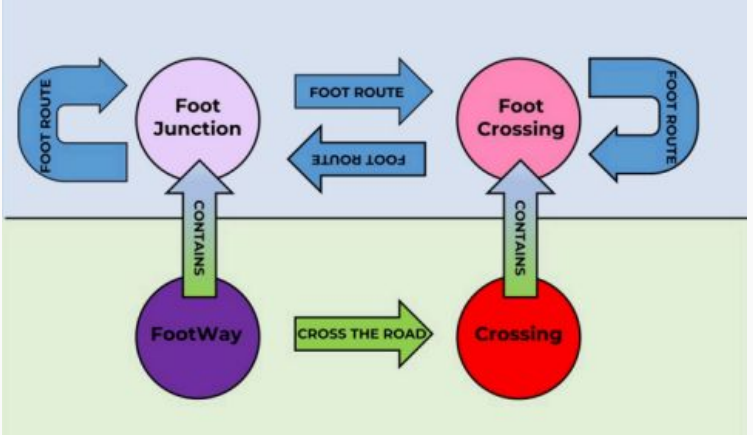
Actual Output

Structural Analysis of the Graph

The first thing to check is the structural integrity of our Graph such as assensing that the attributes have the correct values or making sure that the road network is complete.

With this analysis we found out that:

- The vast majority of edges have the attribute danger set to 3
- The road network does not seem to have missing relationships
- The edges all have a highway attribute which indicates that pedestrian can cross the relationship
- The distance cost have all sensible values

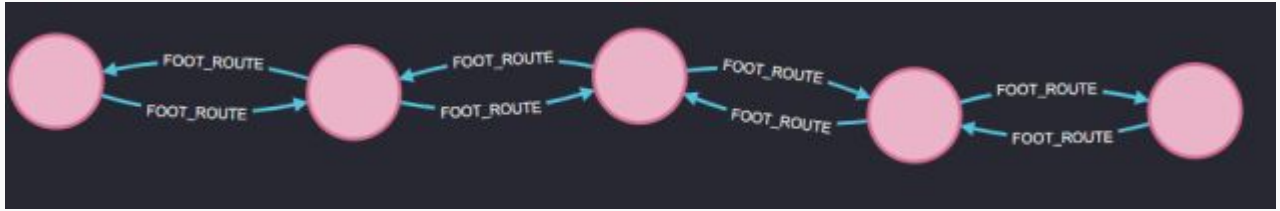


Algorithm execution

After assessing that there are no structural anomalies, we use will use algorithms offered by Neo4j and in particular APOC, using the attribute “distance” as weight, in the following order:

- shortestPath
- Dijksta
- A*

All algorithms have given the same path as output



Routing with the assistance of APIs

After using the routing algorithms, the next step is using a python script that fetches the 2 OSM points and give them to the following APIS to check if the results are consistent with the local algorithms.

The APIs used are:

- Graphhopper
- OSRM
- Valhalla

Differences between routing APIs

Valhalla

- Multi modal routing: optimized for routing with different ways of public transport
- Transit level graph
- Slope based routing for pedestrians and cyclists
- Complex data management such as detailed data about public transport timetables and frequencies

GraphHopper

- Vehicle profile personalization for different needs in routing
- Offline routing
- Real time traffic data integration
-

OSRM

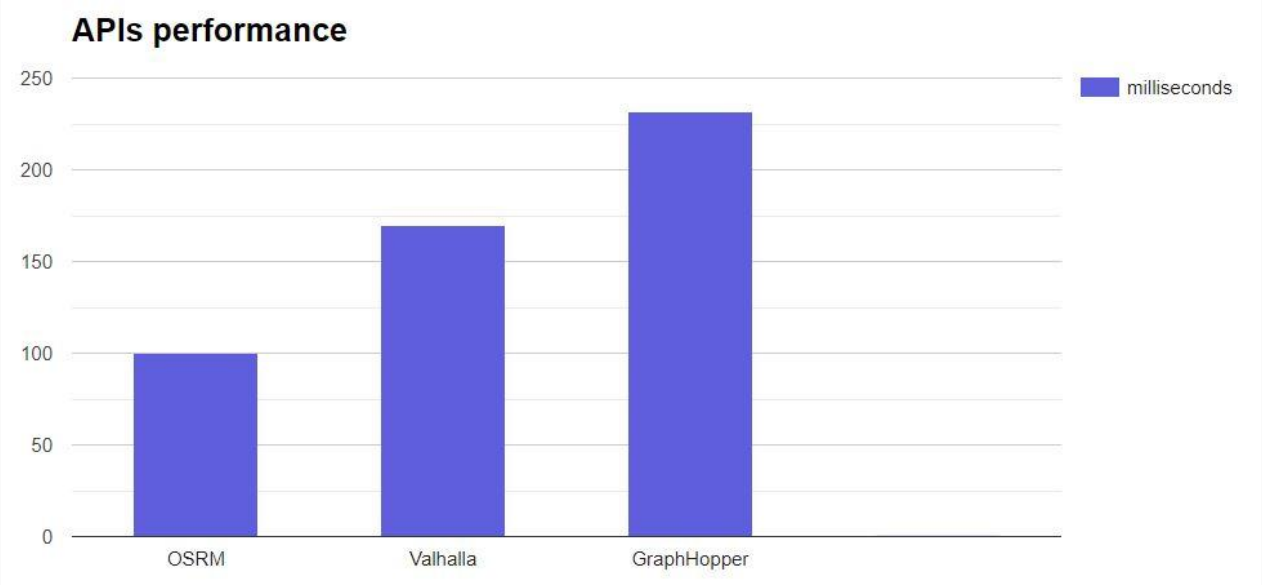
- Usage for statistical graph to speed up the routing procedure
- Graph preprocessing with PBF(Protocol Buffer Format) files, optimized for routing and storage of geographical data
- Convenient for web and mobile applications thanks to its speed
- Basic routing for pedestrians, cyclists and other vehicles with no need for complex data

Comparing results

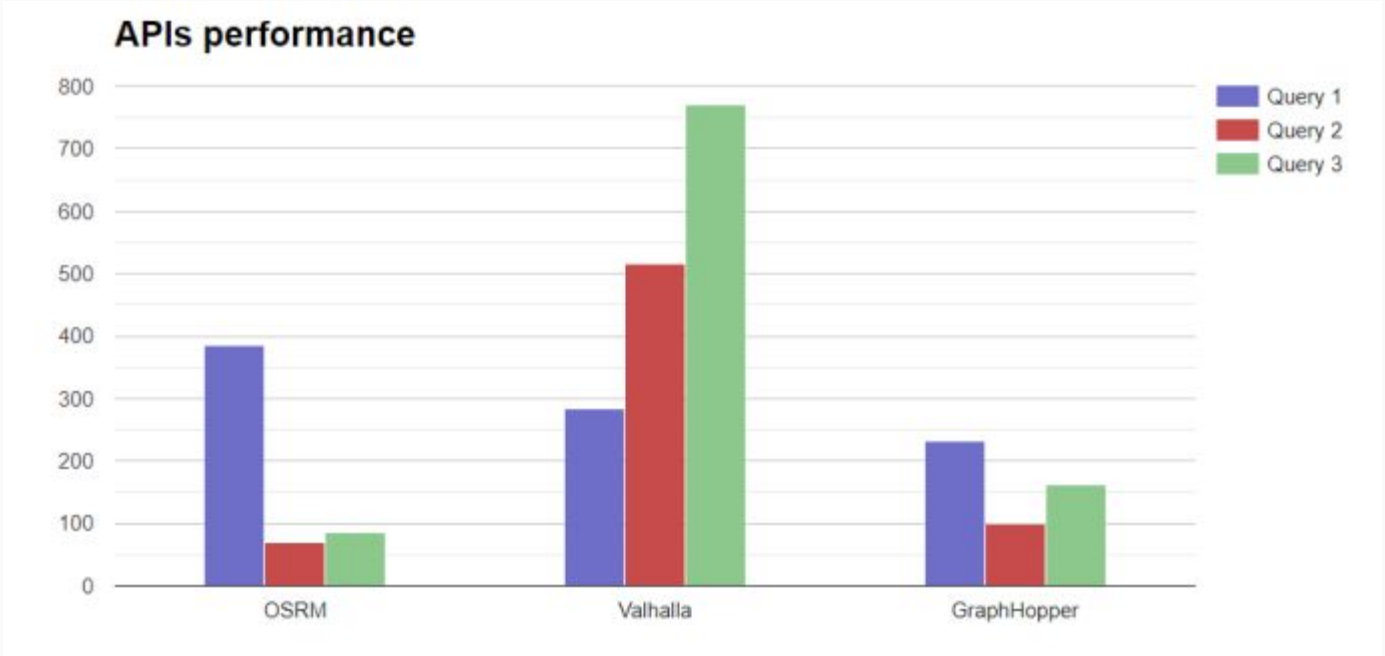
	Valhalla	OSRM	GraphHopper	ShortestPath	Dijkstra	Astar
distance(m)	85	85	85	91	91	91

The result given by APIs are consistent with each other and show a discrepancy with our algorithms, but it’s a negligible amount.

Further results of our APIs



Comparing the routing APIs with random queries



Conclusions

In conclusion, after extensive analysis, the graph database exhibits no anomalies. The routing remains unchanged even when considering the direction or changing the weight from distance to duration. I was unable to replicate the reported error using the provided code. Moreover, routing with OSRM, Valhalla, and GraphHopper consistently aligns with local graph routing using both Dijkstra and A* algorithms. Therefore, it is likely that the routing issue stems from an abnormal loading or import process.