

Carpooling pick-up order finder

Yhoan Alejandro Guzmán García

Juan Sebastián Pérez Salazar

Medellín, May 16

Data Structures

Matrix of Arcs

		<div><div>Arc</div><div>Stored data:</div><ul style="list-style-type: none">- Source Node (node object)- Destination Node (node object)- Time (minutes)- Distance (meters)</div>	

ClassArc
Stored data:

- Source Node (node object)
- Destination Node (node object)
- Time (minutes)
- Distance (meters)

Class Node

Stored data:

- ID (int)
- Latitude (float)
- Longitude (float)
- Description (String)
- Passengers ([])
- Picked-up (boolean)

Figure 1: Matrix of Arcs. An Arc contains source and destination nodes, time and distance. A node has an ID, coordinates (latitude and longitude), description, passengers and a Boolean that indicates if it is picked-up or not.

Algorithm design criteria

We decided to use a matrix to store the graph because we have a complete graph. In the matrix we store Arcs. The arcs have the distance, the weight (time), the source node and the destination node. This helps out with the ordering definition part of the algorithm because we can access the information in $O(1)$ complexity.

If we approach this problem with a brute force algorithm the complexity would be $O(2^n)$ which for $n = 205$ is equivalent to $5.1422017e+61$. This is a ridiculously big number. Thus, we decided to do a greedy algorithm.

Algorithm and Complexity

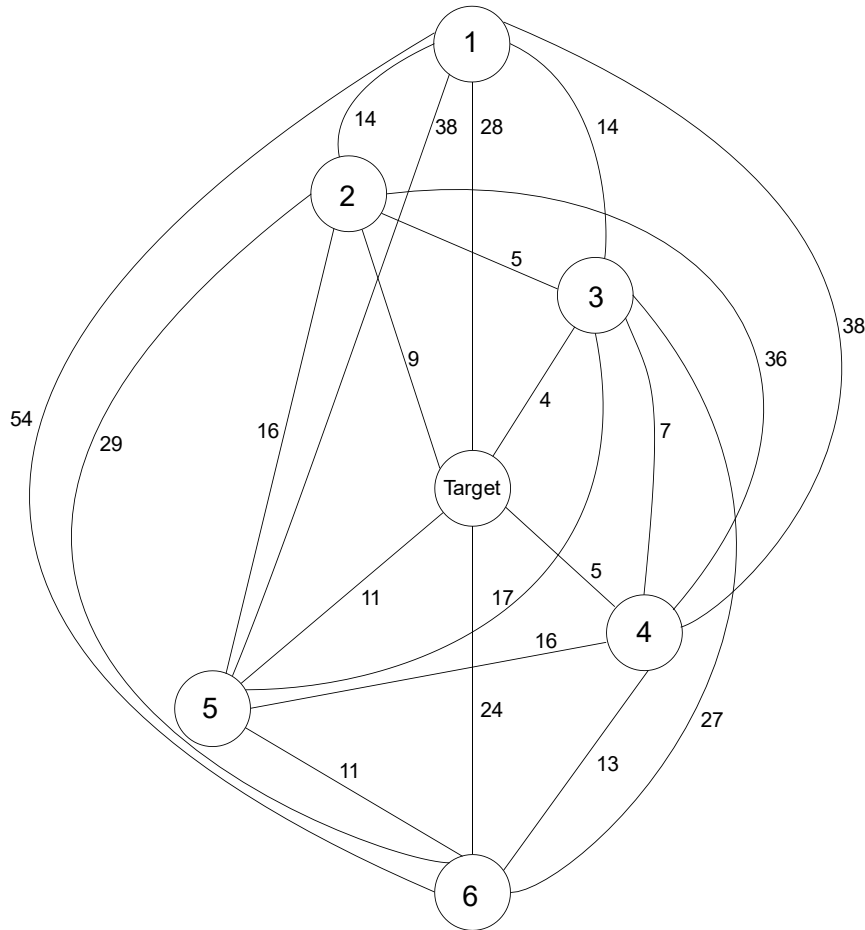


Figure 2: Graph that represents people and routes.

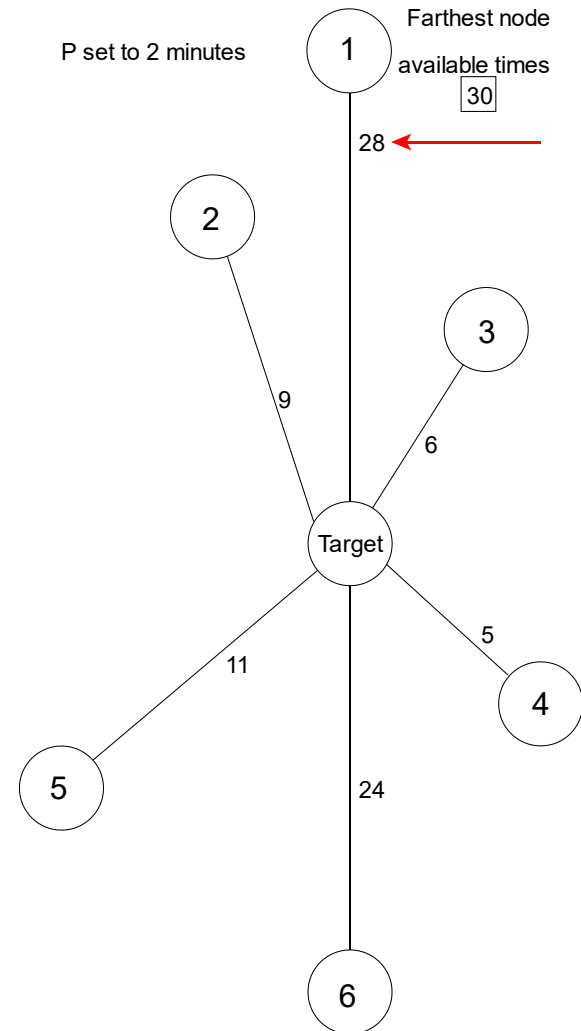


Figure 3: Calculating the farthest Node

Algorithm and Complexity

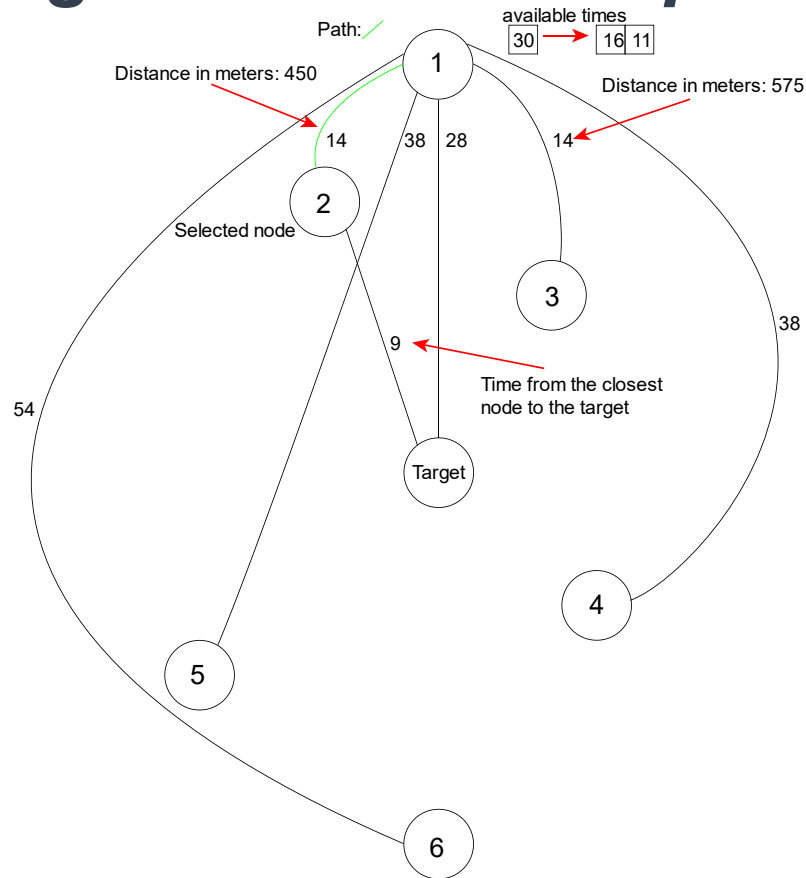


Figure 4: Select a passenger that meets all the conditions.

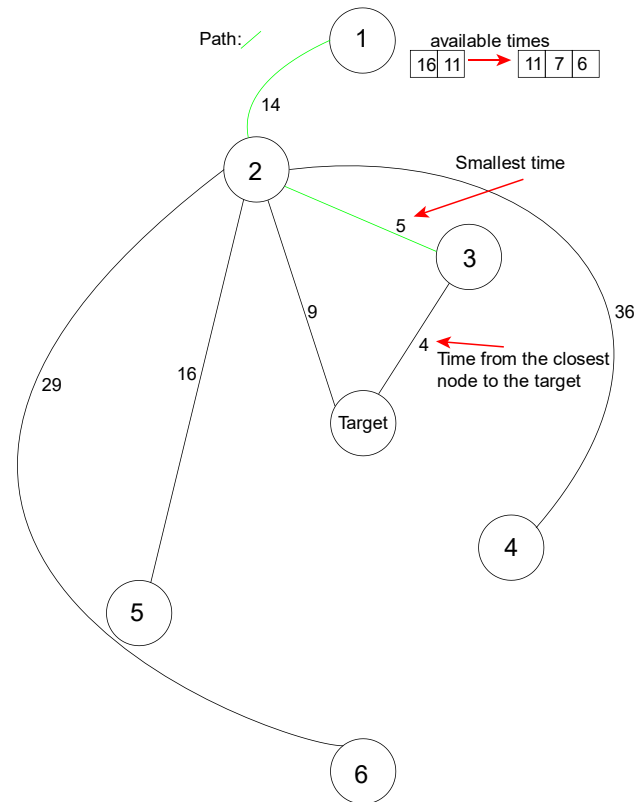


Figure 5: Search the next passenger and select it

Algorithm and Complexity

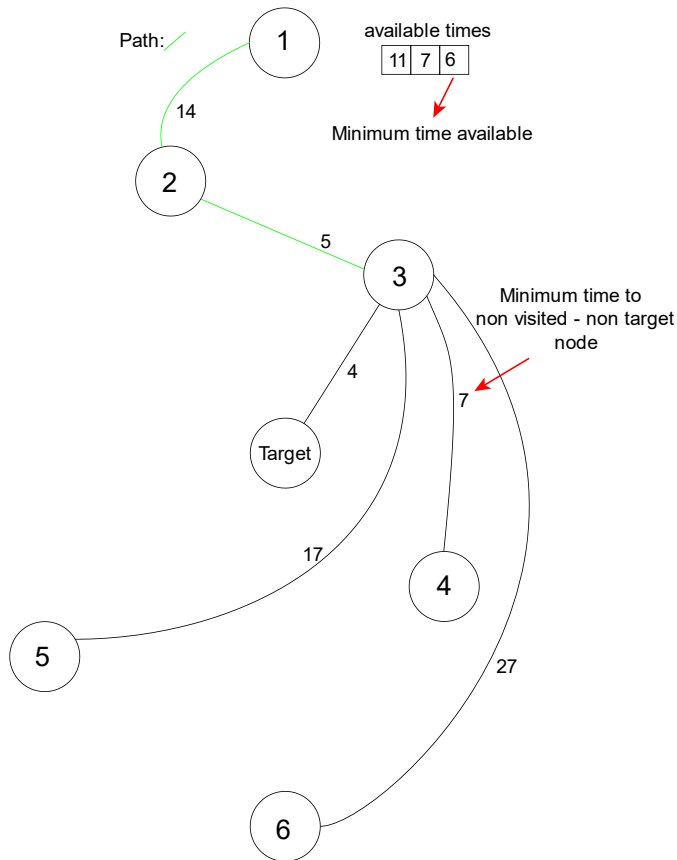


Figure 6: A Node that fails the conditions.

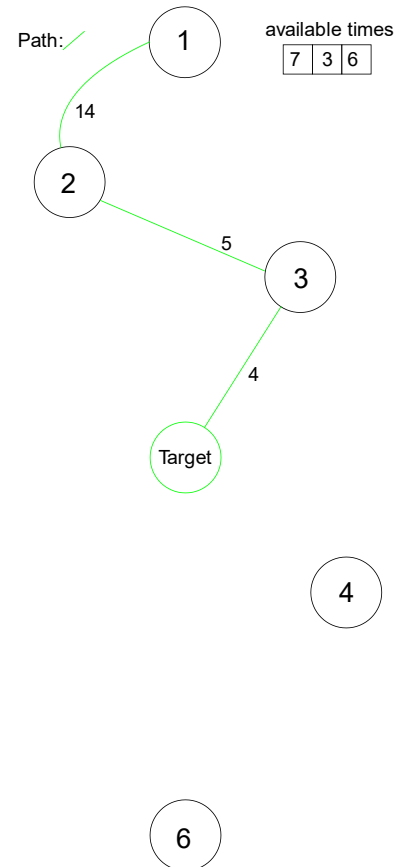
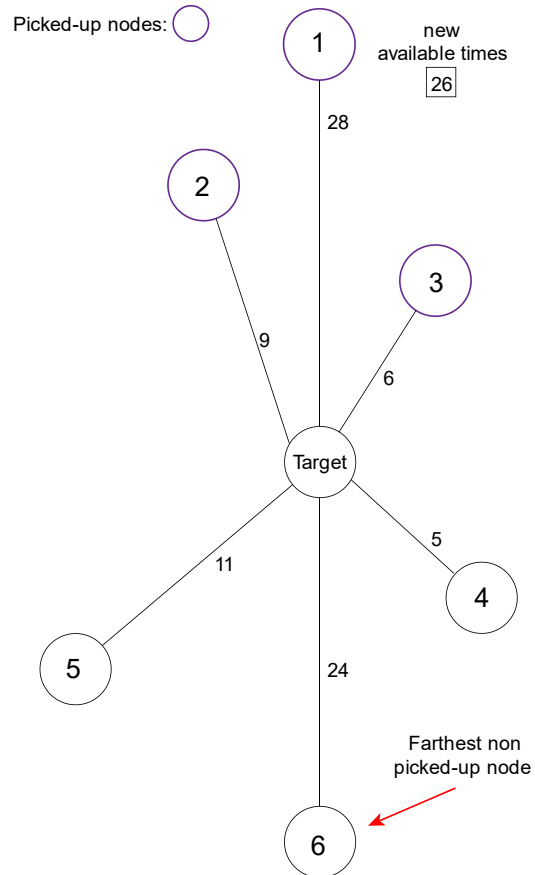


Figure 7: Path from the selected Node to the target

Algorithm and Complexity



Method	Complexity
Read graph (Read file)	O(lines)
Distance between nodes	O(1)
Getters	O(1)
Setters	O(1)
Get target distance	O(nLog(n))
Order Finder	O(n²Log(n))
Get distance from-to sorted	O(nLog(n))
Show answer (Write file)	O(n)

Table 1: Complexity analysis.

Figure 8: Repeating the process with all the Nodes that are not picked-up.

Time and Memory Consumption

Time

<div>Methods Datasets</div>	Read Graph (ms)	Get Target Distance (ms)	Order Finder (ms)	Show Answer (ms)	Total (ms)
U=4 P=1.2	0	0	0	0	0
U=4 P=1.7	0	0	0	0	0
U=11 P=1.1	0	0	0	0	0
U=11 P=1.2	0	0	0	0	0
U=11 P=1.3	0	0	0	0	0
U=205 P=1.1	228	0	47	0	275
U=205 P=1.2	213	0	50	0	263
U=205 P=1.3	223	0	37	0	260
U=205 P=5	221	0	35	0	256

Table 2: Execution time of the operations of the data structure for each data set and method.

Time and Memory Consumption

Memory

*P does not intervene in memory consumption

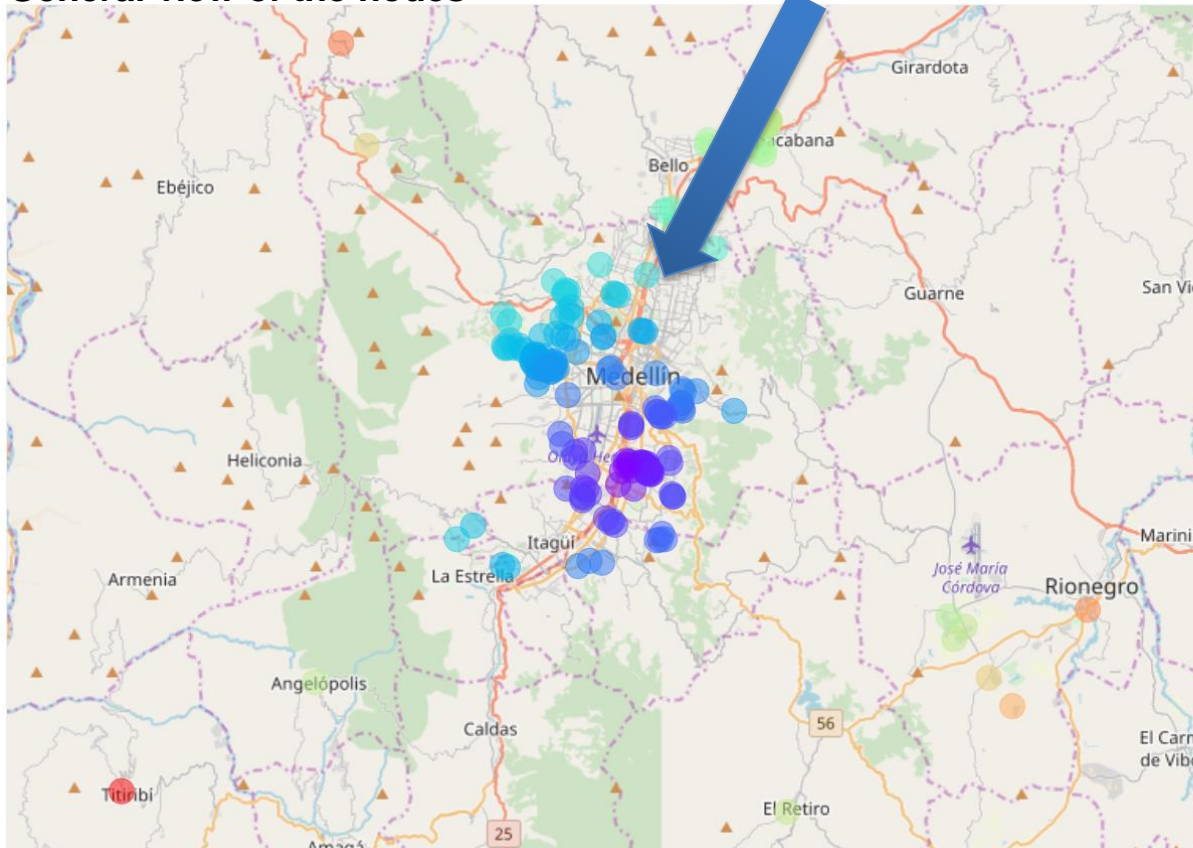
Data set	Memory consumption (MB)
U=4	2
U=11	13
U=205	112

Table 3: Execution time of the operations of the data structure for each data set and method.

Software prototype

General view of the nodes

Nodes
(position of each person)

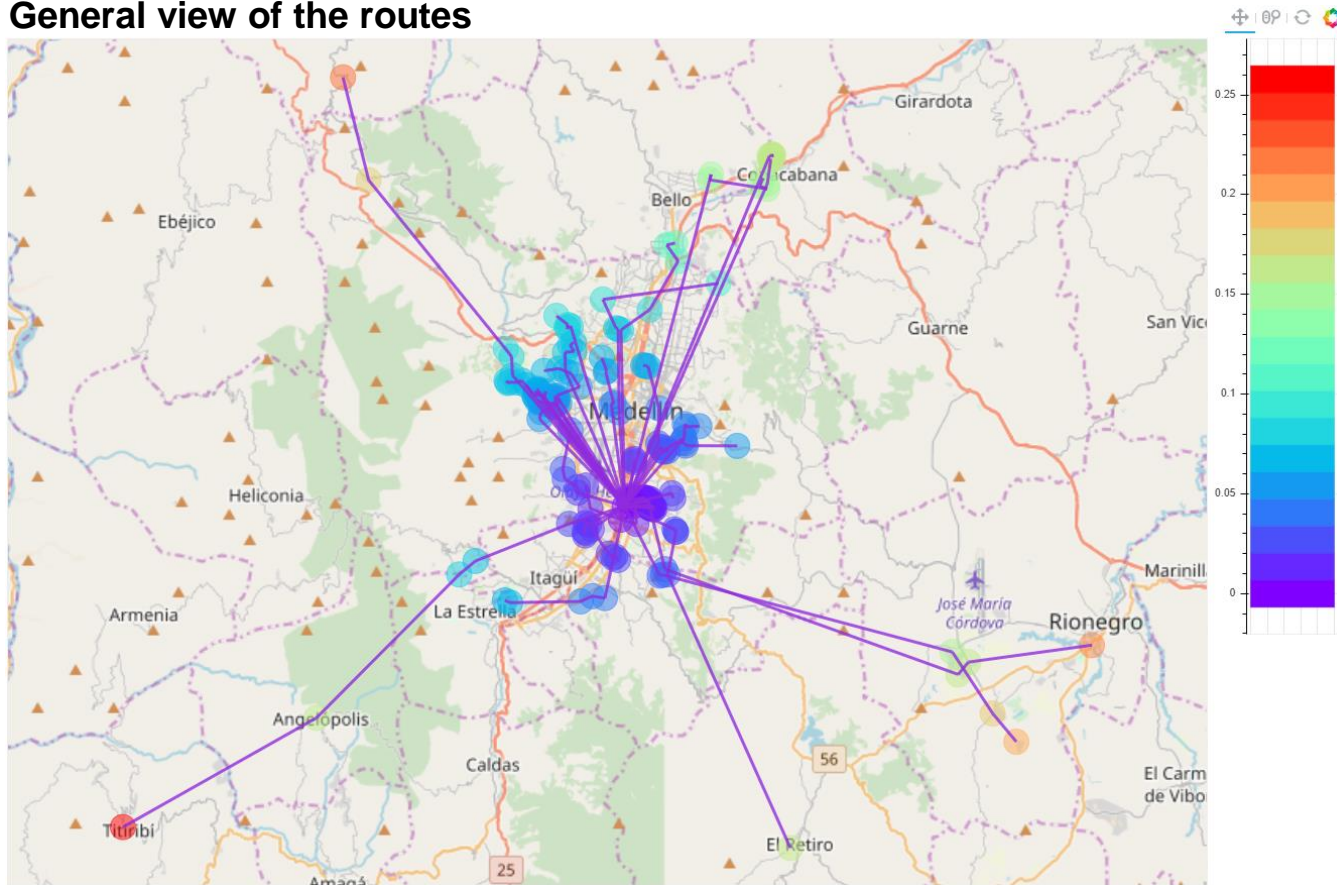


Tools

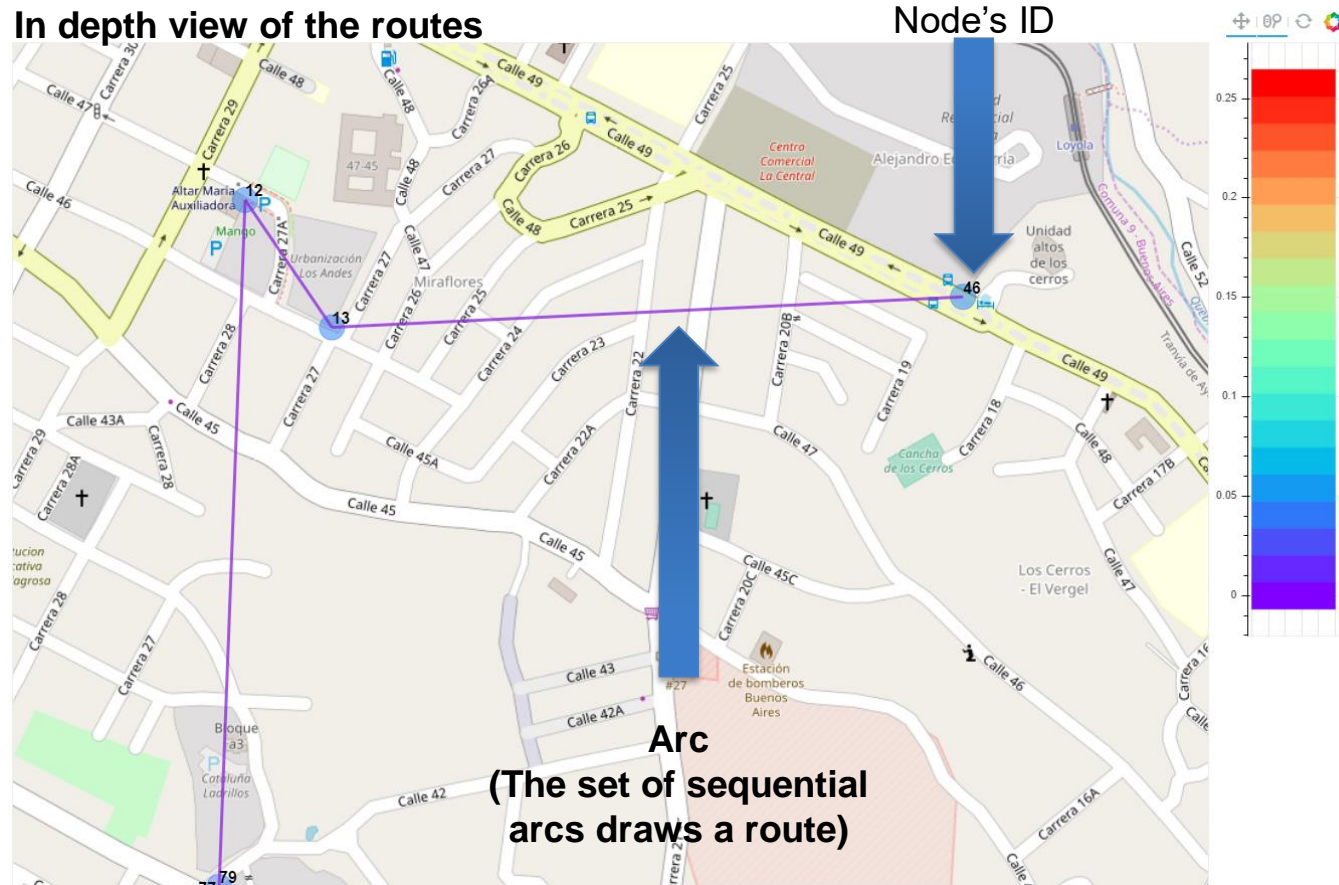
Distances
(In hundreds
of kilometers)

Software prototype

General view of the routes



Software prototype



Are the given coordinates useless?

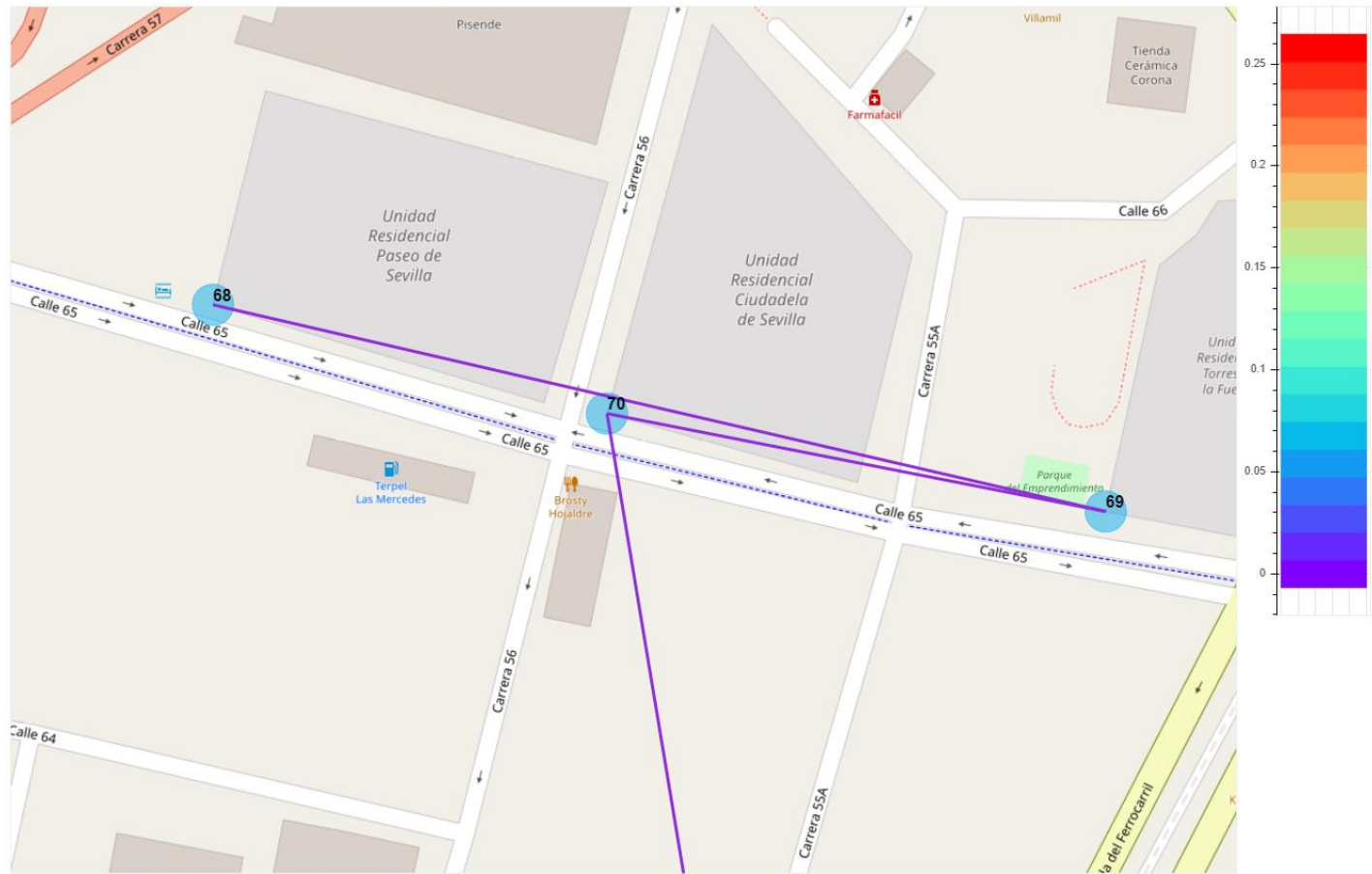


Figure 9: Pick-up order without taking coordinates into account

Are the given coordinates useless?

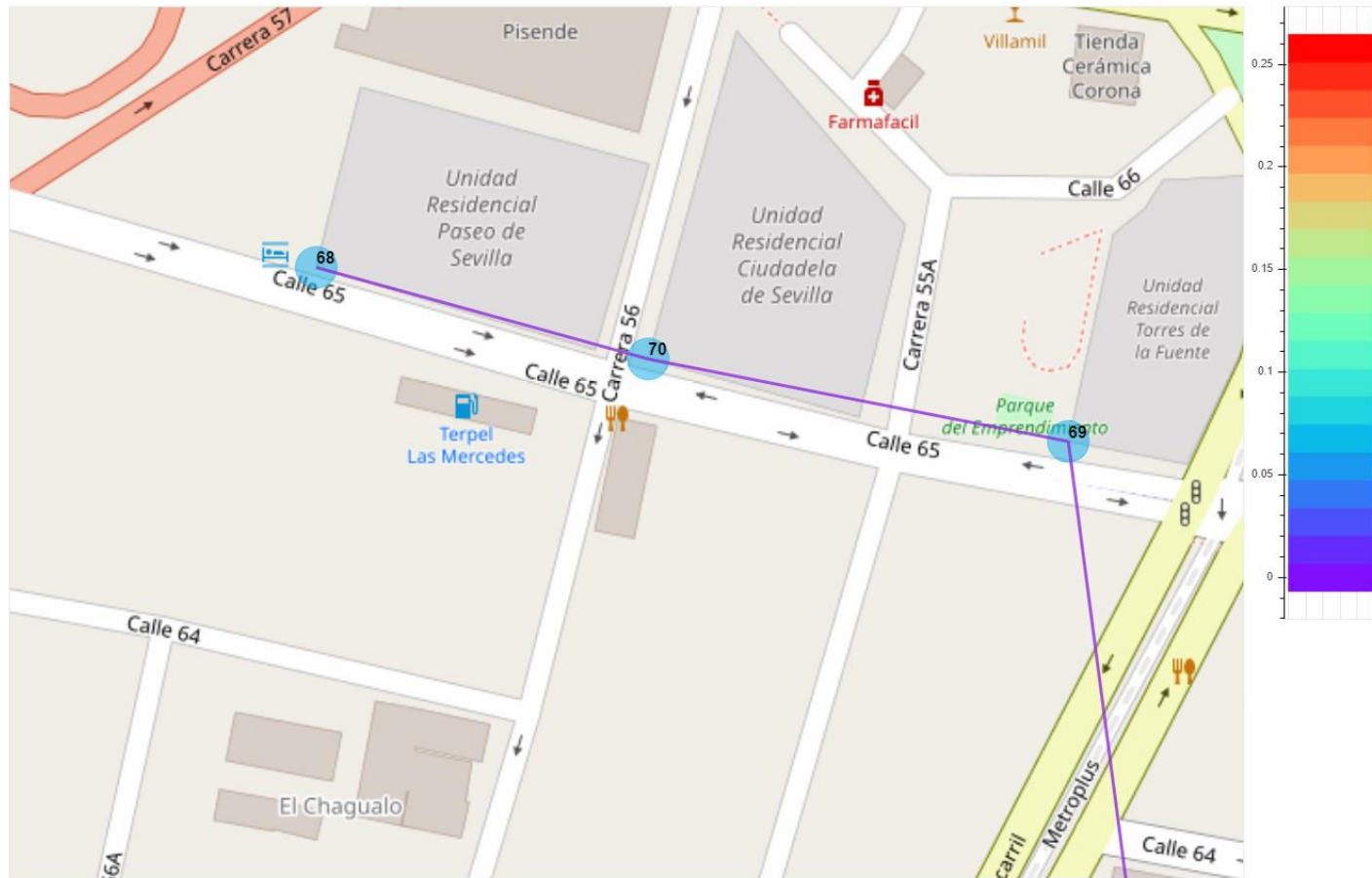


Figure 10: Pick-up order taking coordinates into account

Are the given coordinates useless?

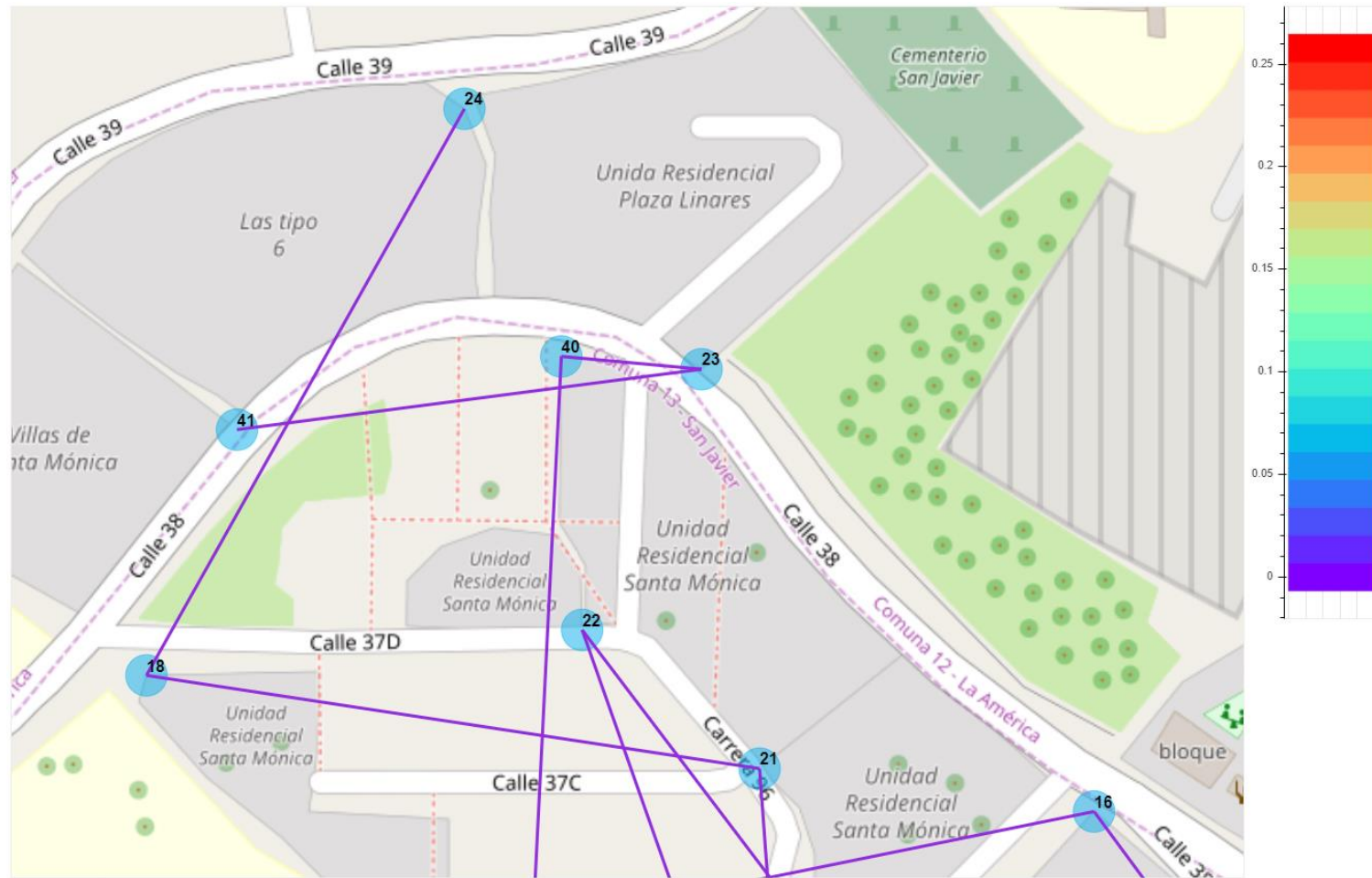


Figure 11: Pick-up order without taking coordinates into account

Are the given coordinates useless?



Figure 12: Pick-up order without taking coordinates into account

Are the given coordinates useless?

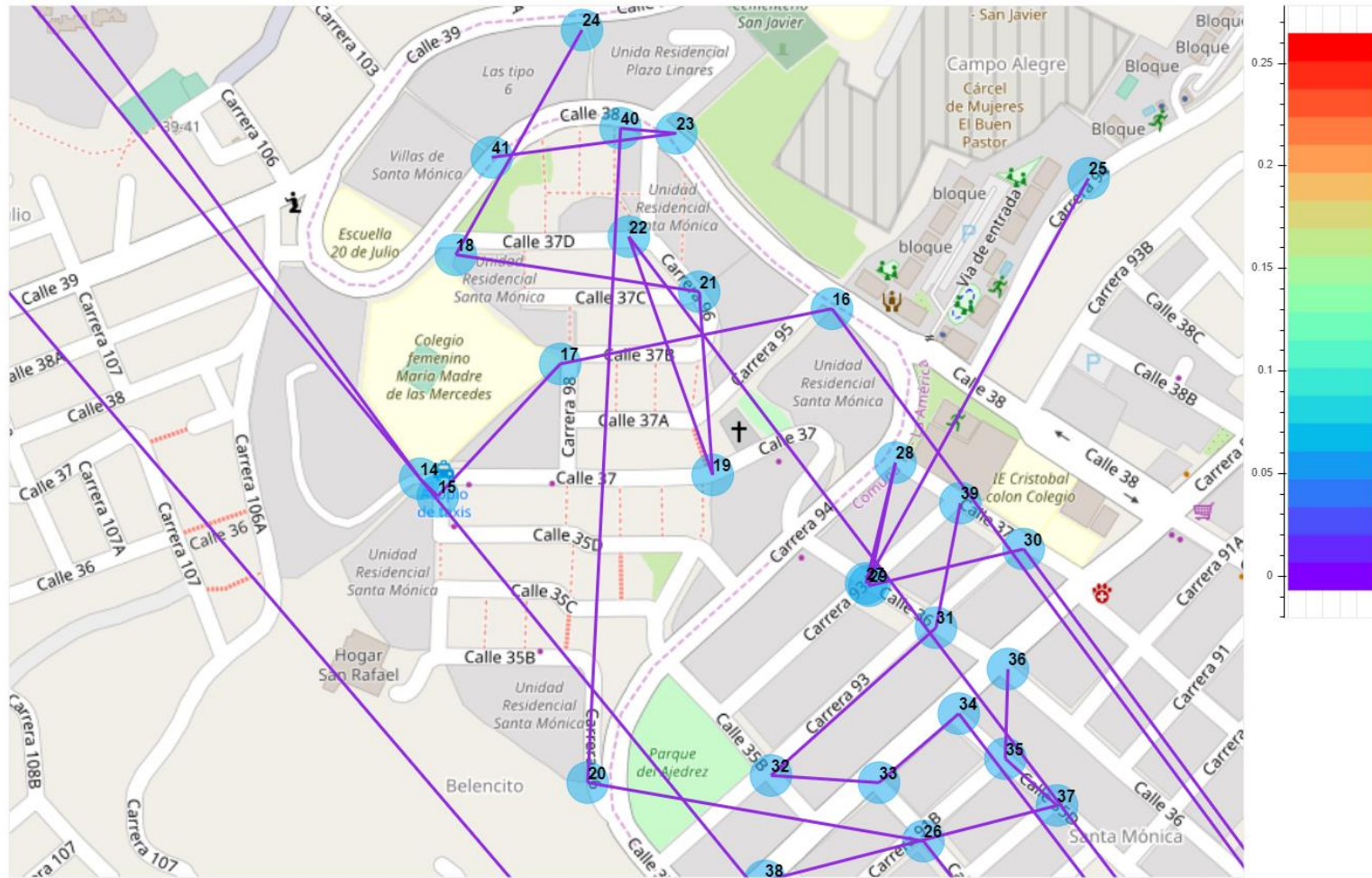


Figure 13: Pick-up order without taking coordinates into account

Are the given coordinates useless?

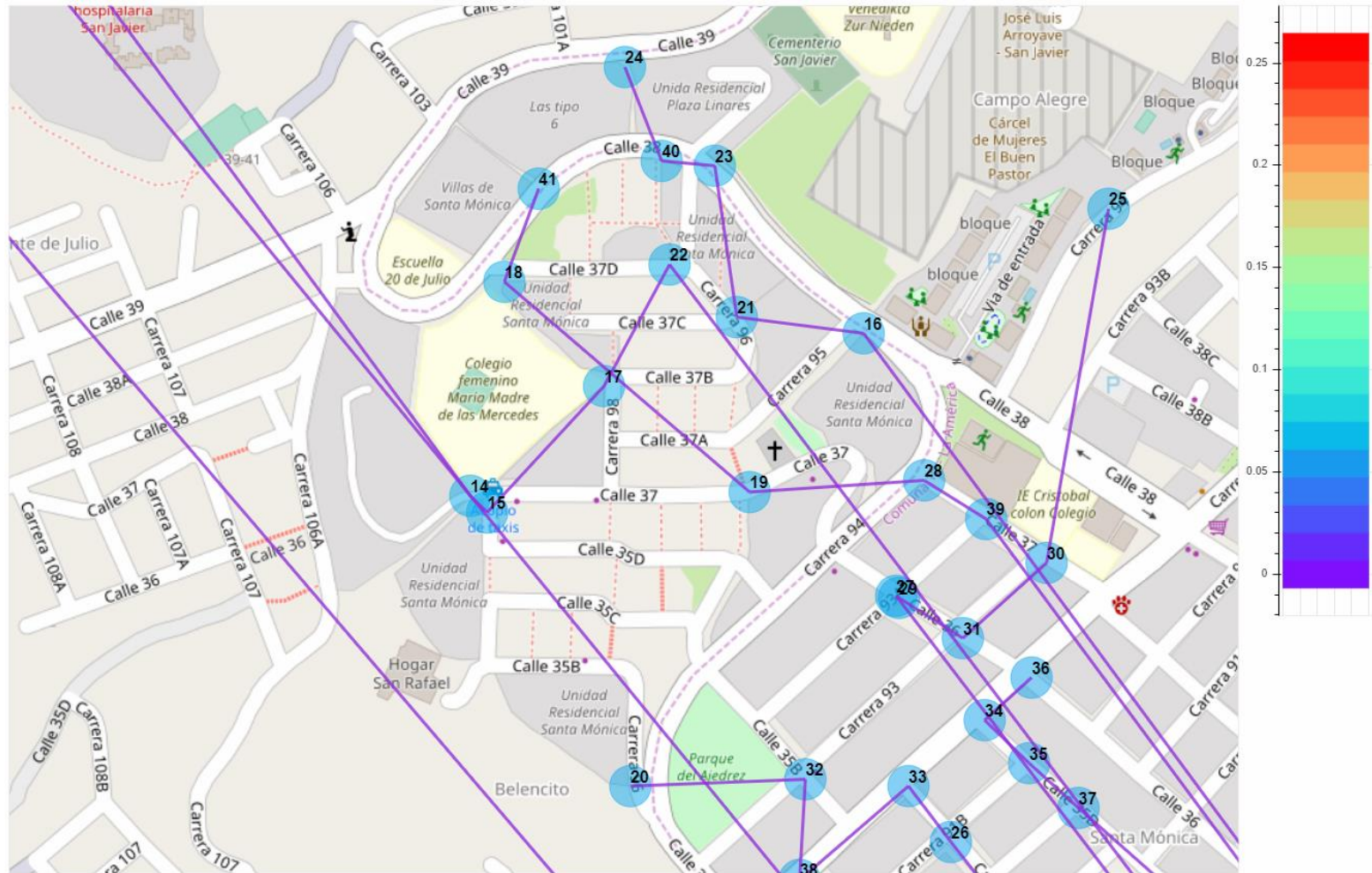


Figure 14: Pick-up order without taking coordinates into account

Are the given coordinates useless?

The next table shows in column X how many times the nodes have multiple nearest non picked-up nodes, for each 205 data set

Nodes	P	X
205	1.1	62
205	1.2	77
205	1.3	81

Table 4: Times a Node has multiple nearest non picked-up nodes

Are the given coordinates useless?

So using coordinates we can go from this:

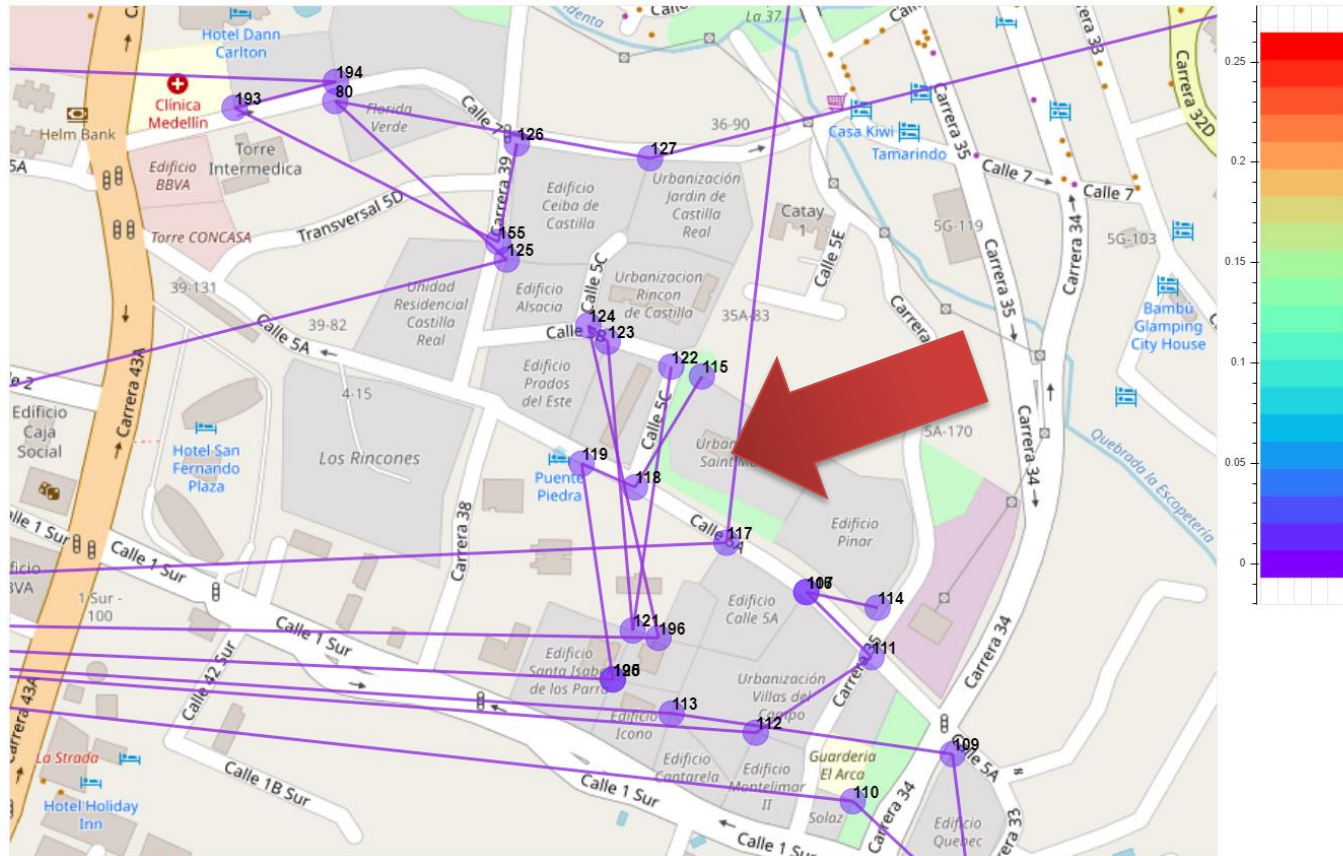


Figure 15: Pick-up order without taking coordinates into account

Are the given coordinates useless?

to this:

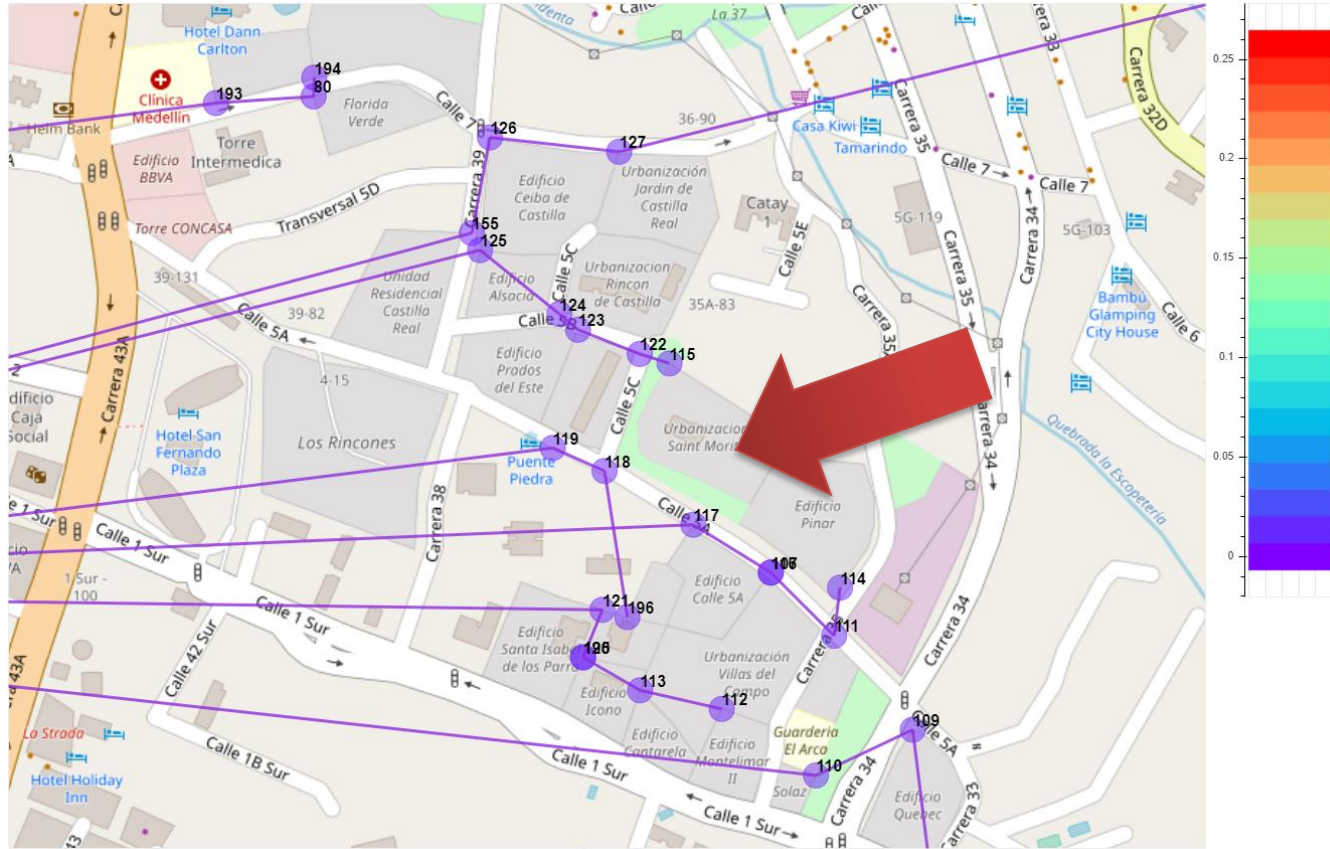


Figure 16: Pick-up order without taking coordinates into account

Is P a proper constraint?

Base line: is P was 1.0 or 0, the total amount of cars would be 77

* Result is the final amount of cars required

**When P
is a percentage:**

Nodes	P	Result	Cars reduced
205	1.1	58	146
205	1.2	50	154
205	1.3	48	156
205	2.67	41	163

Table 5: Total amount of cars with P being a percentage

**What if P was a fixed integer
instead of a percentage?**

Nodes	P	Result	Cars reduced
205	1	54	150
205	3	46	158
205	5	42	162
205	10	41	163

Table 6: Total amount of cars with P being a fixed number of minutes