Route optimization algorithm for ridesharing

María Sofía Uribe
Isabel Cristina Graciano
Medellín, 2019/05/16

Data Structures

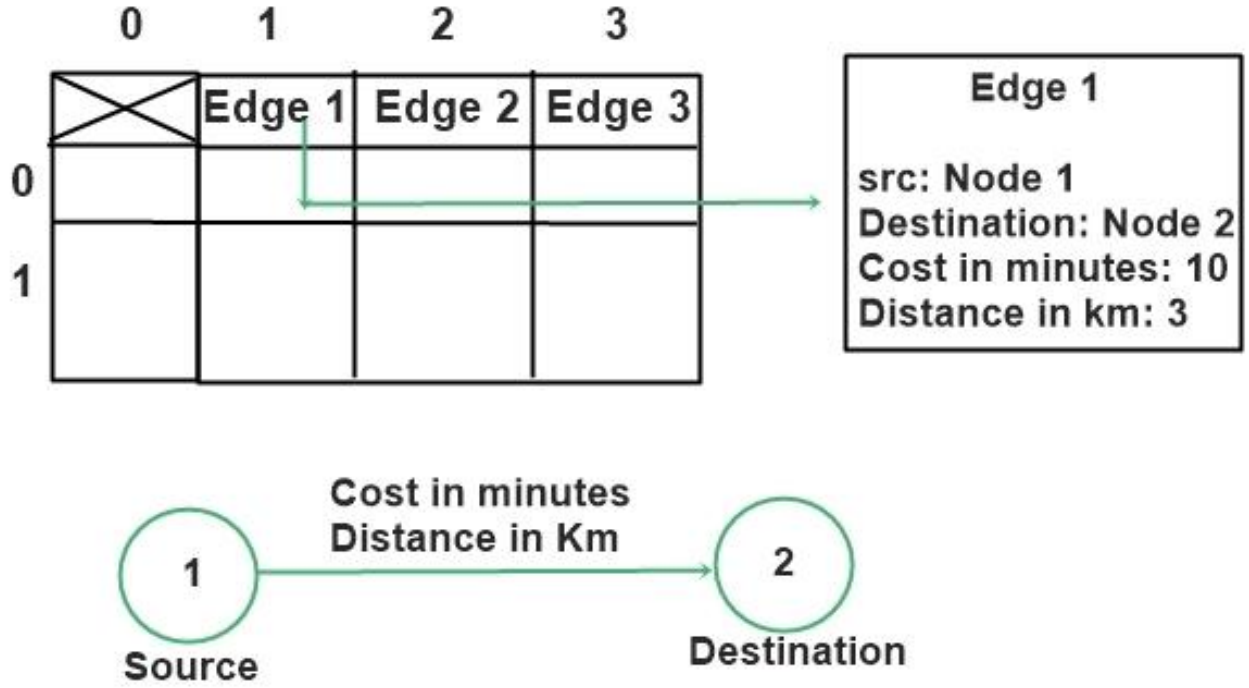


Figure 1: Data structure of the adjacency matrix that stores an edge and this represents the arc connecting two nodes.

Data Structures

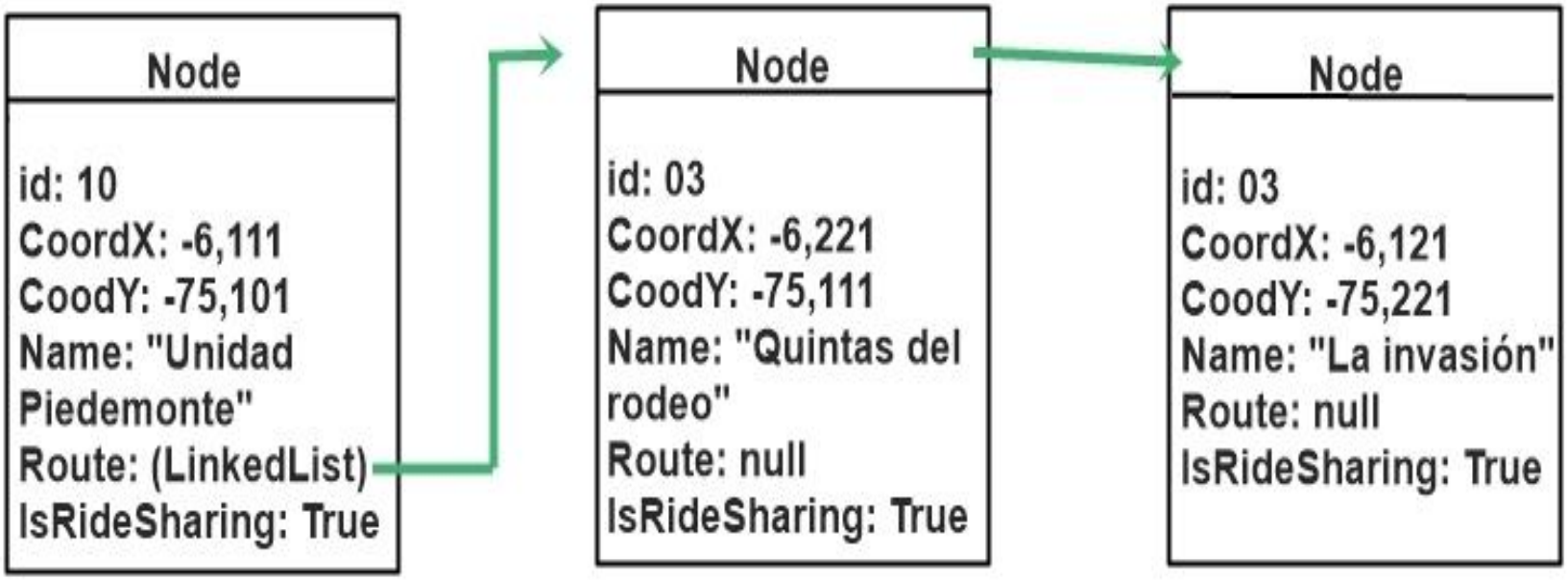
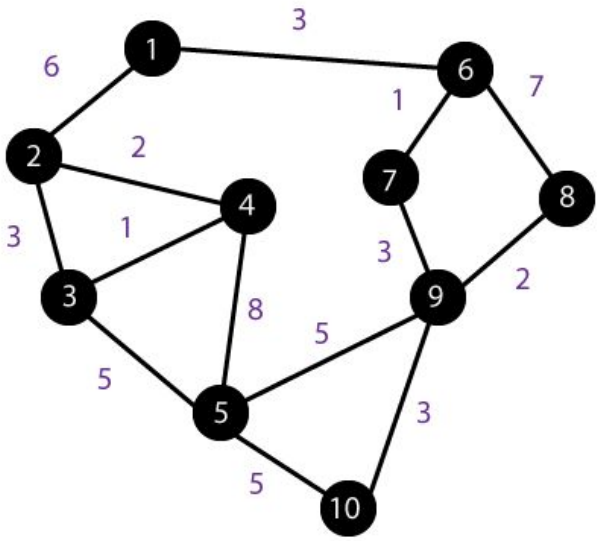
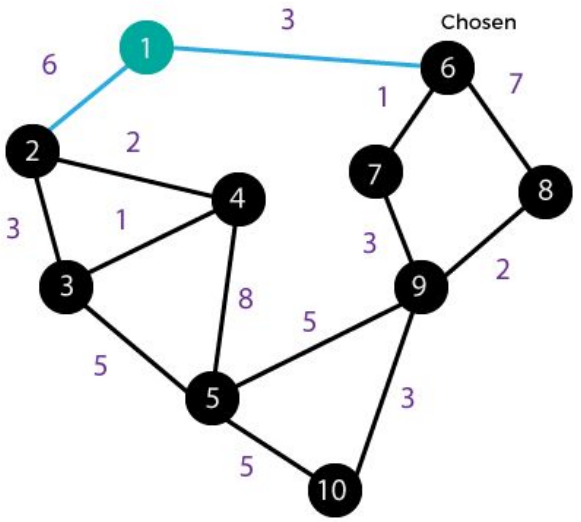


Figure 2: Data structure of the list that contains each node's route.



Heuristic

h(1) = 10
h(2) = 8
h(3) = 5
h(4)= 7
h(5)= 3
h(6)= 6
h(7)= 5
h(8)= 3
h(9)= 1
h(10)= 0



h(1) = 10
h(2) = 8
h(3) = 5
h(4)= 7
h(5)= 3
h(6)= 6
h(7)= 5
h(8)= 3
h(9)= 1
h(10)= 0

Figure 3, 4: Explains the A* search algorithm.

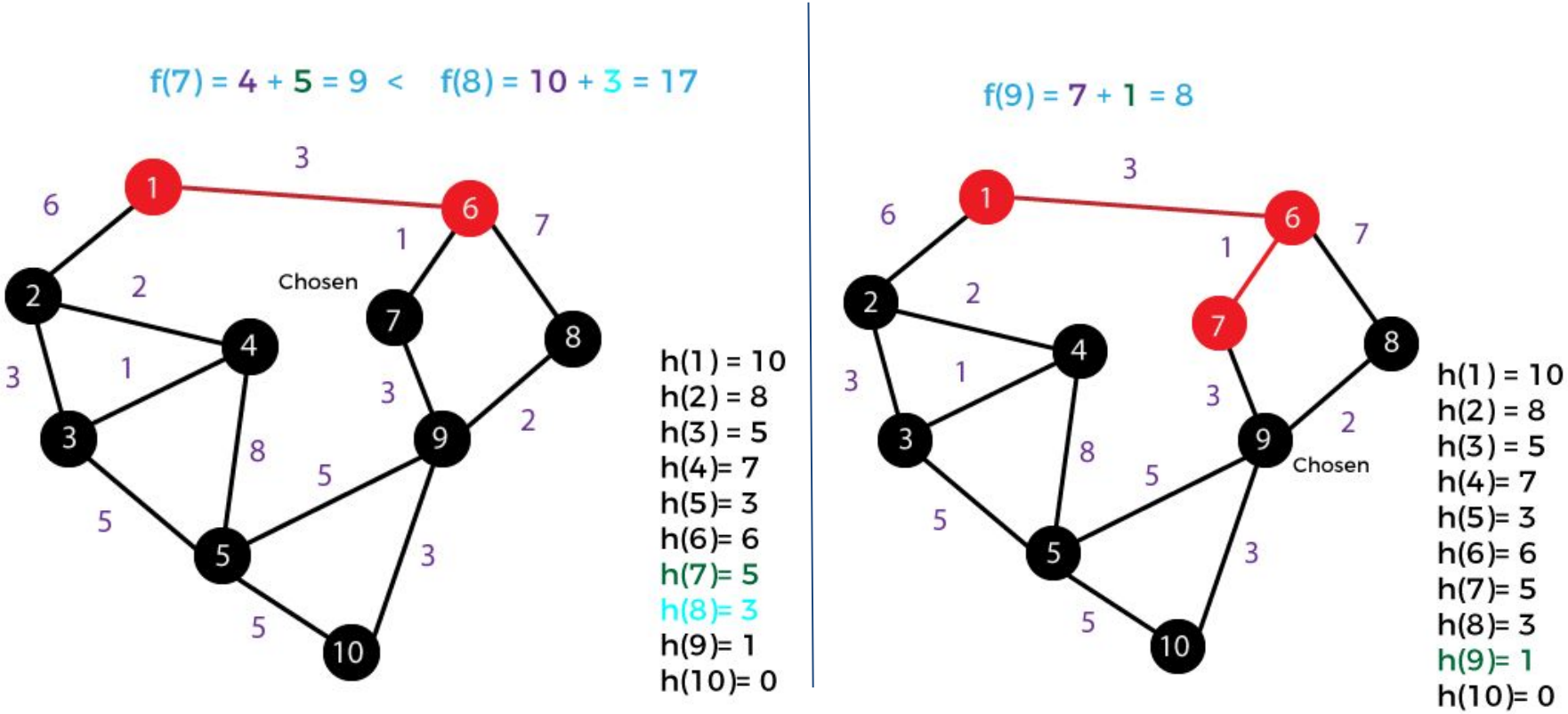


Figure 5, 6: Explains the A* search algorithm.

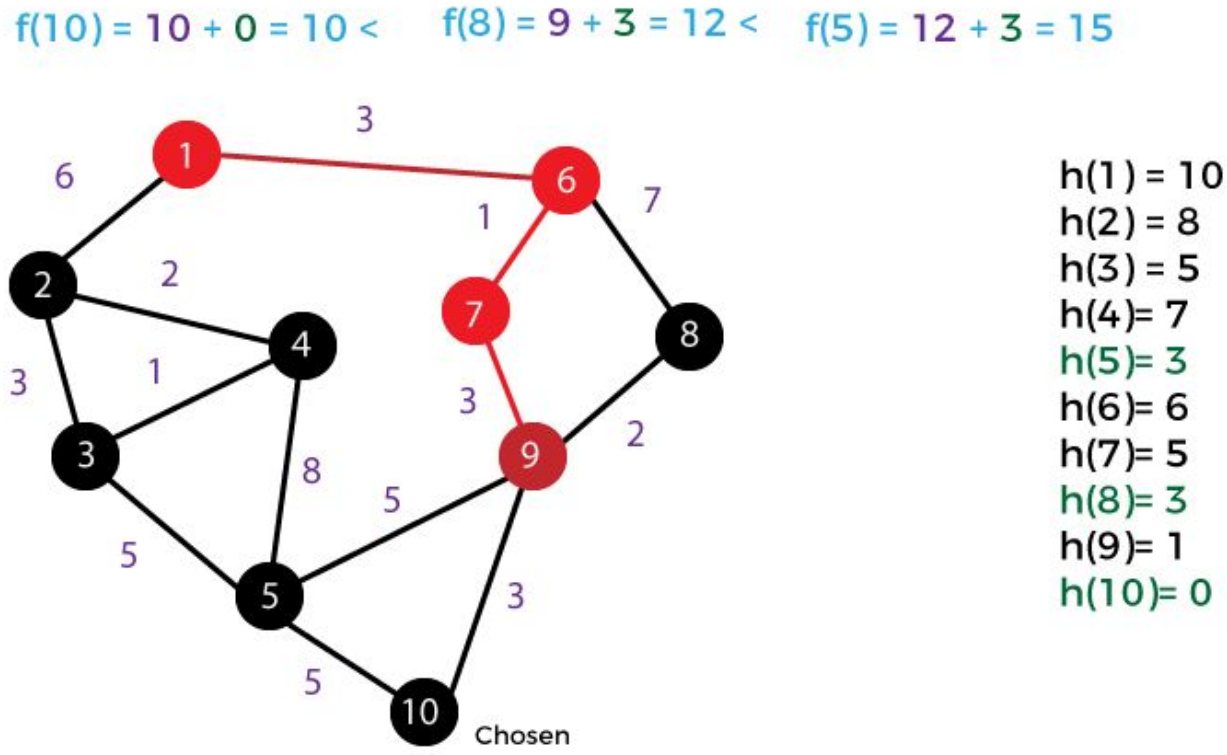


Figure 7, 8: Explains the A* search algorithm.

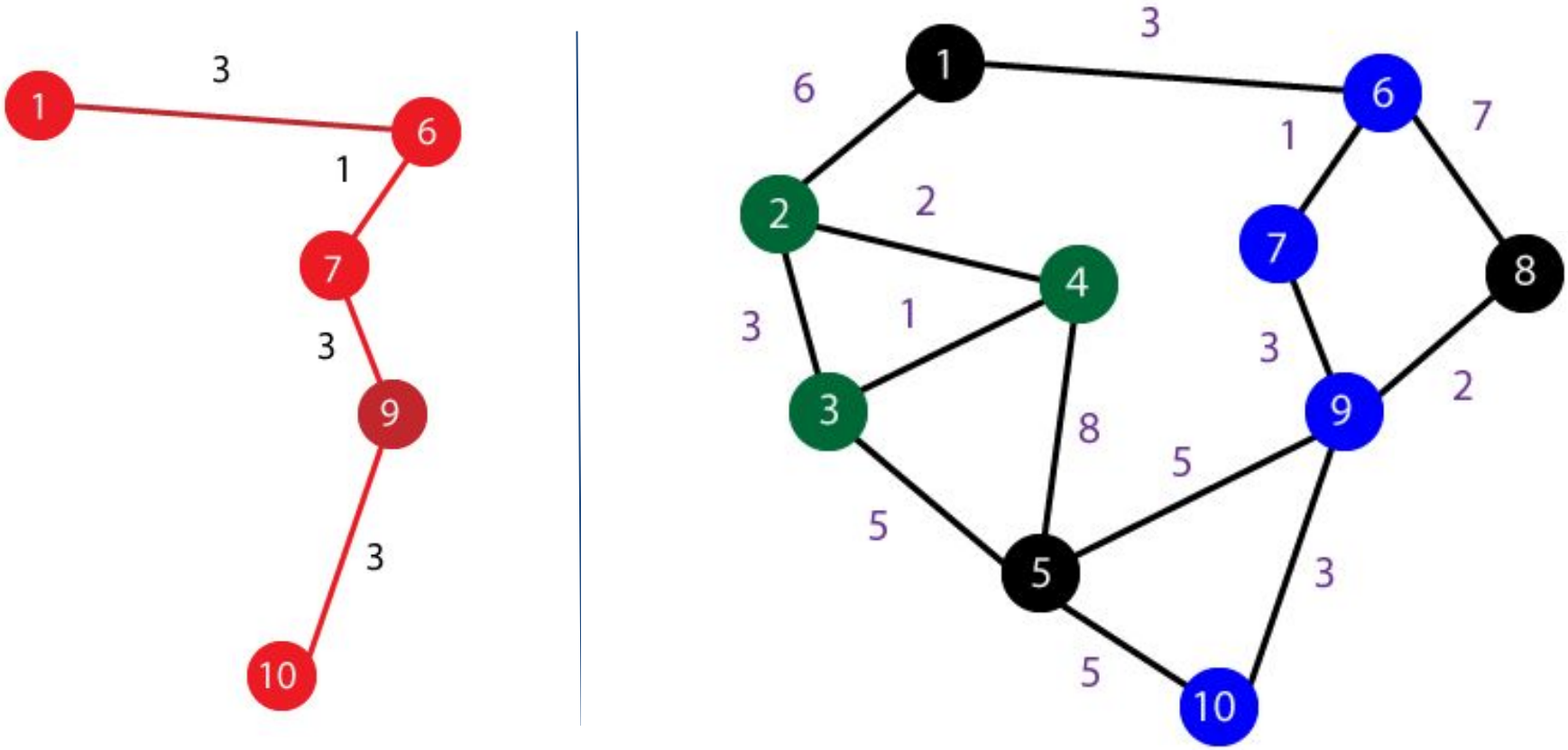


Figure 9, 10: Explains the A* search algorithm.

Algorithm and Complexity

Subproblem	Worst Case Complexity
Creating the graph (Reading the file and creating the graph)	O(V+E)
Directed Search(A*) (Reconstruct one short path to 1)	O(V+E)
Assign Vehicles	O(V^2+EV)
Total complexity	O(V^2+EV)

Table 2: Time complexity of every important method in the program and the total complexity of the algorithm.

Algorithm design criteria

After the analysis of some possible solutions, we concluded that as we have a complete graph is easier to use a matrix representing the edges because it allows us to reduce the required time for searching, insertion, and we also have a lower time complexity than with other methods.

Furthermore, in order to find the shortest path among the nodes in the Graph we used A* algorithm because it has a low complexity in the worst case performance so it will not increase the performance of our algorithm, also, it can find a path between multiple points and as it uses the heuristic cost it does not have to calculate some obvious paths that are not going to match with the optimal solution.

Time and Memory Consumption

Dataset	5 cars P=1.2	5 cars P=1.7	205 cars P=1.1	205 cars P=1.2	205 cars P=1.3	205 cars c= 6.1
Memory Consumption	0 MB	0 MB	8 MB	10 MB	11 MB	12 MB

Table 3: Memory consumption of the ridesharing algorithm based on five given dataset.

Software prototype

Number of cars	5	5	205	205	205	**205**
P or C	P=1.2	P=1.7	P=1.1	P=1.2	P=1.3	**C=6.1 (constant extra time of 6 minutes for all cars)**
Number of routes assigned	2	2	59	57	48	**41**

Table 4: Execution and results of the program.