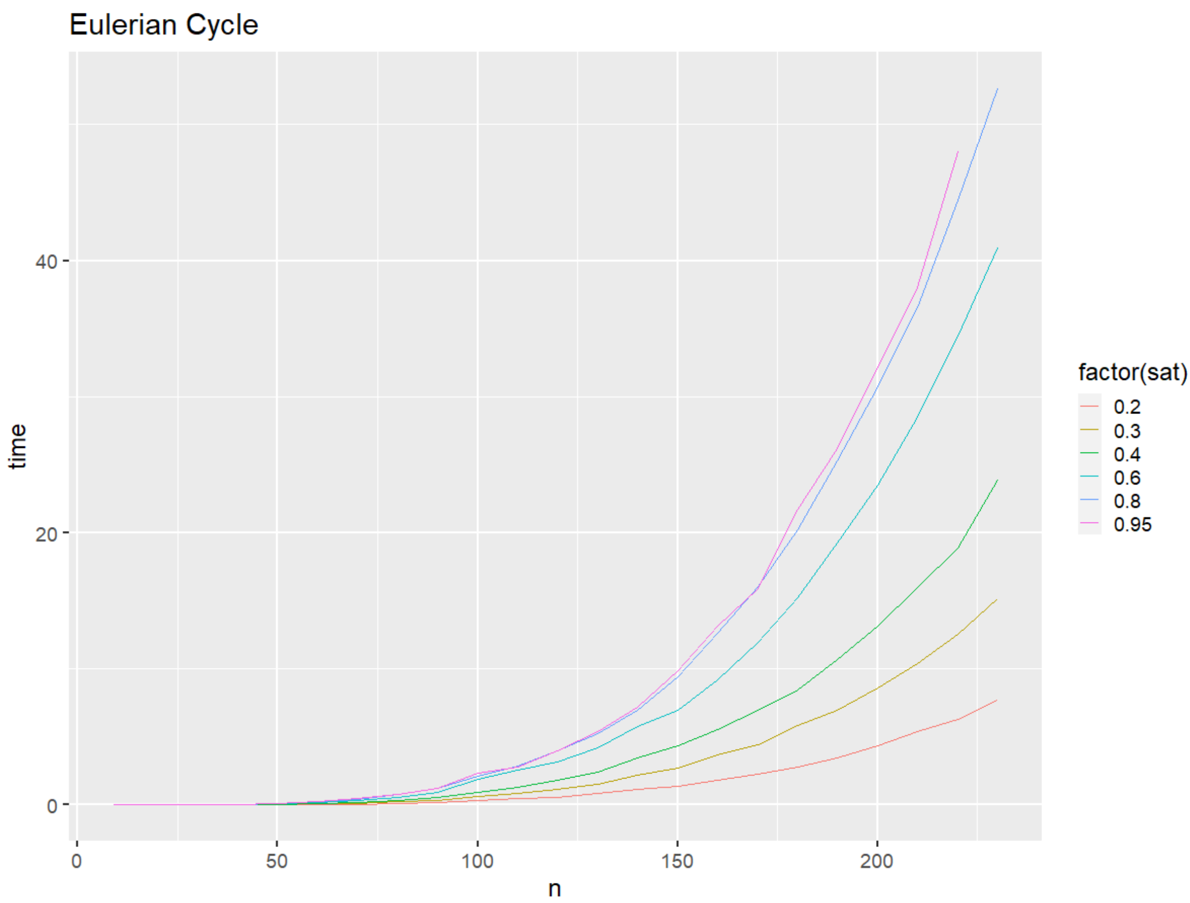
Report 4 - Backtracking Algorithms
Bogna Kilanowska 148252

Time of searching for an Eulerian Cycle for different graph saturations

<u>Data:</u>
The time was measured for random generated graphs represented by List of Incidents, consisting of 10, 20, …, 230 vertices.

<u>Chart:</u>



.<u>Conclusions:</u>
Algorithm performed much better for graphs with lower saturation. The complexity of this algorithm was O(V + E).
Graphs were represented by List of Incidents, due to some important aspects:

- Incident List consumes only (v + e) space in storage, which allows to run the experiments on bigger graphs,
- time complexity of accessing the list of adjacent vertices is O(1) - it was important, because the algorithm was based on the BFS

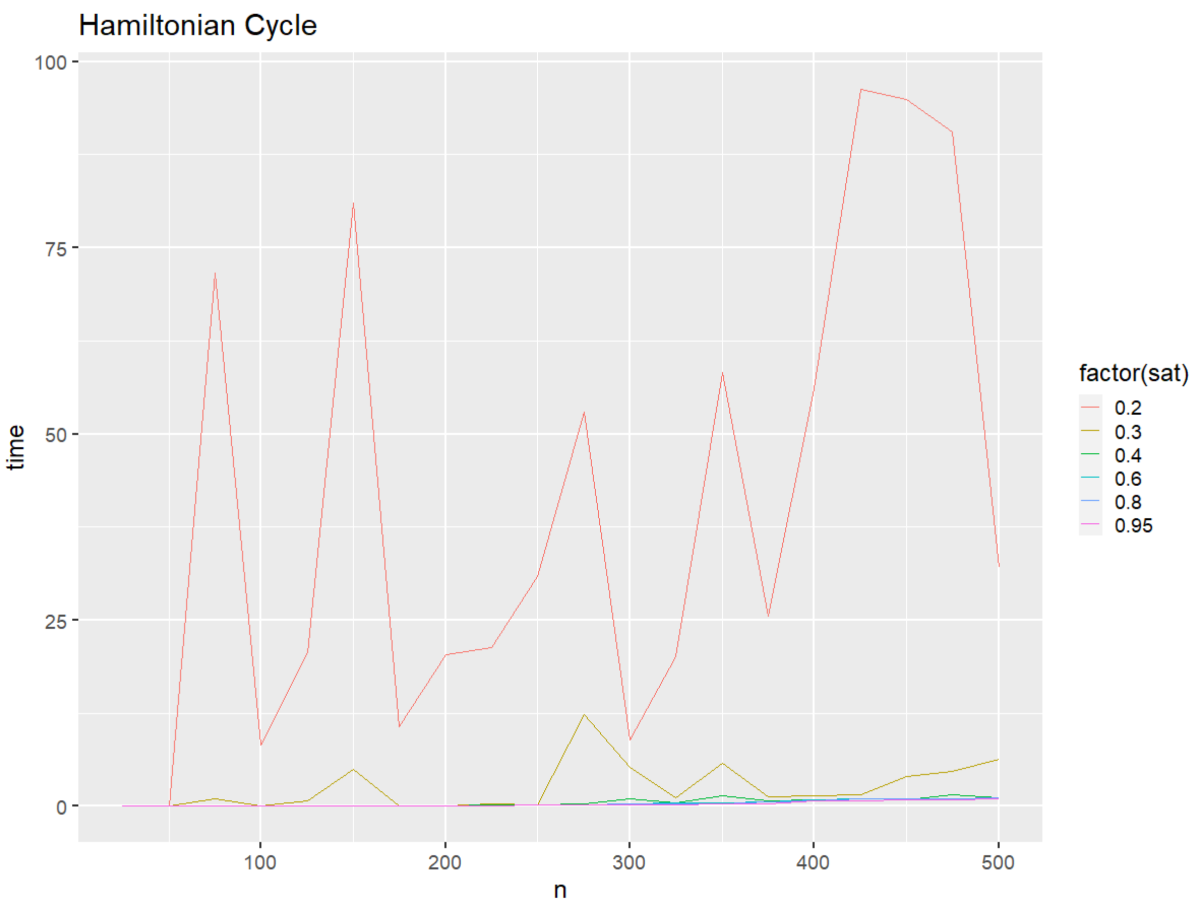
The drawback of this representation was O(v) time complexity of checking the existence of the edge between 2 vertices - time complexity of the same procedure in Adjacency Matrix is just O(1). However, the Adjacency Matrix wasn't chosen, due to its big storage consumption. Edge Matrix has the same problem. List of Edges consumes least storage, but time complexity of searching for an edge is O(e), which in general is longer than O(v).

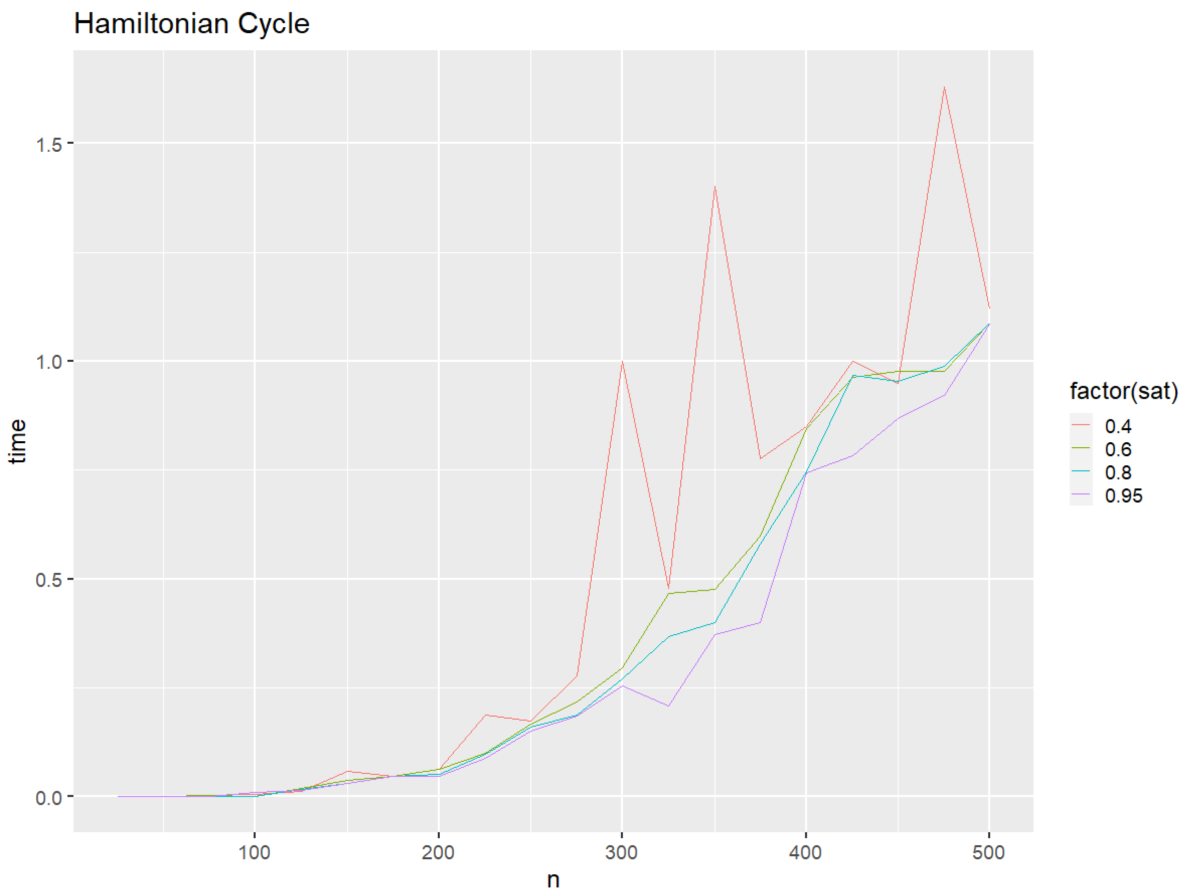
v - number of vertices
e - number of edges

Time of searching for an Hamiltonian Cycle for different graph saturations

<u>Data:</u>
The time was measured 10 times for random generated graphs represented by List of Incidents, consisting of 25, 50, …, 500 vertices.

<u>Chart:</u>





<u>Conclusions:</u>

For the Hamiltonian Cycle, the bigger the saturation, the better the algorithm performs. Its worst time complexity is O(n!), but for higher saturation probability of acquiring the worst case scenario is smaller than for lower saturations.