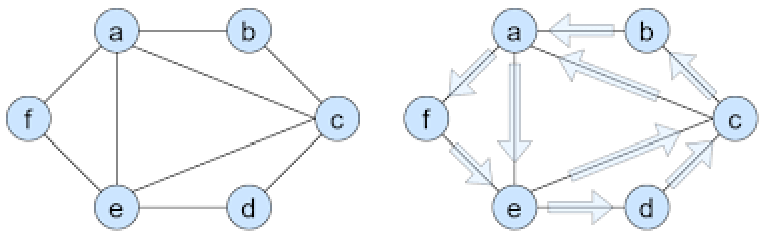
Małgorzata Gierdewicz, ID: 148264

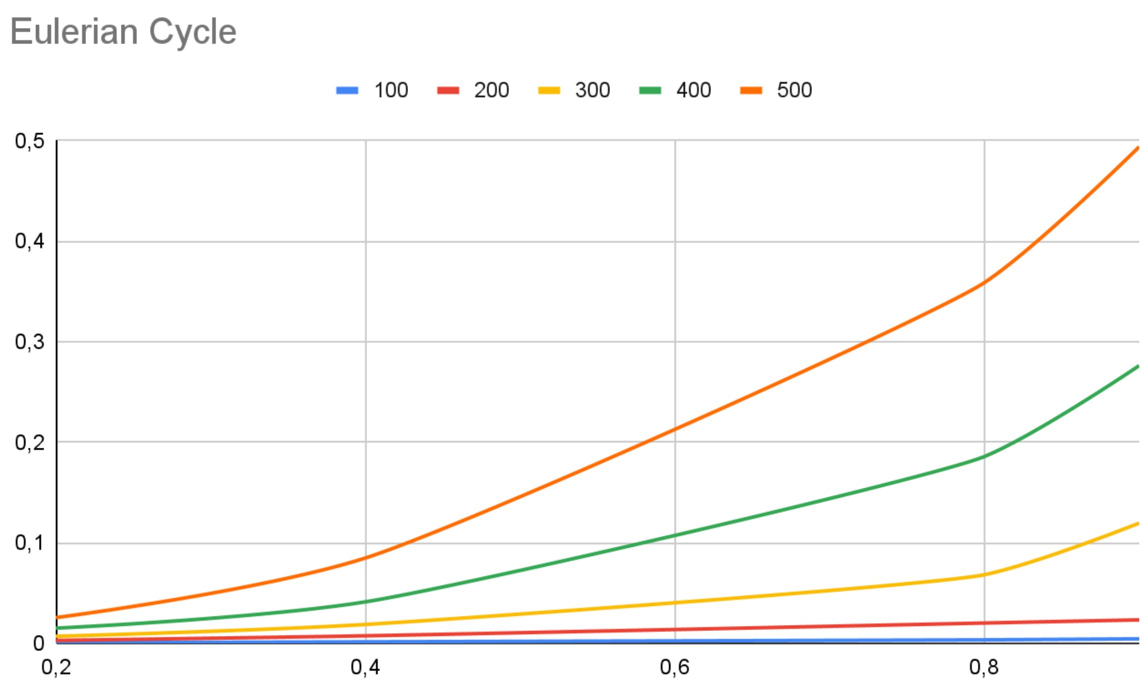
This project considers the hamiltonian cycle search and eulerian cycle search, both of which are part of a much wider scope of problems called backtracking problems.

Eulerian Cycle



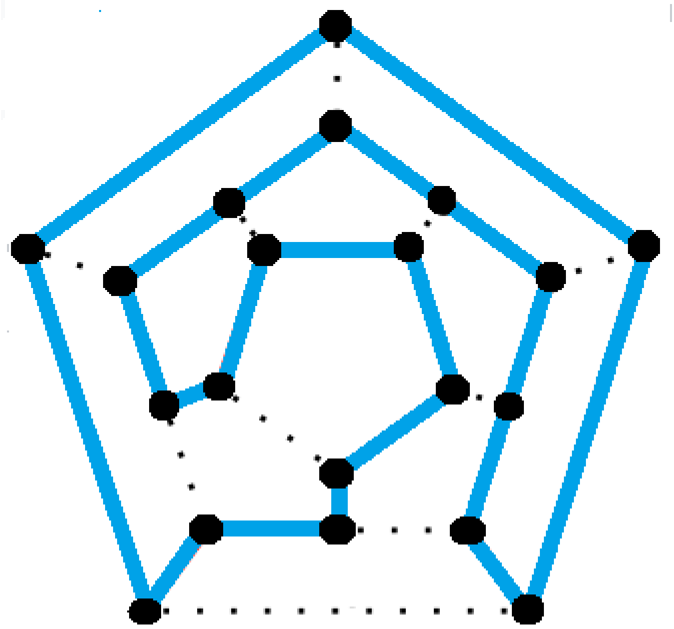
Euler Circuit: a-e-c-a-f-e-d-c-b-a

Eulerian Cycle is a trail that visits each edge in a graph exactly once. Since we need to find a graph that will visit all of the edges, it will take longer to find an eulerian cycle in a graph that has more edges, therefore the search time increases with an increase in graph saturation.

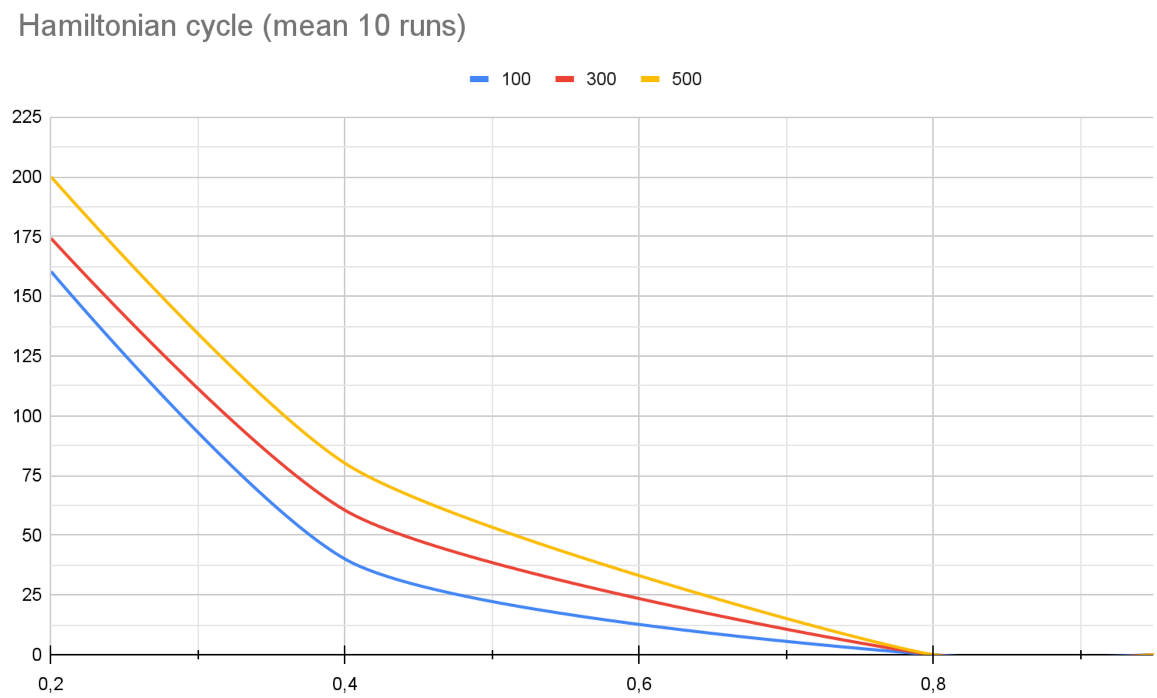


Graphs that were generated for Eulerian Circuit problem were undirected graphs with maximum number of edges equal to (V * (V-1) / 2) * saturation, where V = number of vertices. Since the time complexity of the algorithm is equal to O(n^3), hence the parabolic shape.

Hamiltonian Cycle



Hamiltonian Cycle is a trail in a graph that visits each node exactly once. In order to find a hamiltonian cycle, a backtracking algorithm was used. In the backtracking algorithm, when visiting a node, we need to find a vertex that wasn't traversed before. Search space for this problem is a permutation of all possible paths that we can find in this graph and the solution space is equal to paths from search space that satisfy the constraints. Time complexity of such a problem is O(n!), since we have to check all possible outcomes.



The time needed to find a Hamiltonian Cycle in a graph decreases with an increase in edge saturation, because the more edges the graph has, the more probable it is to find a path that visits each node exactly once. The number of nodes that the traversed graph had wasn't as influential to the time needed to find a circuit as the saturation of a graph. Because of the time complexity of the backtracking algorithm, a time limit was implemented and each operation had to be finished after 200 seconds.