Hidato Solving

Names: Noam Gianella & Noa Finkelshtein
Advisor: Prof. Daniel Berend

Original Hidato: Hidato is a logic puzzle, developed by Dr. Gyora M. Benedek, an Israeli mathematician. In Hidato, a grid of \( n \) cells is given, some cells are labeled with a natural number from 1 to \( n \). The goal is to find the label of each cell, so that there’s a path of adjacent cells (vertically, horizontally or diagonally) from label 1 to label \( n \).

We consider the following generalization of Hidato: Given is a graph on \( n \) vertices, where some of the vertices are labeled by distinct numbers between 1 and \( n \). The goal is to find a Hamiltonian path which passes through the labeled vertices at the times (labels) showed at those vertices.

Graph Generators:
- **G(n,p) Generator:** Constructs a graph according to the G(n,p) model, with an additional parameter, \( l \), the number of labeled vertices. The choice which vertex receives a label is done randomly.
- **Path plus Generator:** Constructs a graph according to the number of vertices, \( n \), number of labeled vertices, \( l \), and a constant \( c \). The generated graph instance starts with \( n-1 \) edges forming a path that visits each vertex exactly once, and then \( c \times n \) edges are added uniformly randomly out of the remaining \( \binom{n}{2} - (n - 1) \) potential edges. An instance generated by this generator will have at least one solution.

Solvers:
- **Naive Solver:** We solve the problem by splitting it according to the intervals defined by the known labels, searching all possible solutions for each interval separately. Afterwards, we combine the intervals’ solutions to a complete one (if such exists).
- **Matching Solver:** We design an algorithm that checks, according to the game constraints, whether there is a vertex in the graph that must have a certain label or there is a label that must appear in a specific vertex.
- **SAT Reduction Solver:** We map an instance of the problem to a CNF expression, use the miniSAT solver, and map the solution back to a solution for our problem.

Statistical Analysis:
- Capabilities and performance analysis of our solvers.
- For graphs that were generated with the G(n,p) generator, we analyzed the relation between \( p \) and the number of solutions.
- For graphs that were generated with the Path plus generator, we analyzed the relation between \( c \), and the number of solutions.