Search Algorithm Evaluation System on 3*3 Eight puzzles

Individual Report

Name: Arshdeep Kaur (aka232)

Group Name: Moodeng

Submission Date: 17th December 2024

1. Framework

During the development and testing phase, several issues were identified in the visualization part of the framework. The command line parser did not correctly handle algorithms requiring heuristic parameters, leading to execution errors or. The issue was resolved by updating the parser logic to properly validate and process heuristic inputs, ensuring accurate algorithm execution.

2. Algorithms

Greedy: Implemented the Greedy algorithm, which makes locally optimal choices at each step to find a solution efficiently, though not always globally optimal.

IDDFS: Combined depth-first search with iterative deepening to achieve completeness while maintaining low memory usage, ideal for deep search spaces.

IDA*: Extended IDDFS with heuristic cost evaluation, allowing for more efficient exploration by prioritizing paths with lower estimated costs.

3. Heuristics

Disjoint PDB: Disjoint sets are utilized to divide puzzle tiles into non-overlapping groups, simplifying the computation of heuristic costs by reducing the state space that must be explored. This approach significantly reduces the computational cost of heuristic calculation by focusing on smaller subsets of tiles. By precomputing the shortest paths from various tile configurations within these groups to the goal state, PDBs help us find the solution to the puzzle faster.

Disjoint set selection: For the 3×3 sliding puzzle (8-puzzle), I tested two different disjoint set configurations: a 3–3–2 set, where tiles are divided into three groups with sizes 3, 3, and 2; and a 4–4 set, where tiles are divided into two groups of size 4. After running these configurations on multiple test cases, I observed that the 3–3–2 disjoint set consistently provided better results. It not only reduced computation times but also produced solutions closer to the suboptimal solution when compared to the 4–4 configuration, making it the preferred choice for this puzzle size.

I also tested different disjoint sets for 4 * 4 such as 4 – 4 – 4 – 3, 9 – 6 but I wasn't able to generate PDB for either of those because of the high computational cost that takes hours which is not feasible on my general desktop computer. The high cost, requiring hours of processing time, made it impractical to compute and use these configurations effectively. Consequently, further experimentation with these disjoint sets was limited due to resource constraints.

4. Report

I worked on the report along with my group partners.