

**SSE:** Fall 2024

**CSC 4301:** Intro to AI

**Project 1:** Report

**Team:** Amine Lahnin, Omar Ismail,

Mohammed Harouak

1. **Introduction*:***

This project aims to study and comprehend the use of heuristic-informed search algorithms to solve search problems. The project revolves around the eight-puzzle problem by transforming it to a fifteen-puzzle problem and applying various search algorithms to determine which performs best and why. Both puzzles share similar criteria that classify them as search problems; they both consist of:

-State Space: Confined in a 3x3 grid for the eight puzzles, and a 4x4 grid for the fifteen puzzles, where the blank tile can move in.

-Successor Function: Which allows the blank tile to move from one state to another within the bounds of the state space.

-Start State: The randomly generated beginning state for each puzzle.

-Goal Test: a test function to assess whether the puzzle is solved. This occurs for the eight puzzle only when the cells are ordered in an ascending order, as follows:

|  |  |  |
| --- | --- | --- |
|  | **1** | **2** |
| **3** | **4** | **5** |
| **6** | **7** | **8** |

As for the fifteen puzzles, the solution must have the blank at the bottom right of the table, with the cells being ordered in an ascending order, as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **1** | **2** | **3** | **4** |
| **5** | **6** | **7** | **8** |
| **9** | **10** | **11** | **12** |
| **13** | **14** | **15** |  |

The project will therefore cover the following tasks:

-Task 1: Transforming an eight puzzle into a fifteen puzzle.

-Task 2: Implementing four different heuristics for the A\* algorithm in the fifteen puzzles.

-Task 3: Comparing the heuristics to determine the most effective.

-Task 4: Comparing the winning heuristic with other uninformed search algorithms (BFS, DFS, and Uniform Cost Search).

Task 1: Transforming 8-puzzle to 15-puzzle

-File: fifteenpuzzle.py

The 8-puzzle operates in a 3 by 3 table, one blank space, and 8 other spaces. As for the 15-puzzle, it should instead operate in a 4 by 4 tables, with 1 blank space and 15 other spaces, hence the name 15-puzzle. To support a 4x4 table, we changed:

-The row & col count both to 4 in the FifteenPuzzleState constructor where the table cells are initialized, and the isGoal(self) function where it handles goal checking of the current state:

A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

-The legalMoves(self) function to not allow movement beyond the newly extended table borders (row < 3 for down, col < 3 for right):

A screen shot of a computer program

Description automatically generated

-The ASCII drawing function getAsciiString(self) to handle drawing a larger 4x4 table:

A screen shot of a computer program

Description automatically generated

-The rest of the changes in the fifteenpuzzle.py file represent name refactoring of the classes and functions to better suit the FifteenPuzzle context, as well as commenting out the unused loadFifteenPuzzle:

A screenshot of a computer program

Description automatically generated

-Execution Trace:  
Running fifteenpuzzle.py:

A screenshot of a computer program

Description automatically generated

The Final State:  
A screenshot of a computer

Description automatically generated

Task 2: Heuristic h1, h2, h3, and h4 implementation in the 15-puzzle:

-Heuristic 1: Number of Misplaced Tiles:  
The first heuristic consists of calculated the number of misplaced tiles from the goal state.

The implemented function h1 goes as follows:

A computer screen shot of a code

Description automatically generated

Usage in the A\* function:

For all the heuristics, replacing the nullHeuristic with the intended one would suffice:

A screenshot of a computer program

Description automatically generated

Execution Trace: