Using A* to Solve the Rush Hour Logic Puzzle

85 Points

CS 360
Spring 2022

1 Objective

In this project you will model and solve a logic puzzle using the A* state space search algorithm. This project will introduce you to the complexities of modeling, heuristic formulation, and state-space search algorithms.

2 Groups/Collaboration

You may collaborate with your fellow students.
That being said each student will be responsible for turning in their own code.
The following are things which constitute collaboration:

- Asking a fellow student to explain an approach.
- Asking a fellow student to check a few lines of code for mistakes.
- Discussing potential approaches to solving the problem.
- Distribution of test cases.

Collaboration with a fellow student needs to be documented. This is as simple as mentioning it in an acknowledgements section in your write up. Please detail the extent of the collaboration as well.

The following are things which do NOT constitute collaboration:

- Copying large chunks of code.
- Copying any part of the write up.
- Using ideas/code without any knowledge of the other party.
- Getting code from a fellow student who is not in this class.
- Getting another student to fix your code.

The actions above are outlines of what is and what isn't collaboration. If you don't know if something would be collaboration please email me, but usually if you are wondering if something would be collaboration it probably isn't. If you take any of the above actions, or similar, it will be considered plagiarism and you will receive a 0. Additionally, normal reporting procedures for academic dishonesty will be followed.

3 Project

This project is to be done in the Python programming language. You program should perform the following tasks:

1. Ask the user to input the name of a file containing a Rush Hour game board.
2. Parses the game board into a usable representation.
3. Apply the A* algorithm to the representation and determine a series of moves that solves the board.
4. Writes a series of moves to a file.

Following the project you will be required to submit a paper at least 1.5 (and no more than 5) pages (excluding acknowledgements) in length (style details below) detailing the heuristic you chose, why it is admissible, how it compares to a heuristic of $h(x) = 0$, and your approach to the problem.

3.1 Board File Format

A Rush Hour board will be given in a CSV file (typically with a .board extension). These files should be considered to be well formed, thus minimal input validation will be required on the part of your program.
Each file's first line will be the dimensions of the board. Each remaining line describes a car/truck which has been placed on the board.
Each line describing a car/truck has the following elements (separated by commas):

1. A unique ID (Integer)
2. The upper left X-coordinate of the car on the grid. (Integer)
3. The upper left Y-coordinate of the car on the grid. (Integer)
4. A V or H indicating that the car/truck is horizontal or vertical. (String)
5. The length of the car. (Integer)
6. A T or F indicating if the car is the goal car (T means it is the goal car). (String)

For simplicity, it is assumed that the upper left corner of the game board is grid cell (0,0) and as you go down the Y value increases and as you go right the X value increases.

3.2 Solution File Format

Your solution file will be formatted as a simple CSV with each line representing a singular move. A move consists of two parts (separated by commas):

1. The unique ID of the car/truck to move.
2. The magnitude of the move.

3.3 Provided Utilities

You will be provided with (and shown how to operate) a program which will visually represent the board and step through a solution you provide. This will assist you in understanding where your code might be going wrong even though it produces a solution.
It should be noted that while you will have access to this code, the code itself does not provide any benefit as a basis for your solution. I know it is tempting to use code that has been provided, but only madness lies that way. I am telling you this now to avoid problems in the future.

3.4 Heuristics

For this assignment you will need to design your own heuristic. This heuristic must be admissible and consistent but cannot be the following:

- $h(x) = 0$
- $h(x) = c$ when not at a goal state, 0 otherwise. Where c is some constant value.

It will be easier to form a heuristic if you choose to only move your cars by a magnitude of 1, but this is not required.

4 Expected Development Time/Other Considerations

This project will take some time and is expected to involve at least 10 hours of development. While testing solutions to the assignment I found that a properly optimized version of the software can find a solution (for a hard instance of a standard Rush Hour problem) in under 4 seconds. If the solution is not properly optimized this process may take more than 10 minutes. On top of this you will also need to develop a method of confirming that your solution is indeed the shortest possible solution (this may require extra development time.)
Your code is required to have the following comments:

- The top of each file should contain: the filename, your name, purpose of the file.
- Above each function there should be a comment describing what the function does, its inputs, and its outputs.
- A comment which describes each step of your algorithm (not each line of code, but segments of code where the meaning may be non-obvious.)

In addition to these comments your code should be well-organized with well-named variables and concise language. An attempt should be made for efficient code where possible.

5 Write Up Requirements

You write up must have the following style:

- 12pt Time New Roman Font
- 1” margins
- Single spaces after periods
- Single spaced

Your write up should include the following information in the top left corner:

1. Name
2. Class
3. Year/Semester

The contents of your write up are up to you, but should include the following:

- Your Heuristic
- Why your Heuristic is admissible
- Why you used the data structures you used
- Comparison between your heuristic and breadth-first search
- Ways your heuristic or implementation could have been improved
- Places where you had trouble with the assignment and how you rectified them
- Acknowledgements of any person you collaborated with

6 Rubric

Item	Points
Code Style and Readability	15
Program Runs	10
Solves Any Given Rush Hour Board	20
Heuristic is Admissible and consistent	20
Write Up	20

7 Deliverables

All code, along with the write-up in either .docx or .pdf format, should be zipped and submitted to Canvas by the due date (listed on Canvas) using the file naming convention <*your_last_name*>_CS360_project_1.zip