A2 – CS4300 Assignment A2 Lab Report

Random Actions in Wumpus World

By Haochen Zhang & Tim Wei
8/29/2017

1. Introduction

Two agents are written for this assignment. The first agent, *CS4300_agent_Astar*, takes random action to find the cell with the gold just like in A1. After reaching the gold cell alive, it will then uses A* algorithm to find the shortest safe way back to the start cell. The second agent, *CS4300_agent_Astar_AC*, will use AC3 to mark the unvisited cell as safe based on the percept it received and A* algorithm to visit the unvisited safe cell; then find its way back to strat cell like agent *CS4300_agent_Astar*. We want to analyse and compare their behaviour on several different boards to answer the following questions:

- Under different boards (from simple to complex which based on how many random decision the Astrae_AC agent have to make), does the use of an arc consistency algorithm improves the survival rate of the agents?
- If the agents survive, does the use of an arc consistency algorithm improves the total steps taken?

2. Method

We implemented 2 agents, one, *CS4300_agent_Astar*, chooses random actions to find the gold, one, *CS4300_agent_Astar_AC*, uses AC3 to do so, and both uses an A* algorithm to find the way back to [1,1]. The A* algorithm uses a marked board, where visited cells are marked as clear and all other cells are marked as pit, and Manhattan distance as heuristic function to search for a lower cost from an initial cell to a goal cell.

CS4300_agent_Astar behave as such:

- Randomly chooses an action and keep track of the current state.
- Marks the cells visited as clear.
- If it moves onto the gold cell, GRAB, and uses the board and A* to prepare the return sequence.
- Once it gets back to the start cell, CLIMB.

CS4300_agent_Astar_AC uses AC3 as such:

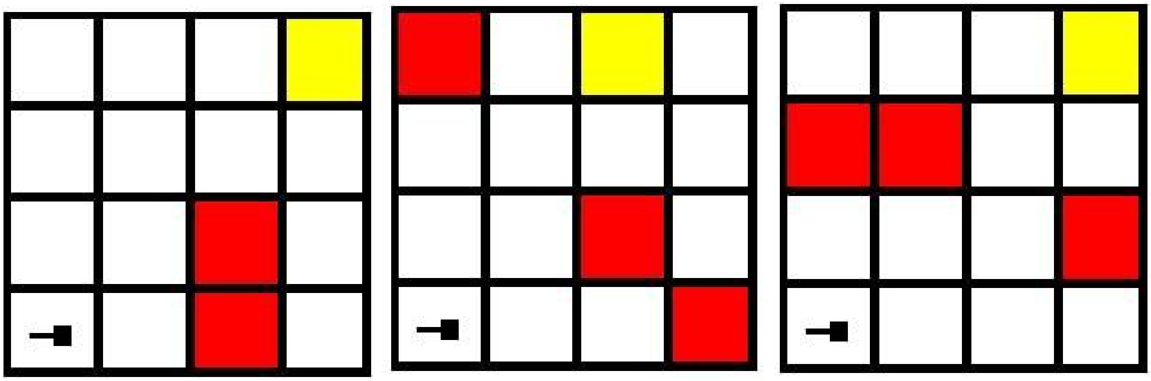
- Cells are represented by a number from 1 to 16. Cells are numbered by this function:
$$Cell\,\# = x + (y-1) * 4$$
Where x and y are the indexes of the cell.
- *G* is the connectivity graph for Wumpus World cells. *G* is a 16x16 boolean array, where each rows and columns represent a cell in the Wampus World, and 1 means that the cells neighbour each other, and 0 means that they are not.
- *D* is the set of labels of the cells. D is a 16x3 boolean array, where the row number represents a cell and the column number represent possible labels, {*C, P, B*}, where *C* means the cell is clear, *P* means there is a pit, and *B* means there is a breeze.
- *P* is the predicate function which implements Wumpus World rules which takes the following arguments:
  - ○ i (int): start node index
  - ○ a (int): start node domain value
  - ○ j (int): end node index
  - ○ b (int): end node domain value

Where a and b are values between 1 to 3 representing {*C, P, B*}. *CS4300_P_no_attack* is used as *P*, which only return 0 if i and j are neighbours, $a = 2$, and $b \neq 3$.

CS4300_agent_Astar_AC's behaviour differ from *CS4300_agent_Astar* before it moves onto the gold:

- Uses percepts to delete labels from *D* appropriately; e.g., if in cell [1,1] and percept(2) is 0 the agent is alive, then delete *B* and *P* label at cells [1,1].
- Uses AC to update *D*.
- Determines if a safe cell exists that has not been visited, and if so uses A* to plan a travel action sequence to get there, otherwise randomly chooses from {Forward, Right, Left}.

We then run both agents 2000 times on 3 different boards and record the chance of the agents surviving and the steps taken until CLIMB for the surviving runs.



(a) Board 1 (b) Board 2 (c) Board 3

Figure 1. Boards Used in the Report.

3. Verification of Program

In order to verify our A* is correct, three boards are chosen to check if our hand developed solutions are the same as the result produced by A*.

The 3 boards we pick is the following graphs. Red cell indicates the cell that is not available while passing into A*. Thus, the A* action sequence will based on the white cells and using the gold cell as the start state. In A*, the fourth column of the solution is the action that the agent should take. Starting the second row, the value of the fourth column (0 is F, 1 is L, 2 is R) should matches the action in our solution.

Our solution are formatted as (x, y, d, g, h, f)

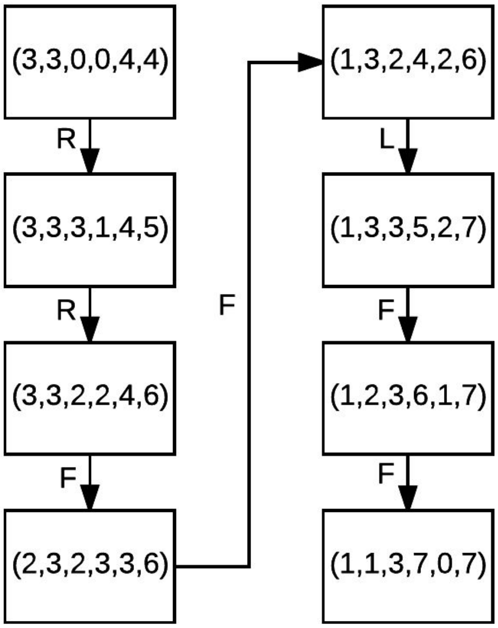
- x: x index of the agent
- y: y index of the agent
- d: direction of the agent
- g: number of action taken
- h: heuristic cost to reach the goal
- f: cost of the node (g + h)

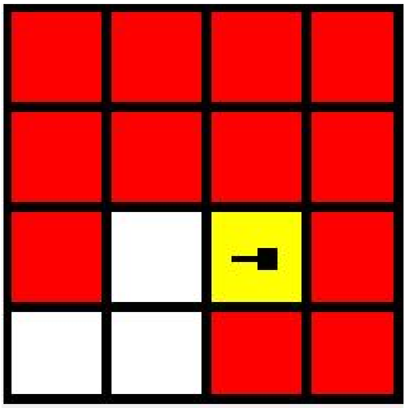
A* solution are formatted as (x, y, d, a)

- x (int): x index of the agent
- y (int): y index of the agent
- d (int): direction of the agent
- a (int): action taken to get this state

In our hand developed solution, we uses the very left branch every time we have a tire in our tree. Our sequence is F, R, L so if our agent need to turn around, it will always choose the branch with turning right. Since A* will chose the last added node if there is a tire and its sequence is F, L, R., it will choose the branch with right turn too. Our solutions match the output of A* perfectly.

Our solution:

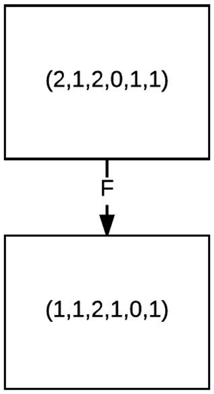


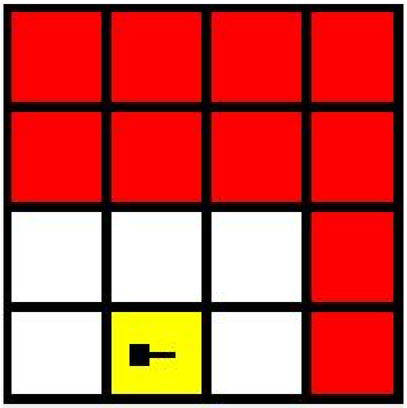


A*'s solution:

- 3 2 0 0
- 3 2 3 2
- 3 2 2 2
- 2 2 2 1
- 2 2 3 3
- 2 1 3 1
- 2 1 2 2
- 1 1 2 1

Our solution:

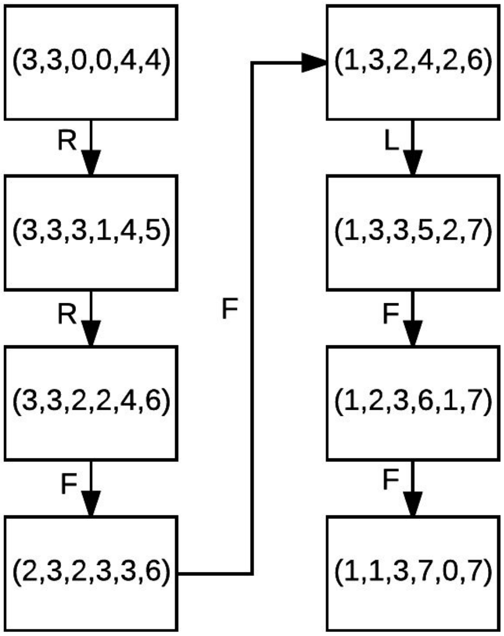


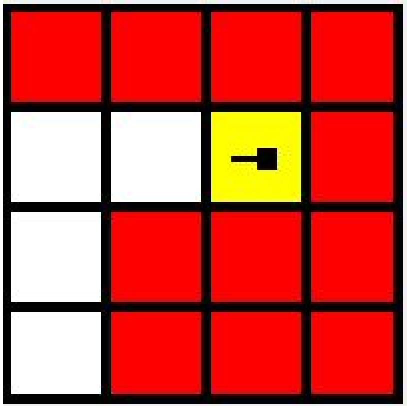


A*'s solution:

- 2 1 2 0
- 1 1 2 1

Our solution:





A*'s solution:

- 3 3 0 0
- 3 3 3 2
- 3 3 2 2
- 2 3 2 1
- 1 3 2 1
- 1 3 3 3
- 1 2 3 1
- 1 1 3 1

4. Data and Analysis

The result shows increases of survival and decreases on both average total steps taken and its variance on all boards using AC3 (fig. 2, 3, 4).

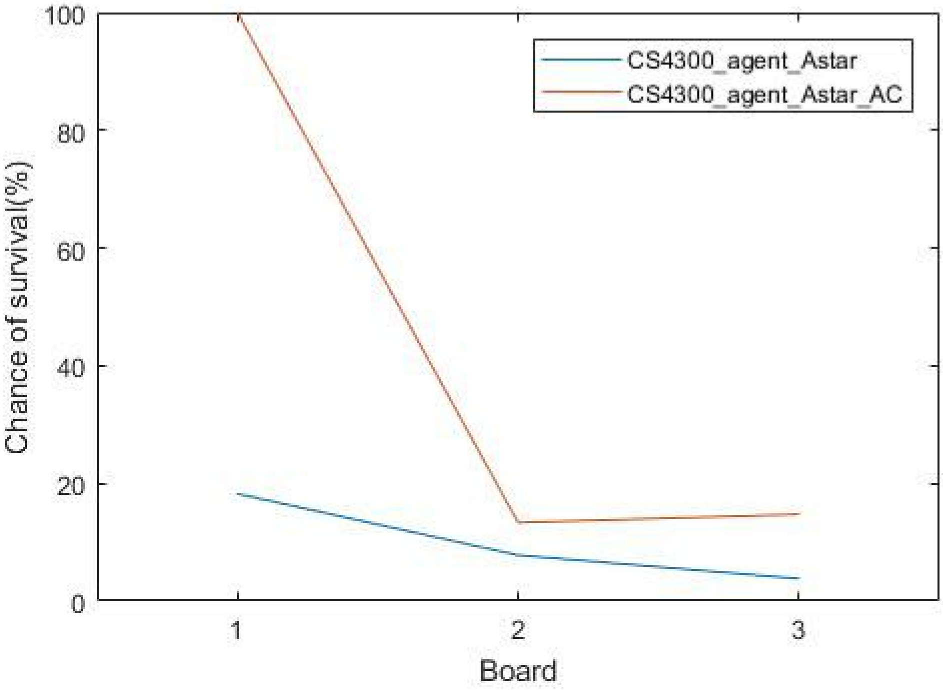


Figure 2. Boards Used in the Report.

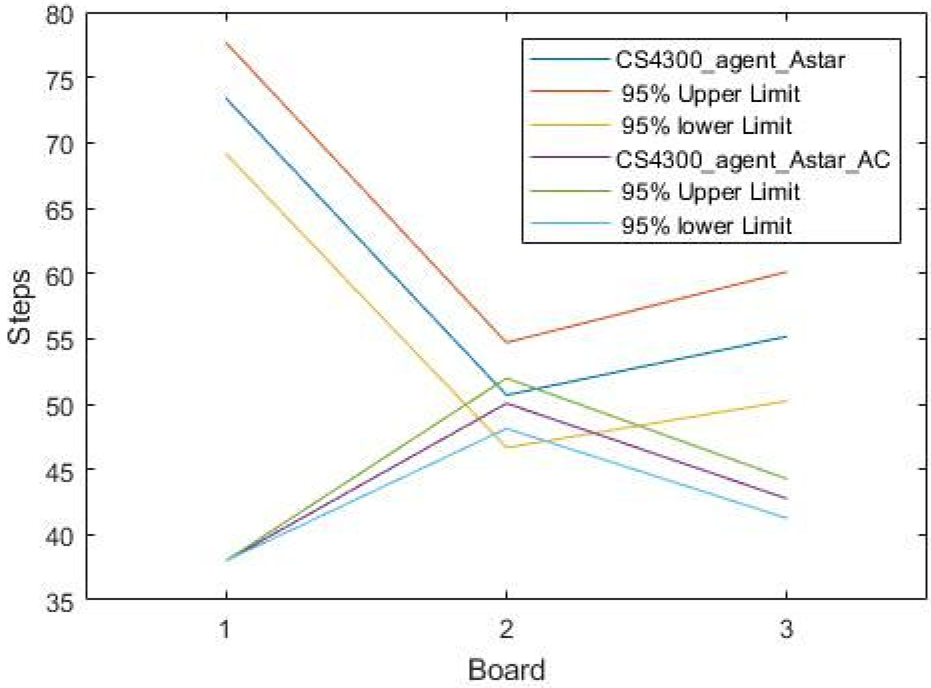


Figure 3. Boards Used in the Report.

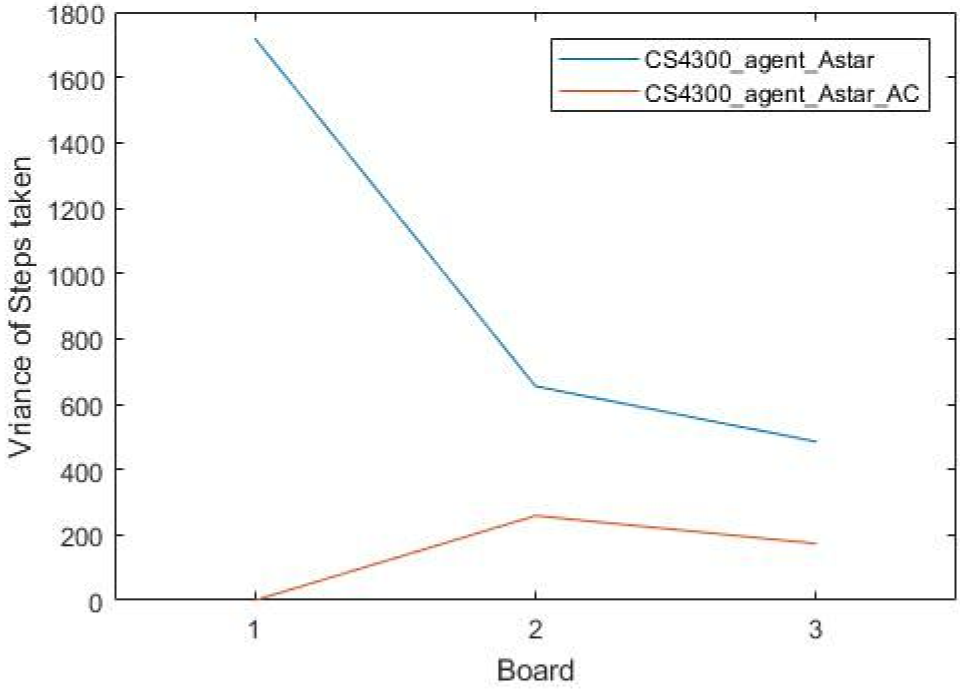


Figure 4. Boards Used in the Report.

5. Interpretation

Because the behavior of ou *CS4300_agent_Astar_AC* is to check all safe cell with the lowest Y value and if the Y value is the same, it will choose the cell with the lowest X value. So we design our first board based on that and the performance of *CS4300_agent_Astar_AC* is much better than *CS4300_agent_Astar*. Because *CS4300_agent_Astar_AC* will always choose the same solution on board 1, its variance is 0.

Based on our board layout, board 2 will cause *CS4300_agent_Astar_AC* to make random choices in two separated section while board 3 only cause one. It should be the main cause that its performance on board 2 and 3 are worse than on board 1. This also makes contribute to board 2 having higher average total steps and variance than board three.

For future work, after Wumpus is added into the board, AC3 algorithm will show more advantages and even more with more complex board. We will be able to have more statistic around AC3 algorithm since there is only breeze and pit that need to be checked.

6. Critique

The number of tested board is strongly restricted by the amount of work required and the time cost for test trials to finishes, making us not able to further analyze the effect of different board layouts. The boards used in the tests are hand designed based on *CS4300_agent_Astar_AC* by us, so there may be biases on our choices, making the result skewed towards favoring *CS4300_agent_Astar_AC*'s performance. A more insightful analyzation can be done with more randomly generated boards.

7. Log

Tim Wei (Section 2, 4, 6)
A total of 6 hours was spent performing the experiment in Matlab.
A total of 5 hours was spent performing the experiment in writing the report.

Haochen Zhang (Section 1, 3, 5)
A total of 5 hours was spent performing the experiment in Matlab.
A total of 4 hours was spent performing the experiment in writing the report.