Introduction to Artificial Intelligence Assignment #1 report

Anton Brisilin, BS18-02 Student,
a.brisilin@innopolis.university

March 10, 2020

1 Assumtions

The description of the Assignment is ambiguous, and require to make several assumtions in order to implement it:

1. Agent can not go out of bounds of field
2. Cost of moving TO cell with human is 0, cost of moving FROM is 1
3. For third algorithm, field is completely unknown and should be explored to find the location of touchdown
4. Agent ALWAYS present on cell (0,0), even if there is no `h(0,0)` in the input file
5. If agent passes in wrong direction in random search, or in third algorithm search attempt fails (in case of third algorithm it means that whole search fails).

2 Search algorithms description

2.1 Random Search

The random search algorithm was implemented as a first one. At each turn agent randomly perform either step or pass. There is almost nothing to say besides that there was eliminated possibility of agent to move to the boundary of field and fail because of it.

2.2 Backtracking Search

The backtracking search tries all of possible paths and looking for the shortest with DFS. However some optimisation I have added few optimisations for it. First one is that backtracking search never visits the same node twice - because shortest path to touchdown never contains loops. The second optimisation is to keep track of already found paths and fail backtracking branches that are leading to path longer than one that is already found.

2.3 My own algorithm

The third implemented algorithm is not standart one. It is greedy algorithm that on each step chooses the most unknown cell around agent, and then agent tries to go there. The algorithm is oriented to exploration, rather than exploitation. The degree of cell's uncertainty is determined by the following function:

$$\begin{cases} D(goal) = & 0, \\ D(ordinary\ cell) = & 1 - \frac{Visible - Unknown}{MaxVisible}, \\ D(cell\ with\ orc) = & 1 \end{cases}$$

In this function $Visible$ is number of cells, visible from the examined one, $Unknown$ is number of cells, visible from examined that are unknown to the agent. $MaxVisible$ is maximal number of visible cells - it is depend only on radius of vision. Agent always choose cell with least value to examine next. If the uncertainty degree of two or more adjacent cells are equal, agent chooses one with cheaper movement cost (i.e. cell with human will be preferred cell without one).

On each step, algorithm keep *Reachable* stack of cells that are not visited yet, but can be visited later. When all cells adjacent to agent are visited, *Reachable* will be popped, and next cell to explore will be chosen from it. Path to this cell is searched with A* algorithm. If the *Reachable* stack is empty, then algorithm fails.

There is the limitation put in A*: it can visit only cells that were already visited by the main algorithm. Other cells are considered as obstacles. This limitation was put because A* operates in fully observable environment, whereas Assumption 3 states that the field is unknown. As a heuristics for A* there was chosen manhattan distance to the goal.

When the goal is reached, algorithm ends with success and path to the goal point is returned.

The algorithm is not capable of performing passes, because they does not make sense in case of vision with radius 1 (there is no point in blind pass, because in this case search fails).

3 Testing algorithms on several maps

3.1 Maps description

There were generated several maps of diferent types. For maps generation there was written Python script, that produces file with Prolog predicates, that can be given to Prolog interpreter as input knowledge base.
Tested maps are listed below in text format: they are all 20*20 cells, with $(0, 0)$ as the top-left cell, X axis directed to right, Y axis directed to the bottom of page.

Legend

- `O` - cell with orc
- `H` - cell with human
- `T` - cell with touchdown point
- `.` - empty cell

The following next pages contains considered maps together with time results and statistical analysis of time results.

Map 1.

```
. H H H O . O H O H H O . . H H . H . .
O . O . O O O H . . . H O H . H O H H H
O . H . H H . . H H H H O H H H . H H O
O H O H O . O . H O O H T H . O H O H H
H . H O O O H H O O O H H . O H . O H H
. . O H H O O H O H . . H . H O . H H .
. H . O . H H H . . H O H O . H H H H O
O H . O O H H H O O . H O . . O H . O H
O H H O . H H . H H O H H H H . . . . .
. O H H O H H . H O H H H H O . H . . H
H O H H H O . O H O H . O H O H H H H O
H O O . O . H O O O H H . H O . . H H O
H H . O O H H H O H . O . H H H H O . H
O O H H . H . O O H H O O H . . H . O O
O O O O . O O H . H H O O H . H H H O .
O O O . O . O . . . H H H O H H H O . .
H O H . H O H O O . O . O O H O O H . H
O H H O H H . . O H H H H H . O . O H O
H H H H . H H H O H H H H . H O O H O H
. H H O . O O O O . O H O O . H H . . H
```

Solvable - YES.

Search	Number of attempts	Best time	Mean time	Variance
Random Search	100 × 100	– (no solutions found)	–	–
Backtracking	100	0.6194s	0.5819s	0.01134
My algorithm	100	0.01466	0.0162	9.83e-6

Map 2.

```
. O . . . O . . . H O . O H . H . . H .
O H . . H H H H . . . H O . H H . H H .
. . O H . O H O O . H . . . H H H O . T
. . . O . . . H H . . H H O . . . . . H
O H O O . . . H O O O O H H . O H . . H
. O . . O . . . H O . O . . . . . . H O
. . . H . O . O . O H O O O H H . H H O
H . H . . . . H O . H H . H O . O H . .
H . . . H . O H H O O O H O H . H . . H
. . O . H . . O . . H O O . . H O H . .
H . . . H H O . . O . O . . H H . . . .
. O . O . H . . . O . O . . . H O . . .
O O . H H . . . H O O H O H . H . H O .
. . . H . . . . . O H H . H . . H . O .
O O H H O . O . . H . . H . O H . H O .
. H O H O H H . . O O O . . O . . H H .
. . O O H O . . . . . . . . O . O . . H
O . O H . . O . . O . H O . . . . . O O
. O O . . H . O . O . O H . . . H H H .
H O . . H . . H . . . O . . . . O . O H
```

Solvable - Only with passes.
Results:.

Search	Number of attempts	Best time	Mean time	Variance
Random Search	100 × 100	– (no solutions found)	–	–
Backtracking	100	0.0038s	0.0043s	7.3e-7
My algorithm	100	– (no solutions found)	–	–

Map 3.

```
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . T
```

Solvable - Yes.

Search	Number of attempts	Best time	Mean time	Variance
Random Search	100 $\times$ 100	– (no solutions found)	–	–
Backtracking	100	0.0004s	0.0006s	9.18e-8
My algorithm	100	0.1317	0.1405	0.00029

Map 4.

```
. . . . . . . . . . . . . . . . . . . .
. T . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . .
```

Solvable - Yes.

Search	Number of attempts	Best time	Mean time	Variance
Random Search	100 $\times$ 100	0.0025s	0.0032s	7.93e-7
Backtracking	100	0.0552s	0.0705s	0.0003
My algorithm	100	0.0002s	0.0002s	1.44e-8

Map 5.

```
. . . . H . H H . . . . H . H . . . . .
H H . . H H . . . . H . . . H . H . H .
H T . . . . H . . H H . H H . . . . . .
. H . H H . H . . . . . H . . H . . . H
. . . . H . H H . H . H . H . . . H . .
. . . . H H . . H . . H H . . . H H . .
. H H H . . . H H . . H . H . . . . H .
H . . H H H . H H H H H . . . H . . H .
H . . . . H H . H . . . . . . . . H H H
H H . H H . H . . . . . H H H . . . . H
. . . H . . H H H . H . . H H . H . . H
. H . H . . H . H . . . H . H H H O . .
H . . H H . . H H . . H . . . . H H . H
. . . . H . . . . . . H . H H . H . . .
. H H . H H . . . . . H H . H . . . . .
H H H . . . . H . . . . . . . . H . H .
. H . H . . . H H H H . H H H . . . . H
. . . . . H . . H O . . H . H . . . . H
. H . H . . H . . . H H H H . . . H . H
. . H . . . . . H . . H . H . H H . . H
```

Solvable - Yes.

Search	Number of attempts	Best time	Mean time	Variance
Random Search	100 × 100	0.0416s	0.1407s	0.0032
Backtracking	100	0.07202s	0.07648s	4.1015e-6
My algorithm	100	0.0003s	0.0004s	5.34e-8

4 Algorithms comparison

Random search can not be considered as best algorithm to find paths. It is not reliable (not guarantee, that path will be found, if one exist). It has not really big chances to find solution on orc-dense and/or human-sparse maps. I recommend to use Random search only in case when touchdown is really close to start point, because in case of several lucky steps it can found best path to it faster than backtracking

Backtracking search guaranties that path to the point will be found, also it can be used to obtain optimal path to the goal. But on orc-sparse maps, there are too many possible paths, so search for an optimal path can take unacceptably long time (hours, days...).

My algorithm works pretty well in case of both orc-sparse and orc-dense maps. Also it was designed in order to be utilised in partially observable environments. It works faster than even searching FIRST path with backtracking, in general case. However it did not found path on 2nd map, because, as was stated above, it does not capable of doing passes, and 2nd map is unsolvable without passes. Also, in 3rd map my algorithm works slower, because of specific location of goal point: according to my heuristics, corner points has lowest priority to search in them.

5 Improved Vision

In second part of the assignment we was asked to compare same searching techniques but with improved vision. However, for first and second algorithms it does not really make sense, due to their nature: random search does not consider vision at all, but backtracking, in contrast, works in fully observable environment.
In case of third algorithm improved can make solvable maps, that can not be solved without passes, because there is meaning in making passes now (when you see the person who you passing to).

6 Source code

Prolog source code files, together with Python maps generator are available on my Github repository, the link: https://github.com/Mexator/IAI-Assignment.
You can read `README.md` on the repository, to obtain instructions of how to run the assignment.