520 INTRO TO AI Fall 2020 Assignment 1

Abhishek Bhatt - RUID 198009864
Po Yuan Huang - RUID 195003420 ()

October 13, 2020

1. Part 0 - Setup your Environments

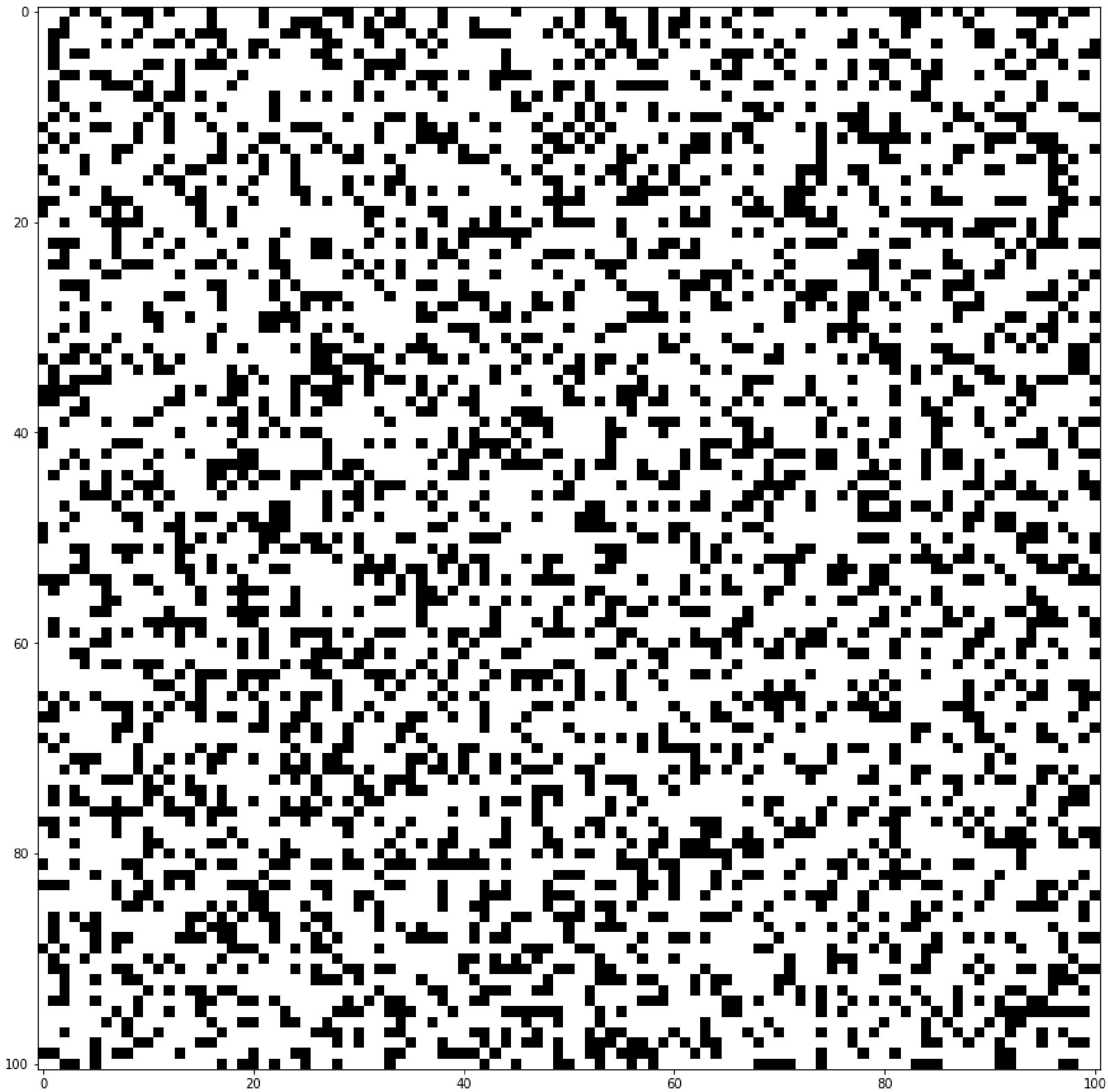
In the Python notebook, we generate and visualize a n by n maze (n=101), in which each cell has probability $p = 0.3$ to be blocked. The blocked cells correspond to entries "1" and the unblocked by "0" entries. Figure 1 shows the maze generated by this logic, on which the experiments described later are performed.

The agent will find a path from "source" to "goal" in this maze. If the chosen "start" and "end" are blocked cells, we automatically set maze[0][0] and maze[n-1][n-1] (which are always unblocked by default) to be our source and goal cells respectively. We will also demonstrate how the agent explores a mental maze (where we only know the source cell initially), in order to move along a path from source to goal on the original maze. If no such path exists, the program finally exits with the message 'the goal is not reachable from the given source'.

2. Part 1 - Understanding the methods (a)

`Explain in your report why the first move of the agent for the example search problem from Figure 8 is to the east rather than the north given that the agent does not know initially which cells are blocked.`

Figure 1: Maze



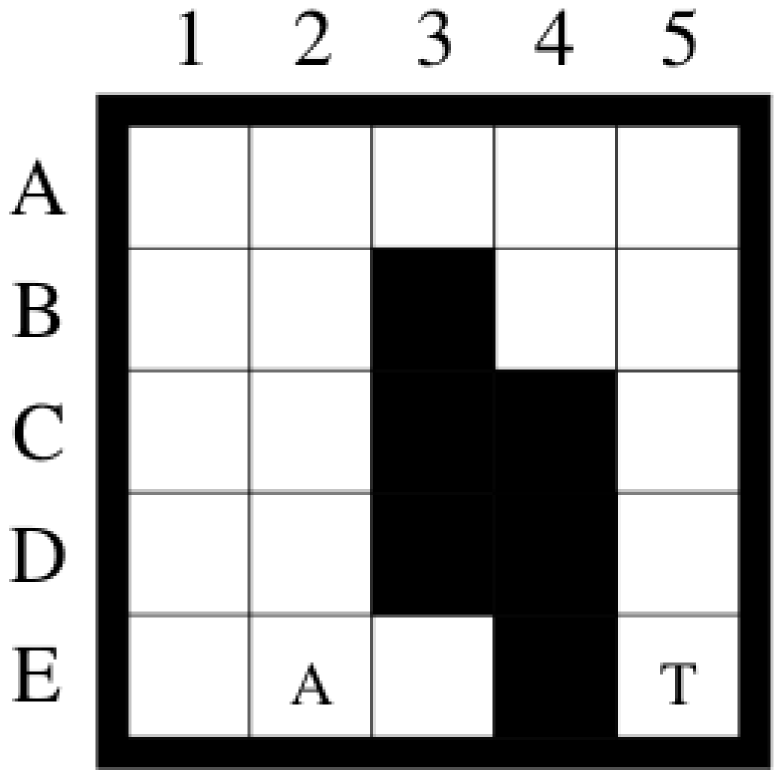


Figure 8: Second Example Search Problem

For the example in the given figure, the neighbor cell in the north (D2) has g-value=1 from the current (source) cell, and h-value=4. Similarly, for neighbor cell to the east (E3), g-value=1 and h-value=3. Since A* search always finds a path such that f-value is minimized with each move, the agent will move to E3 with f-value = g-value + h-value = 1 + 3 = 4, rather than D2 with f-value = 1 + 4 = 5.

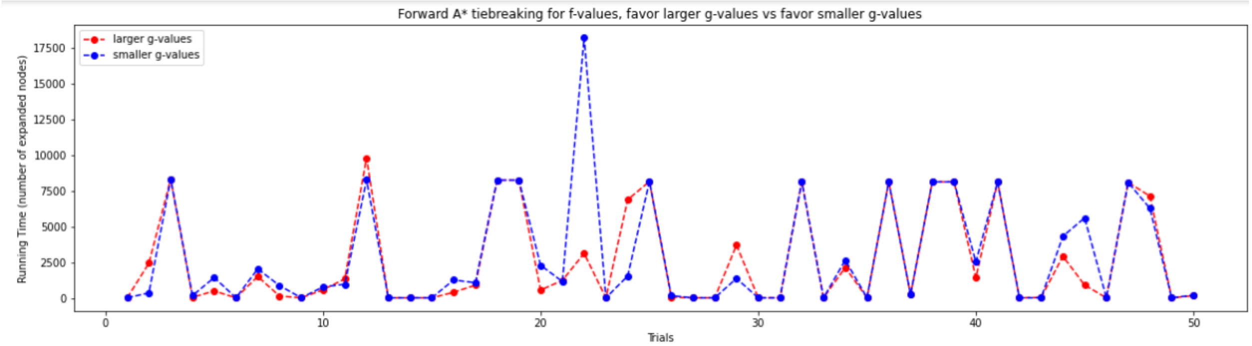
3. Part 1 - Understanding the methods (b)

`Give a convincing argument that the agent in finite grid-worlds either reaches the target or discovers that this is impossible in finite time.`

If a path exists from source to goal, in each repeated call to A* search the agent will find a new path from current cell to the goal. The agent will not follow any previously discovered paths since it now knows that they contain a blocked cell. Thus, the exploration will not continue indefinitely and the search exits when goal cell is encountered.

If goal is not reachable from the source, the agent will reach a cell such that there are no neighbors to add to open list for further expansion. The search again exits when open list is emptied. Thus the search will

Figure 2: The Effects of Ties




```
# larger g-values

print('avg number of nodes expanded for larger g-values', sum(run_time1)/num_trials)
```

```
avg number of nodes expanded for larger g-values 2605.16
```

```
# smaller g-values

print('avg number of nodes expanded for smaller g-values', sum(run_time2)/num_trials)
```

```
avg number of nodes expanded for smaller g-values 2917.54
```

return that the goal is not reachable from the given source in finite time.

Since the grid world is a finite space, the number of unblocked cells in the maze is also finite. In any call to A* search, the agent finds a path from one unblocked cell to the goal (which is also unblocked). If the total number of unblocked cells in the maze is k, there are $\binom{k}{2}$ possible combinations of source and goal. As a result, in the worst case, the agent will make number of moves in the order of magnitude of k^2. Thus, the number of moves of the agent until it reaches the target or discovers that this is impossible is bounded from above by the number of unblocked cells squared.

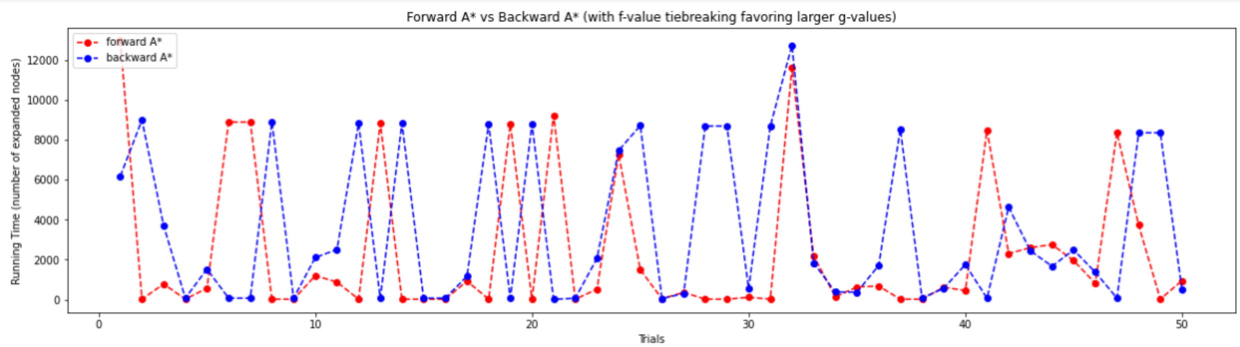
4. Part 2 - The Effects of Ties

In our experiment, we performed 50 trials to compare repeated forward A* search for the same randomly chosen source and goal on the same 101x101 maze, such that in one run we break ties for f-values in the favor of larger g-values and in another run we break ties in favor of smaller g-values.

The search with tie-breaking in the favor of larger g-values performed better than that in the favor of smaller g-values. The average number of nodes expanded for larger g-value priority is 2605.16 compared to 2917.54 for smaller g-value priority. Figure 2 shows the observations.

Explanation : When the agent encounters cells having the same f-value, picking larger g-value corresponds to moving further away from the current cell, or equivalently closer to the goal, than when picking the cell with smaller g-value. This tiebreak strategy will, therefore, result in lesser nodes being expanded to reach the goal during the search.

Figure 3: Forward vs Backward




```
# forward A*
print('avg number of nodes expanded for forward A*', sum(run_time1)/num_trials)
```

```
avg number of nodes expanded for forward A* 2404.82
```

```
# backward A*
print('avg number of nodes expanded for backward A*', sum(run_time2)/num_trials)
```

```
avg number of nodes expanded for backward A* 3479.92
```

5. Part 3 - Forward vs. Backward

In our experiment, we performed 50 trials to compare repeated forward A* search with repeated backward A* search, for the same randomly chosen source and goal on the same 101x101 maze, such that we break ties for f-values in the favor of larger g-values in both cases.

The search with forward A* performed better than with backward A*. The average number of nodes expanded for forward A* is 2404.82 compared to 3479.92 for backward A*. Figure 3 shows the observations.

Explanation : When exploring from current node towards the goal, based on previous searches, the mental maze near the current node is similar to the original maze. On the other hand, no such information is available when exploring from goal towards the current cell. This results in expanding more nodes in backward search than in forward search.

6. Part 4 - Heuristics in the Adaptive A*

`Prove that the Manhattan distances are consistent in grid worlds in which the agent can move only in the four main compass directions`

(a) Consider h(s) - h(next) where next is the successful state that the agent takes an action from s i.e, succ(s,a). The four directions will allow us to have $x_s = x_{next}$ or $y_s = y_{next}$. W.l.o.g, let $y_s = y_{next}$

$$
\begin{aligned}
h(s) - h(next) &= |x_s - x_{goal}| + |y_s - y_{goal}| - |x_{next} - x_{goal}| - |y_{next} - y_{goal}| \\
&= |x_s - x_{goal}| - |x_{next} - x_{goal}| \\
&\leq |x_s - x_{next}| = 1 = c(s, a)
\end{aligned} \tag{1}
$$

We get $h(s) \leq h(next) + c(s, a)$. Therefore, the Manhattan distances are consistent in grid worlds where the agent can only move in the four main compass directions.

`Prove that Adaptive A* leaves initially consistent h-values consistent even if action costs can increase.`

(b) Denote n = succ(s,a). The heuristic is consistent if the triangle inequality holds. The question can be separated into three cases:

Case1: both s and n are expanded
$h_{new}(s) = g(s_{goal}) - g(s)$and $h_{new}(n) = g(s_{goal}) - g(n)$. From the current state s to n, g value of n may be different so potentially there exists a shorter path $g(n) \leq g(s) + c(s, a)$ than the A* search.

$$
h_{new}(s) = g(s_{goal}) - g(s) \leq g(s_{goal}) - g(n) + c(s, a) = h_{new}(n) + c(s, a) \tag{2}
$$

Case2: s is and n is not expanded
Then $h_{new}(n) = h(n)$and $f(s) \leq f(n)$.

$$
\begin{aligned}
h_{new}(s) &= f(s_{goal}) - g(s) \leq f(s) - g(s) \\
&\leq f(n) - g(s) = g(n) + h_{new}(n) - g(s) \\
&\leq g(n) + h_{new}(n) - g(n) + c(s, a) = h_{new}(n) + c(s, a)
\end{aligned} \tag{3}
$$

Case3: s is not expanded
Note that the h is a non-decreasing function w.r.t to time for the same state. So $h_{new}(s) = h(s) \leq h(n) + c(s, a) \leq h_{new}(n) + c(s, a)$.

7. Part 5 - Heuristics in the Adaptive A*

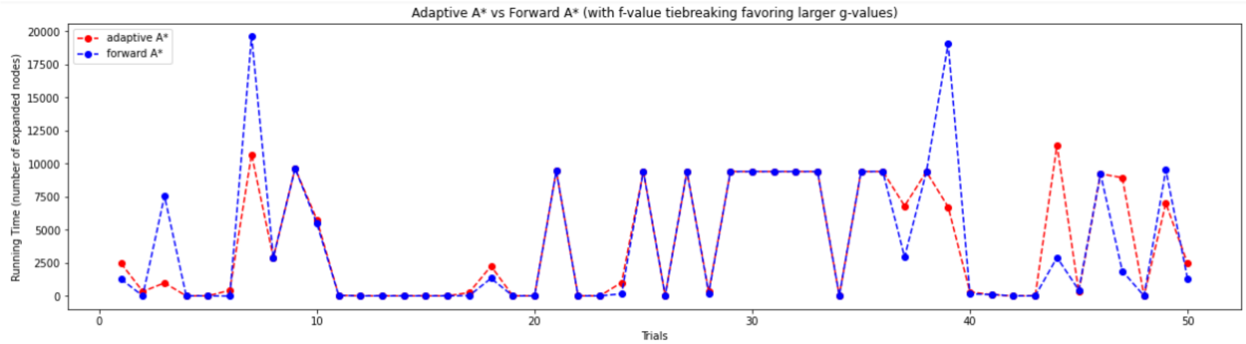
`Compare Repeated Forward A* and Adaptive A* with respect to their run-time.`

In our experiment, we performed 50 trials to compare adaptive A* search with repeated forward A* search, for the same randomly chosen source and goal on the same 101x101 maze, such that we break ties for f-values in the favor of larger g-values in both cases.

The search with adaptive A* performed better than that forward A*. The average number of nodes expanded for adaptive A* is 3891.74 compared to 3998.76 for backward A*. Figure 4 shows the observations.

Explanation : In theory, adaptive A* will be much faster than forward A* since the adaptive A* will expand no more states than forward A* does. The reason is the improved heuristic $h_{new}(n) = g(s_{goal}) - g(n)$ that is a better estimation over Manhattan distance. This is because Manhattan distance does not take blocked cells into consideration, meaning a smaller h-value with blocked cells along the way to goal actually has

Figure 4: Adaptive vs Forward




```
# adaptive A*
print('avg number of nodes expanded for adaptive A*', sum(run_time1)/num_trials)
```

```
avg number of nodes expanded for adaptive A* 3891.74
```

```
# forward A*
print('avg number of nodes expanded for forward A*', sum(run_time2)/num_trials)
```

```
avg number of nodes expanded for forward A* 3998.76
```

pretty large h_{new} values. The observations verify this theoretical reasoning.

8. Part 6 - Statistical Significance

`Describe one of the experimental questions above exactly how a statistical hypothesis test could be performed and to determine whether they are systematic`

We shall conduct Paired Two Sample t-test in comparison of the two algorithm. Let μ_* denote the average number of expanded nodes for Strategy or Algorithm*. Take Adaptive A* vs Forward A* for example. Since they are only two search algorithms executing on the same simulated mazes or, statistically speaking the same population, We are going to test the difference between the mean of expanded nodes for adaptive and forward search.

$$H0 : \mu_{adp} = \mu_{fwd}$$
$$H1 : \mu_{adp} \neq \mu_{fwd}$$

We compute t-value $\frac{\widehat{X_{adp}} - \widehat{X_{fwd}}}{s_p\sqrt{(\frac{2}{n})}}$ where $s_p = \sqrt{\frac{s^2_{adp}+s^2_{fwd}}{2}}$ the pooled standard deviation. Once we set up confidence level α. We can look up p-value from Student t-table using t-value and d.f $= 2n - 2$. Finally, we say H0 is rejected if $p \leq \alpha$, meaning the two search are not systematic and vice versa if $p > \alpha$.

9. References

[1] Adaptive A*. *http://idm-lab.org/bib/abstracts/papers/aamas08b.pdf*

[2] T-test *https://en.wikipedia.org/wiki/Student%27s_ t-test*

[3] Statistical significance. Cohen, Empirical Methods for Artificial Intelligence
`https://www.researchgate.net/publication/216300475_Empirical_Methods_for_Artificial_Intelligence`