George Maroun CS 4641- Guzdial
Assignment #4

Reinforcement Learning Report

Chosen MDP's:

For the purposes of this assignment, I used the RL_Sim Framework (as referenced in the README attached) to load in and execute various grid-world mazes ran over Value & Policy Iteration. The implementations I chose as the MDP's were a small, medium, and large size world. Each grid-world consists of an N X M maze in which there is a starting square and orange reward(s) terminating location. With a reward of -50 for running into walls on the edges, as well as -1 for staying in a particular state, the goal is to reach the termination as quickly as possible to minimize loss. There is also a free parameter (PJOG) which describes the probability that the agent will not perform the originally intended task.

Maze 1, the smallest maze, is a 3 X 3 maze (9 states) with a protruding from the middle and one reward state. It will be interesting for two reasons: I chose Maze 1 because of its simplicity when stacked up against the complex grids, and I want to see how well and how fast it converges given such a simple world.

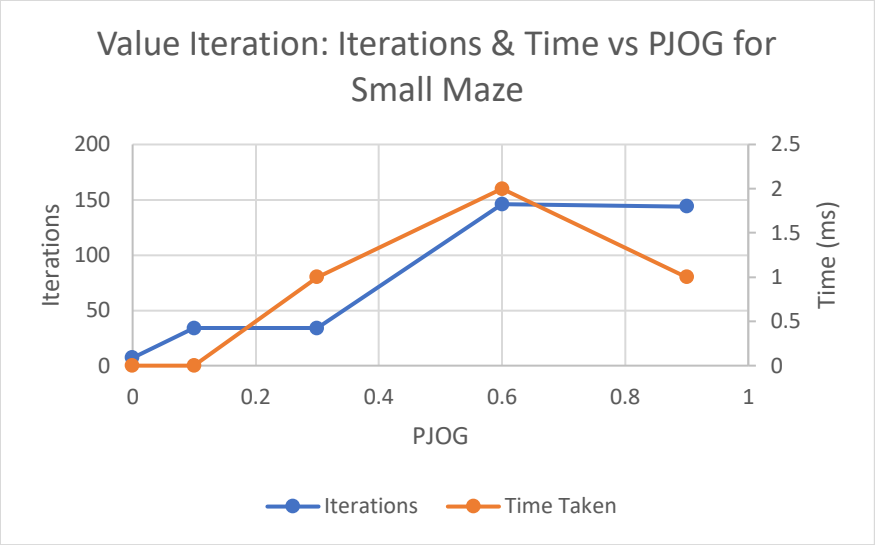
Maze 2, a medium maze, is a 10 X 10 maze (100 states) with various walls scattered throughout, as well as multiple reward states near each other. Not only did the size just increase from the original world by and order of ~11, but we have no introduced a multi-goal environment which I made a point to include to see visually how much that changes the algorithm if at all. It will be interesting to see how much the agent's behavior changes here.

Maze 3, the biggest maze, is a 45 X 45 maze (2,025 states) with, again, many different wall structures blocking viable paths, and a reward state near the bottom. We are now reverting to a single-reward environment, however this is arguably the first world in which the optimal path is non-decipherable to the human eye, so there is something to learn about how the algorithm plans on approaching it. It will be interesting to see how all of these stack up against one another.

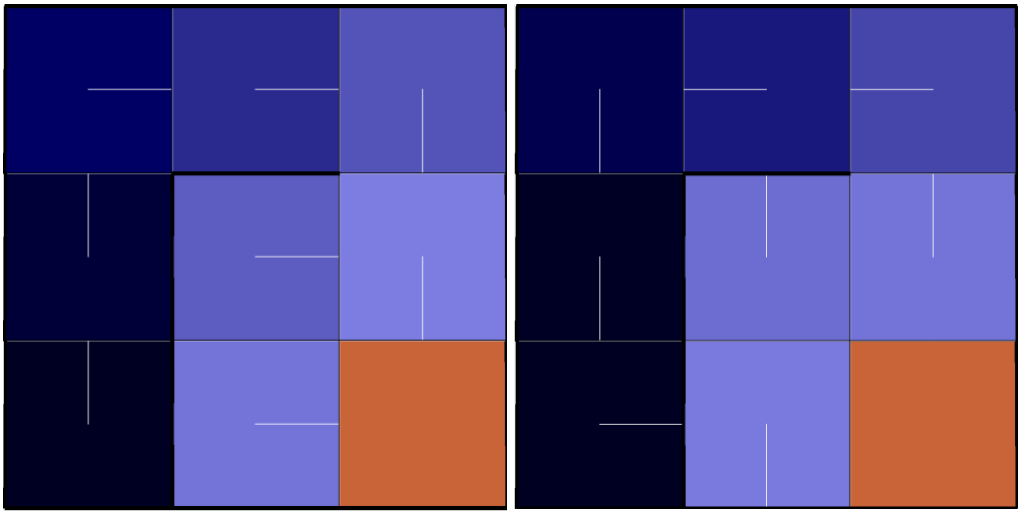
Value Iteration:

In value iteration, you start with a random value function and then find a new (improved) value function in an iterative process, until reaching the optimal value function.

Smallest Maze- smallest2.maze



Under value iteration for the smallest maze, there is a clear convergence trend going on in the graph. The underlying relationship is that as the amount of noise in the model is increased (modeled by PJOG), so is the amount of iterations needed for the algorithm to converge. This makes sense, notably, because the noise takes longer to be fleshed out of the algorithm. The correlation is not direct but strong between iterations and time taken. There is a turnaround point around PJOG=.6, which could hint at the mathematical nature of this parameter. Before we reach a value of .9, the intended action is most likely to be chosen, but when we cross this threshold, the intended action is taken only 10% of the time (1-PJOG). Effectively, there is an inflection in which the noise it being mitigated with a larger PJOG value as the intended action becomes linearly less likely.



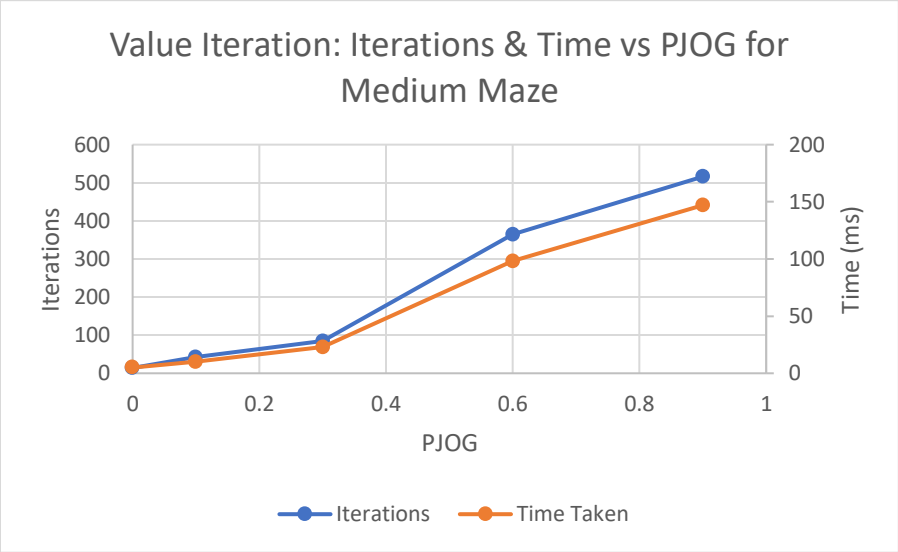
PJOG = .3 PJOG = .9

This point is further solidified looking now at the visual output of value iteration. As a quick note, these plots have individual lines representing the optimal decision to move to the next space (up, down, left, right) to maximize reward. Also, the color denotes become darker for the higher-value, worse states. We can see with the left grid that the solution for each space is nearly always the truly optimal one, which happens because the level of noise is not too blaring in the algorithm. The optimal actions on the right, however, flip and become rarely to never the

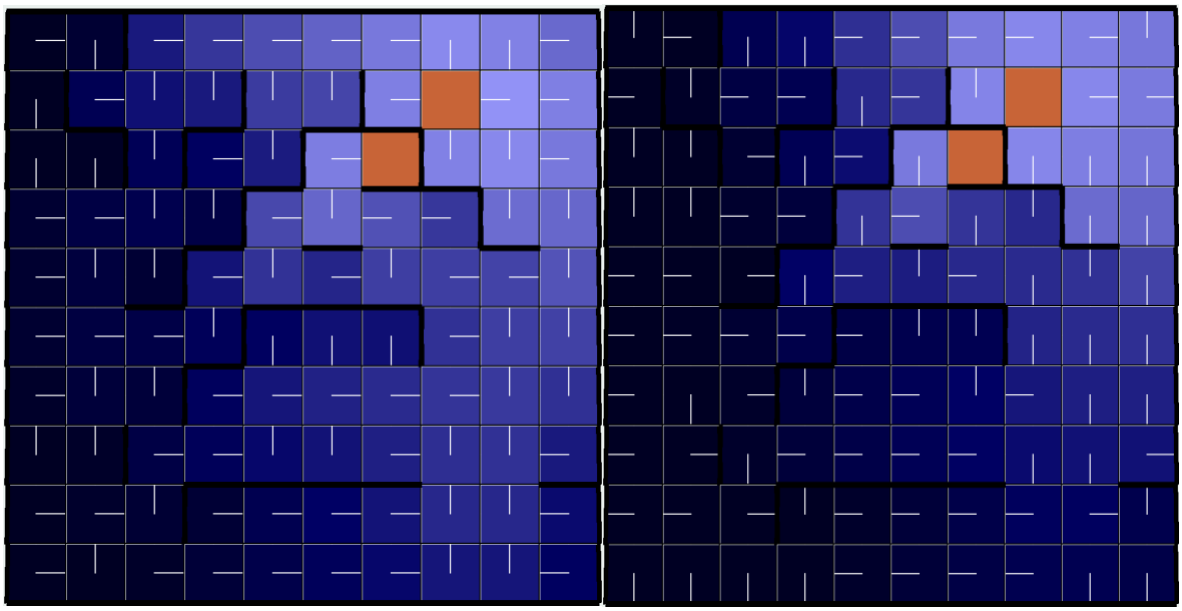
intuitively correct actions mirrored in the latter graph. I suspect that even though the algorithm reaches peak iterations before this time, because PJOG is so high we never have a chance regardless to even pick the optimal which is part of why is has less of a problem converging.

Nonetheless, this grid-world is of trivial size, so it will be interesting to see what happens as we move into bigger spaces. The color scheme seems to be static across noise levels because there is not much room for improvement/error, but this small size is integral in visualizing how the output becomes interesting within the same algorithm.

Medium Maze- medium_complex_multigoal.maze



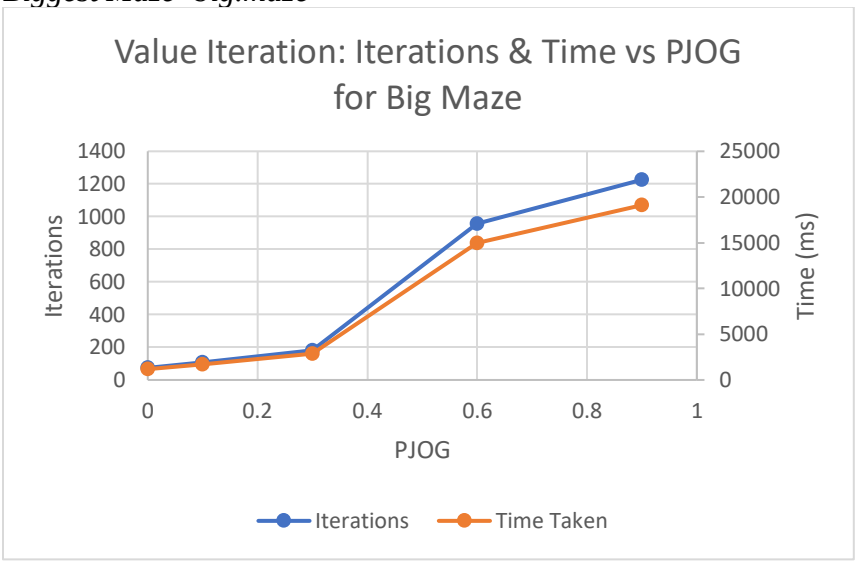
Now the mechanics of the algorithm begin to become interesting seeing as the convergence point that we saw with the Maze 1 is no longer the case and the PJOG/iteration relationship continues to rise. This is quite interesting and may well be attributed to the multi-goal orientation of the grid which is somehow contributing to the noise. Albeit, we can see with iterations and run time the effects of a larger, convoluted grid-world as compared with Maze 1 which ran instantaneously and hit a max iteration of ~145. In this case, we reach over 500 to converge at the optimal value, something we largely expect.



PJOG = .3 PJOG = .9

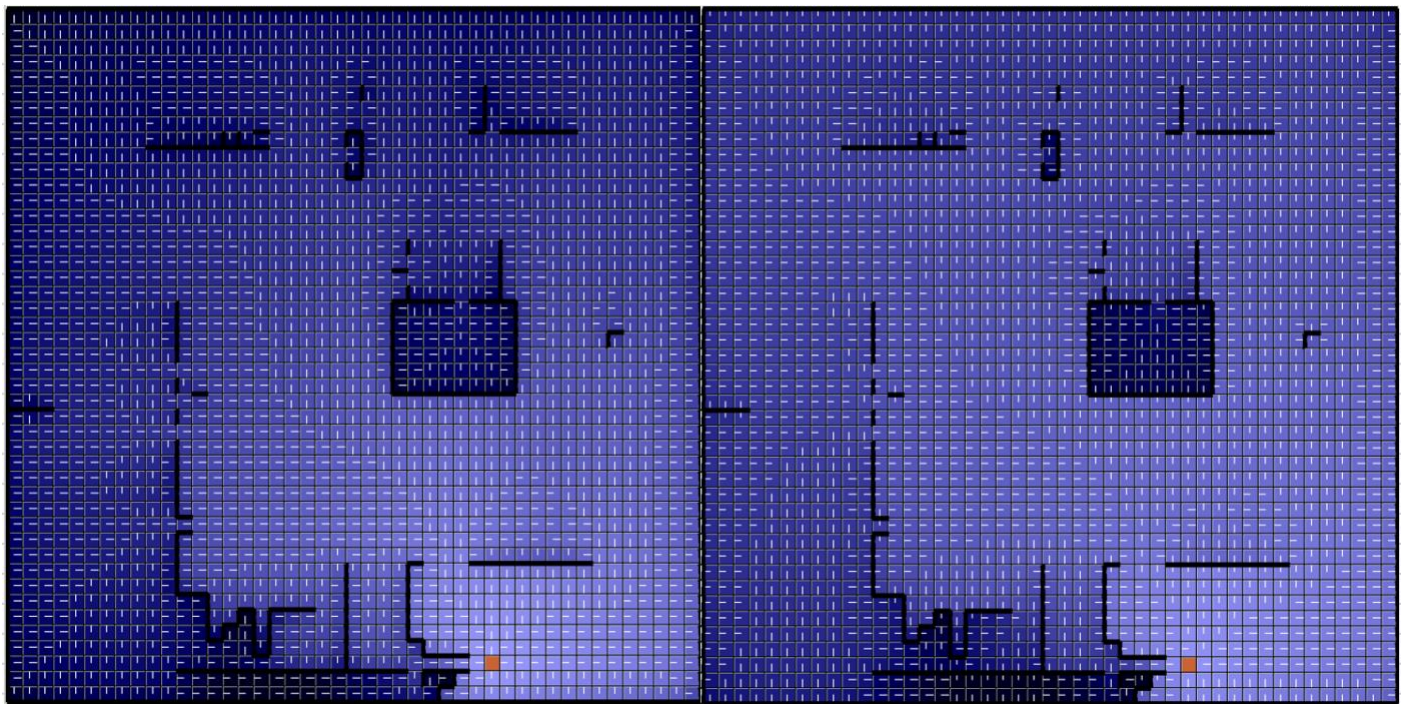
Even with a larger search spaces, we see once more that the optimal moves are more readily manifested on the left plot, attributed to the lack of relative noise. This time, however, the right plot is significantly darker in more of the plot, denoting its lesser inclination to converge on optimality. It becomes extremely dark in the case where the action is by a bottom wall on the right, indicating how easily the noise can trap the algorithm, and increase the amount of time taken.

Biggest Maze- big.maze



This side-by-side graph is perhaps the coolest to visualize since the correlation between time taken and iterations is near identical. Again, we see a lack of inflection with regards to the amount of iterations needed for the algorithm to converge as we increase the random chance (PJOG). Because of the sheer size (even with only 1 reward) value iteration takes ~133 more times longer to run than the complex medium graph, much longer than the simple graph. The

amount of iterations is fairly intuitive given the 45^2 grid, coming in at 1200 to converge at the highest.



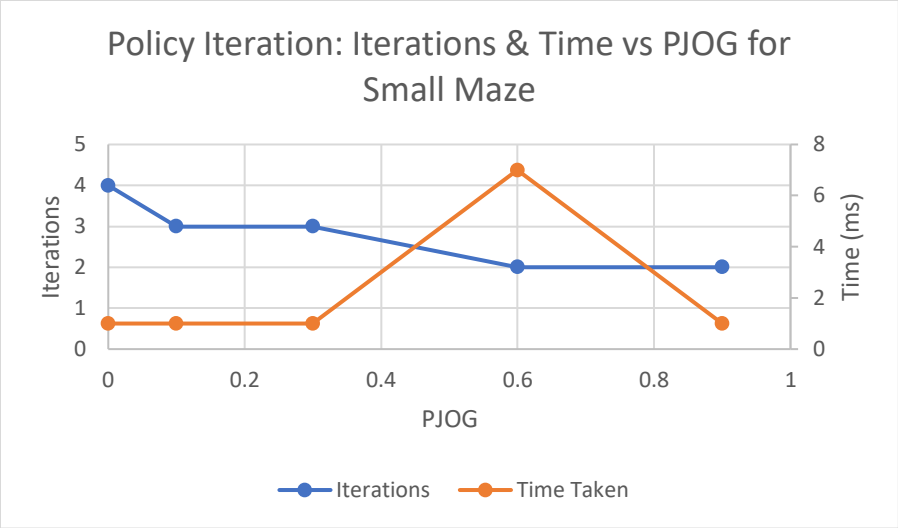
PJOG = .3 PJOG = .9

If you localize on the middle of each plot in order to understand what its local decision process was, you can easily see the manifestation of the biggest maze losing its overall optimality. What we notice again as in Maze 2 is how easily a high PJOG value becomes trapped with only one clear exit point. You can also see the gravity of the effect of introducing noise when looking at squares immediate to the reward, and how yet the algorithm strays from the obvious decision.

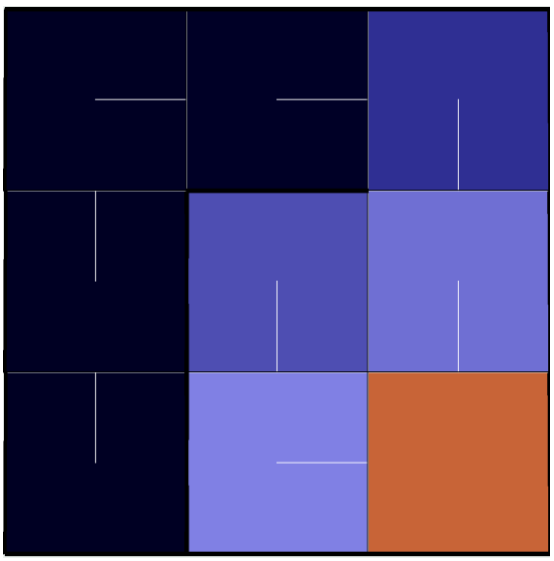
Policy Iteration:

In policy iteration, you start with a random policy, then find the value function of that policy (policy evaluation step), then find a new (improved) policy based on the previous value function, and so on.

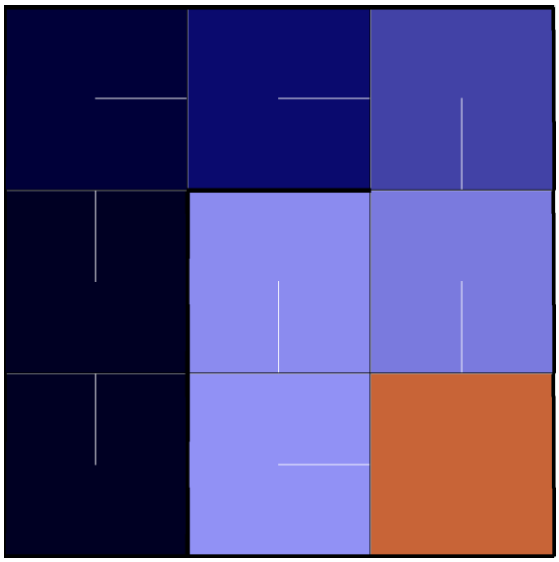
Smallest Maze- smallest2.maze



Policy iteration led to similar times, but much smaller amount of iterations for the same grid size than did value. It is interesting, though, because it becomes harder to flesh out the relationships between parameters and output that we observed in the first run through. Not only is the algorithm now minimally affected by changes in the noise level, there seems to be an awkward spike in time taken at PJOG=.6, which is hard to assign meaning given the overall graph trend and limited variables.



PJOG = .3



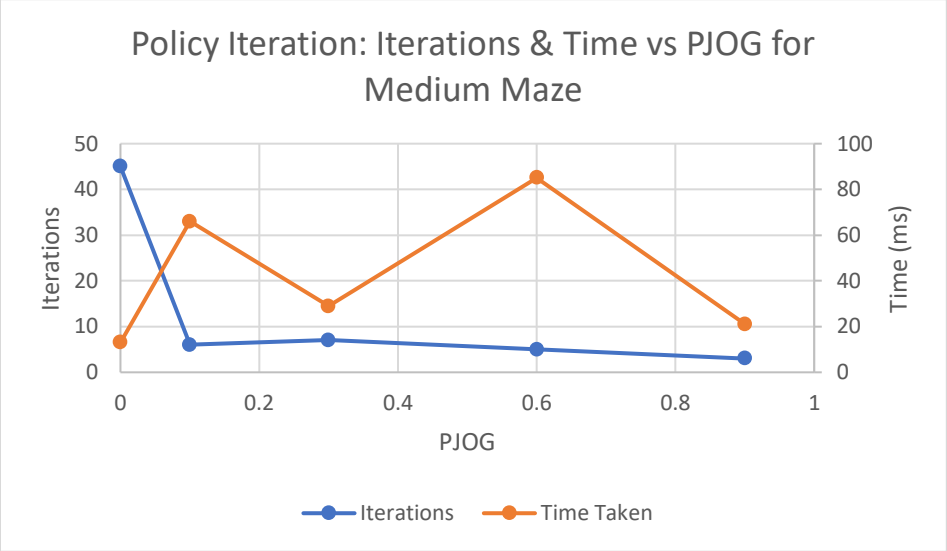
PJOG = .9

What now become very interesting is how negligible of an effect the level of noise has on policy iterations decision. The grid output is identical regardless of a value, which clearly differs from the exact same grid run under value iteration. Since the nature of both algorithms promote that fact that the converged decision has a guarantee to be the optimal one, it is extremely interesting that the outputs differ heavily in such a small space. It might be that policy fails to see that states far away from the reward are relatively worse to be in with a higher amount of noise.

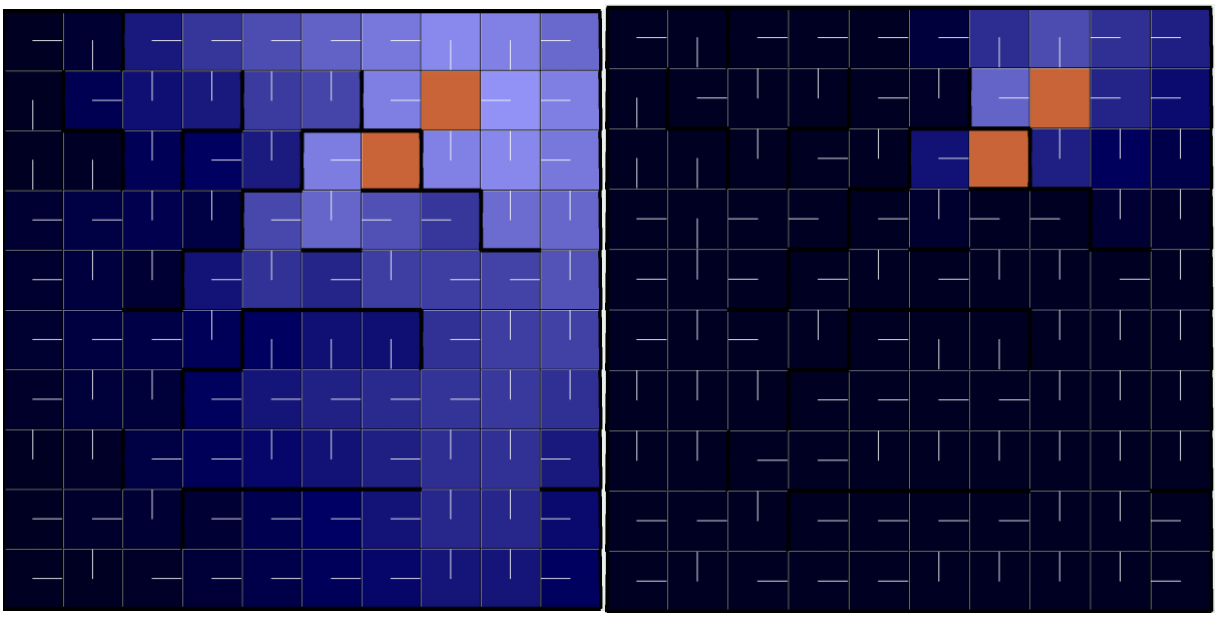
The algorithms are mechanically different in that policy iteration is choosing the correct value based on a specific stochastic policy, not a transition probability. Because of this, the algorithm will yet create a policy that moves towards lower states and, in extreme cases of PJOG

(such as on the right), a policy will be falsely generated in which the agent does not actively try to move towards the goal with relative "lower" states becoming attractive.

Medium Maze- medium_complex_multigoal.maze



Again, comparably across the same maze, policy iteration converges much quicker than the latter. It is interesting to dissect the relationship between iterations and time seeing as the medium size clearly provides ambiguity to this point. It is important to note once more that this is the only maze in which the multi-goal variable is present in the grid which alters at some level how the algorithm goes about converging.

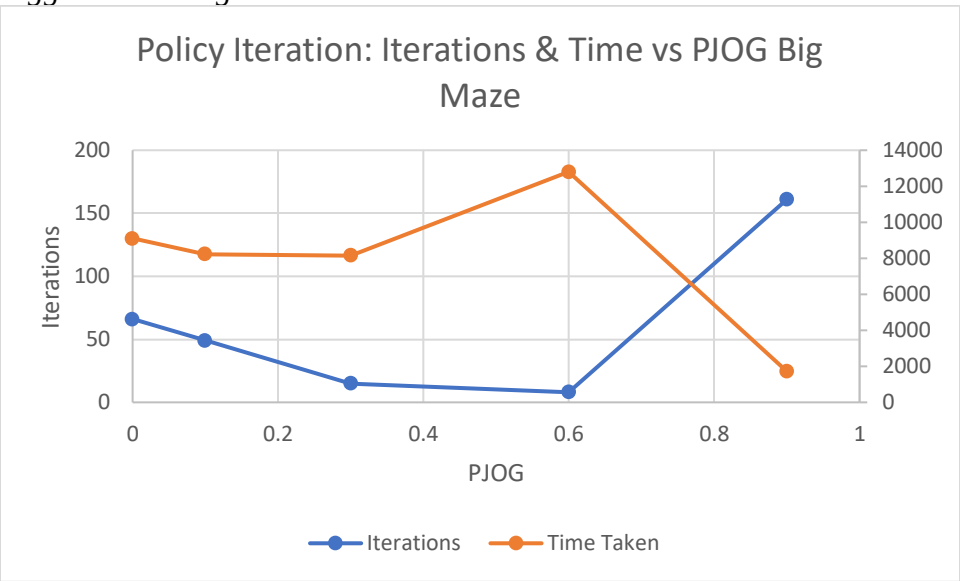


PJOG = .3 PJOG = .9

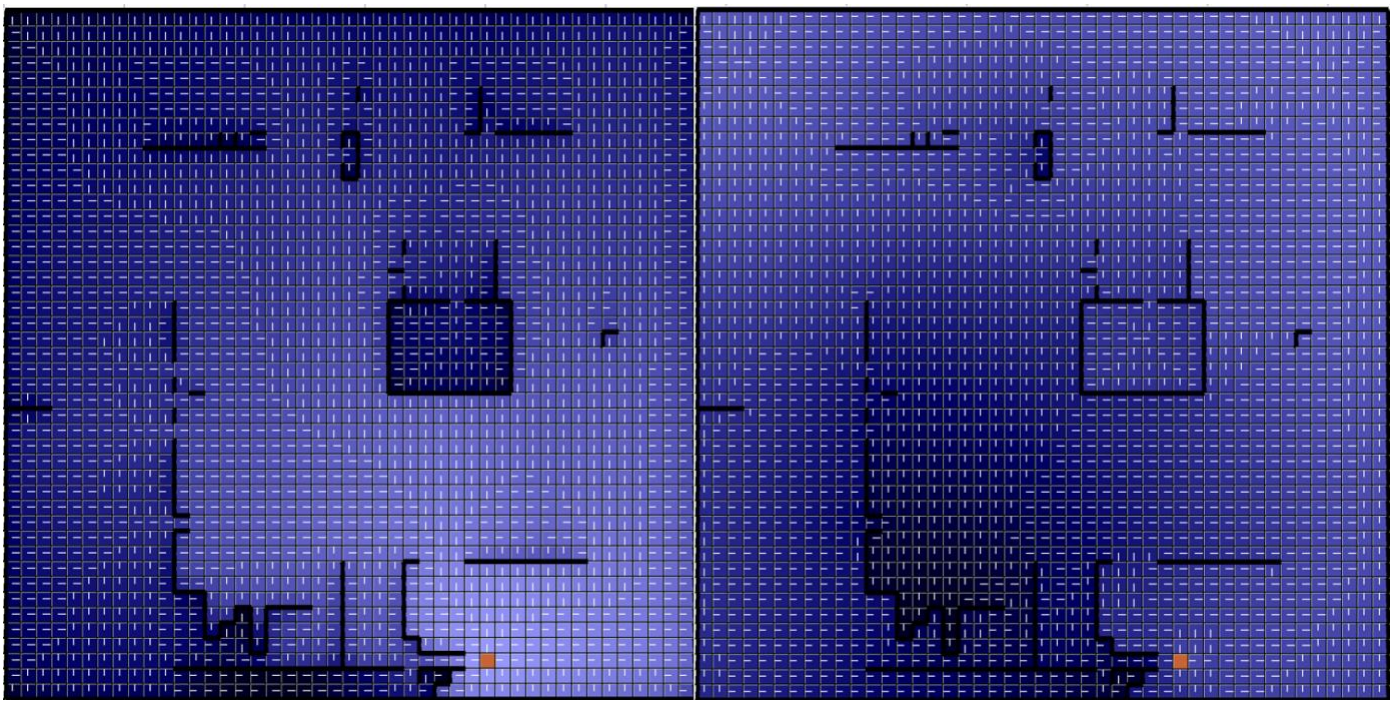
The color scheme difference is rather hyper-polarized on the Maze 2 graph than for Maze 1 seeing as nearly the entire plot is engulfed with worse plots. We see the effect more clearly of higher values centralizing near the walls as compared with what we've seen thus far. With such a high PJOG and given the above approach on the intuition of value iteration, the algorithm is

tending the optimize the grid agent to only avoid walls when a considerable distance from the reward state. Although the overall optimization would lend itself to approaching states near walls but not hitting them, the stochastic algorithm doesn't foresee this which leads to many darker states.

Biggest Maze- big.maze



This is interesting because not only do we see a quicker amount of convergence across the board, but there is a glaringly inverse relationship between how long policy iteration takes and the amount of iterations it goes through. What's most interesting, however, is the fact that we've seen thus far a negative relationship with PJOG vs. iterations, but in this case the amount spikes back up, which directly flies in the face of my initial conjecture with the opposite graph in value iteration. This is a testament to the whimsicality that such a large space can cause in terms of convergence.



PJOG = .3 PJOG = .9

This algorithm perhaps does better on the Maze 3 than the multi-goal Maze 2, which I suppose can be characterized by the multi-goal aspect of the latter seeing as the policy iteration had a harder relative time converging by thinking all the states nearly walls couldn't possibly be fruitful in reaching either of the rewards. What remains of the utmost interest is the difference across the identical Maze with how polarized the algorithms dependency on the PJOG value can become. Although we see a few mechanical blips when we increase random chance on policy iteration, there is not nearly as much of a fundamental shift in classification as in value iteration for this maze.

Conclusion:

An imperative difference to take note of in order to better understand each algorithm is the relative nature of convergence. For Maze 1, value iteration took both longer in terms of iterations and very similar time (~0) to converge, whereas in Maze 2 and 3 the policy iteration trails converged in fewer iterations with the stipulation that they general took more time to run through these iterations. We can point to the high-level intuition that the complexity within an iteration of value is much less than policy so the relationship can't be viewed as a linear one. When we start abstract with regards to create larger search spaces and grid world sizes (which are arguably the only real scenarios in which we learn something important), time complexity becomes an important attribute and we can already see how fundamentally complex individual iterations can become which could lead to a problem in efficiency.

Another interesting quality to take note of was value iteration's susceptibility to the PJOG parameter. Again, what this means intuitively is that the inverse percentage amount of time the algorithm produces the intended action, and otherwise there is an equally likely chance of it choosing the other neighboring states. Increasing noise level until the threshold delineated in the graph resulted in exponential growth in the output which we measured in this experiment. We saw, which we dove deeper into with the visualizations, a lesser degree of correlation

between PJOG and iterations, with extreme cases of no influence really at all which is a remarkably interesting difference.

Something else to note is the sort of linearly consistent effects of increasing state size on iterations. If we grab the biggest and smallest states for value iteration, we see a ~8.5 times increase in iterations while this order being around ~25 for policy (albeit on a much smaller scale because it converges faster). I believe that these algorithms both lend themselves to an O(n^2) operation which, when normalized, makes sense given the data at hand.

What's most interesting, and is also a point we've reached multiple times already, is the fact that the algorithms both inherently promise an optimal generated policy per state. This is intriguing because they visually arrive at different holistic actions across the grid. The one variable that could provide some context as to why this is happening is the PJOG chance percentage which clouds the official decision of the algorithm for any given state, more so for value iteration, but makes sense long-term. I once learned about chaos theory in a course, and this idea of arbitrarily changing reward functions or grid obstacles and objectives having such huge ramifications makes total sense in this context.