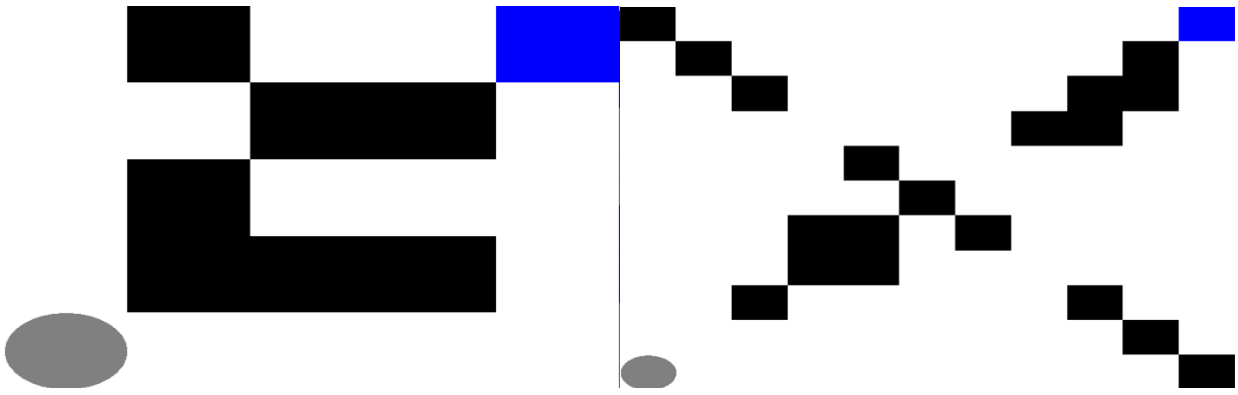
Stanley “Mason” Buchanan
CS 4641 Assignment 4
Analysis

Markov Decision Processes

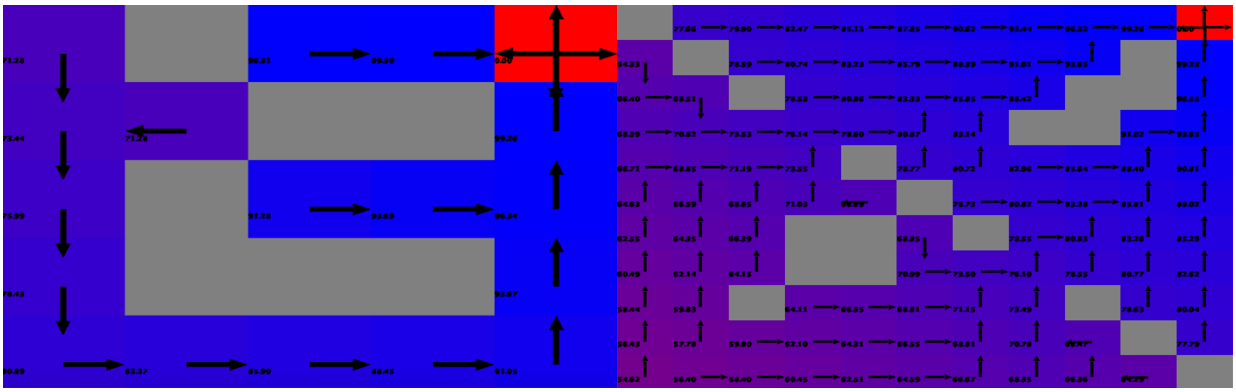
Introduction:

The Markov Decision Processes (MDP) I decided to analyze are shown in the figures below, with the one on the left being the small gridworld and the one on the right being the large gridworld. These are interesting because, though they seem simple, it does not take as much as one would initially presume to build an effective model of real-world situations. These could be anything from traffic interactions to room traversals; the use of a gridworld is truly flexible. Once I decided on these models, I analyzed the performance of Value Iteration (VI), Policy Iteration (PI), and Q-Learning (QL) on a variety of different reward structures. I decided to only vary them one part at a time, to ensure that the experiments maintained internal validity so I could make justifiable conclusions regarding the performance of the various algorithms. I had three different conditions I tested: a goal state reward of 100 and a constant per-state reward of -1, a goal state of 10,000 and a constant per-state reward of -1, and a goal state of -2 and a constant per-state reward of -1. I did have a fourth condition with a goal state of -2 and a constant per-state reward of 1, but, for fairly obvious reasons, the algorithms would not approach the goal state, thus excluding it from any viable analysis. Also, for Q-Learning I utilized epsilon-greedy as my exploration strategy so I could spend more time focusing on how reward structures affected algorithm performance.



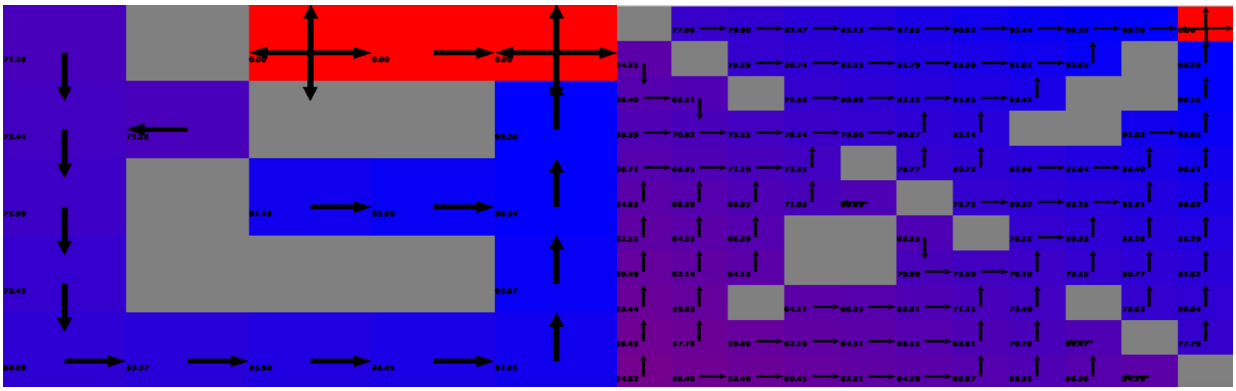
Reward [100, -1]:

Value Iteration:



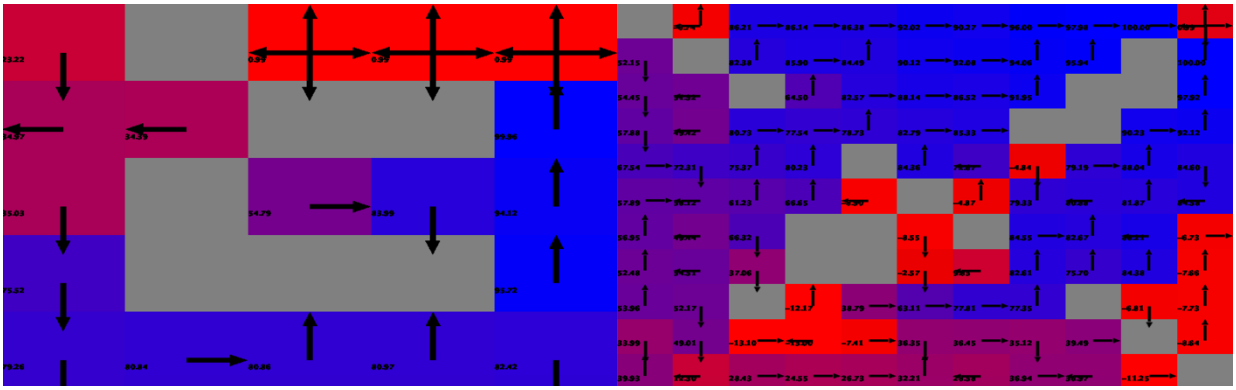
VI did a reasonable job of assigning utility values to the states, showing a distinct increase in utility as approaching the goal state. Interestingly, it assigns positive utility values to the two states to the left of the goal state found in the small gridworld. Realistically, this state would never be reached by the agent, so this utility assignment could be viewed as “wrong.”

Policy Iteration:



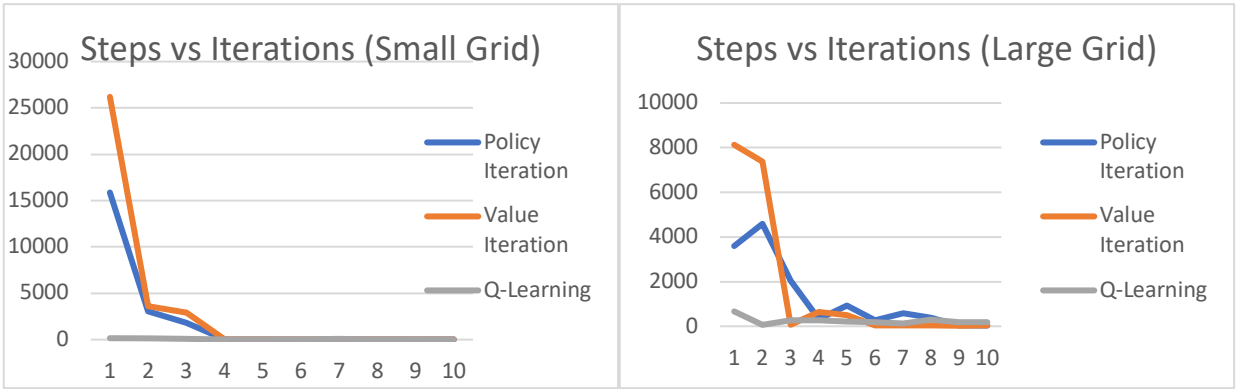
PI responded very similar to VI in that there is a very smooth utility gradient that increases as the agent approaches the goal state. Unlike VI, PI gave negative values to the two states to the left of the goal state in the small gridworld, since the agent will never reach those states.

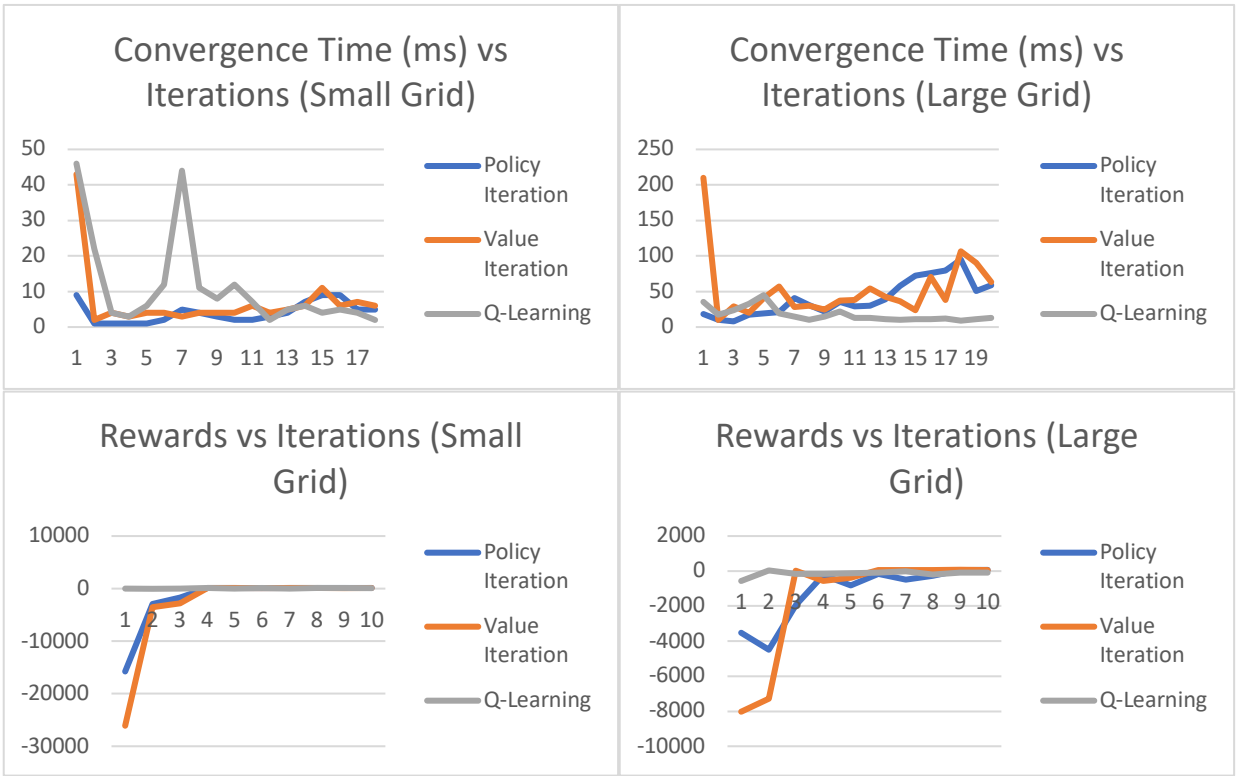
Q-Learning:



QL responded with much more variance than PI and VI, which is expected since QL is a model-free learner. However, it did more "intelligently" map out the value of moving to the two space to the left of the goal state of the smaller gridworld, which is never actually reached by the agent. It also better represented the upper-left corner of the small gridworld, which should be avoided by the agent if they are trying to maximize their reward. It is interesting to me that QL developed such a clear preference towards the upper-left diagonal of the large gridworld. It very strongly preferred the path that was only slightly shorter, whereas PI and VI were both relatively indifferent. Otherwise, the policy found by QL is all over the place, indicating that it is not a good fit for this reward structure or that the lack of a model made it more difficult to define a better policy.

Results:

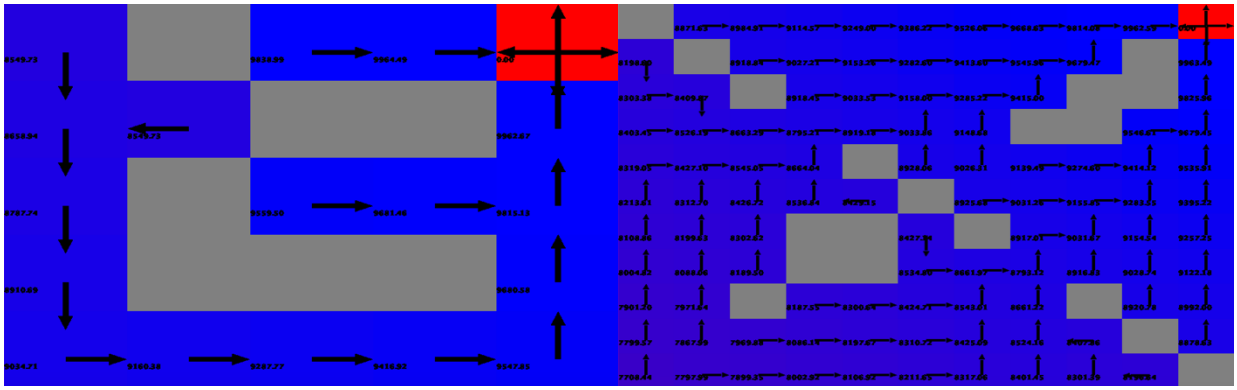




With respect to the number of steps taken per iteration, QL greatly outperformed both VI and PI in both the large and small gridworlds. Alongside this, QL performs better than both VI and PI, barring the random jump in convergence time for the small gridworld, for both convergence time and reward accumulation for both gridworlds. This is primarily interesting because of the model-free nature of QL; perhaps VI and PI are suffering from something akin to overthinking, in that they are paying close attention to things that do not necessarily matter, costing them in the long run. PI seemed to acclimate much quicker than VI did, although they did have fairly similar results. This is likely due to the higher computational cost associated with VI, which would lead to longer latency times and higher amounts of processing. The way in which VI behaved on the two states to the left of the goal state in the small gridworld goes along with this idea as well as the “overthinking” hypothesis I presented. It is interesting to me that, even though the reward at the goal state was relatively high compared to the per-state reward, the algorithms only managed to not lose any points on average and did not prioritize maximal rewards.

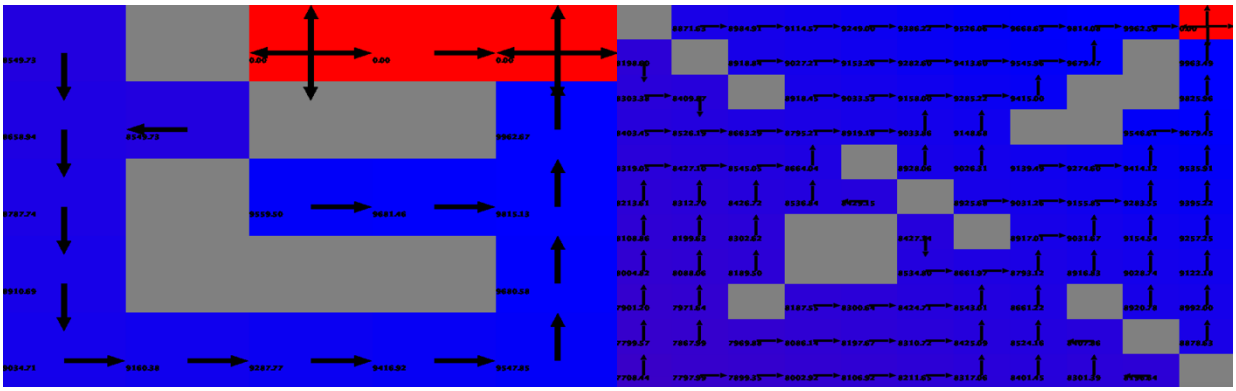
Reward [10000, -1]:

Value Iteration:



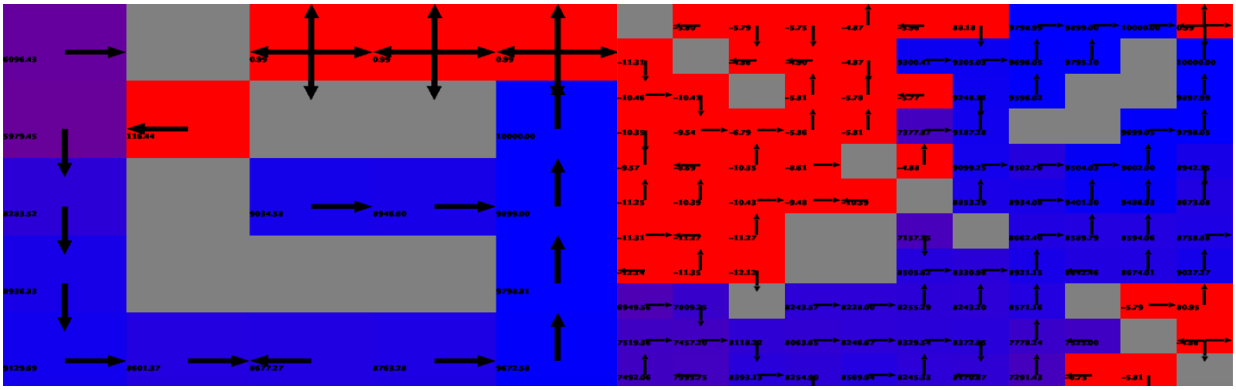
In this case, VI performed very similarly to how it did for the previous case, except, if you look closely enough, you will notice that the gradient is much less clear as you move towards the goal state. It is almost as if the algorithm realized it could be a bit more lackadaisical due to the high reward of the goal state.

Policy Iteration:



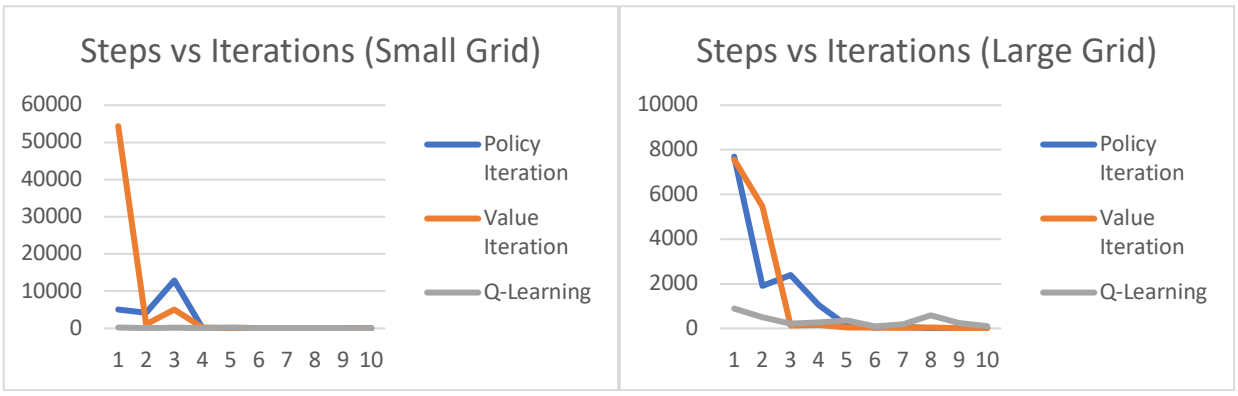
Similar to VI, PI also had a much sharper gradient in both gridworlds. This supports my idea that the algorithms were more or less acting “lazier” due to the extremely high difference between the reward of the goal state and the per-state reward. The optimal policy for PI and VI are effectively the same in this case: move towards the goal state at a comfortable pace.

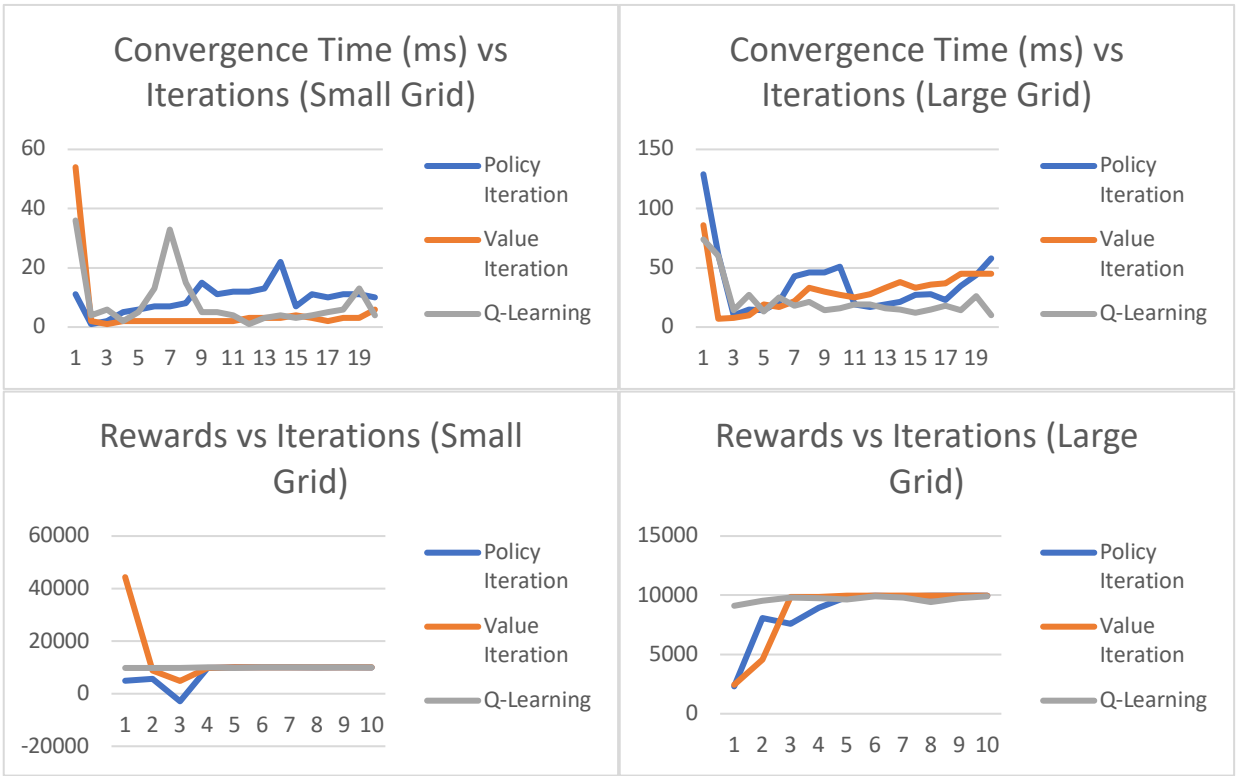
Q-Learning:



QL also exhibited the sharper gradient found in VI and PI, with the previously purple regions of the large gridworld being very red. This is particularly interesting to me, as VI and PI exhibited a sense of "laziness," QL shows more severity in its utility assignments. However, there is a noticeable difference in the preferred route for this reward structure as compared to that of the [100, -1] structure, with the lower-right diagonal being the obvious preference in this case. This could be the point at which the "laziness" appears in QL; with the allowance of wiggle room due to the high goal state reward, the learner no longer finds it necessary to travel the shortest path. It is my sneaking suspicion that due to the fact that I used epsilon-greedy as our exploration strategy that, in this case, it explored to the bottom-right to start and then very strongly assigned utility values. If exploration would have went to the upper-left instead, I believe we would see a reversal of this pattern. The policy here is much more streamlined than the previous reward structure, indicating that the previous reward structure did not provide enough leeway when QL was attempting to explore the environment, thus preventing it from developing a more optimal policy.

Results:

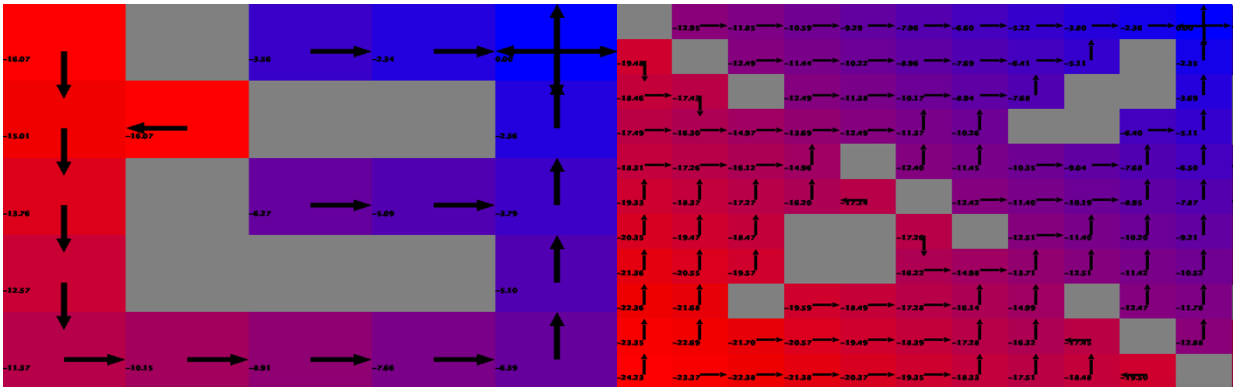




Again, QL performed very well with this reward structure, outperforming both VI and PI in every metric. It is interesting to note that the blip where QL spikes in terms of convergence time is at 7 iterations; this is also the case for [100, -1] reward structure as well, so it would be interesting to come up with some explanation, which I will attempt to by the end of this analysis. It is also intriguing that the algorithms did a better job of maximizing final reward, so perhaps it is necessary to have a large difference between the goal state reward and the constant per-state reward to cause the algorithms to mitigate the damages caused by the constant drain on the goal state reward. QL also did a better job of dealing with the unreachable states, but this is less obvious than in the previous example due to the lenience provided by having such a large goal state reward.

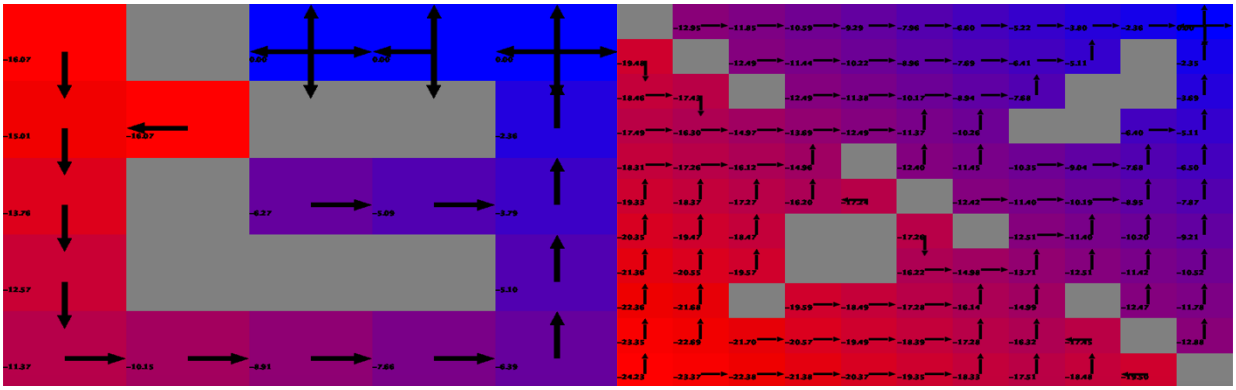
Reward [-2, -1]:

Value Iteration:



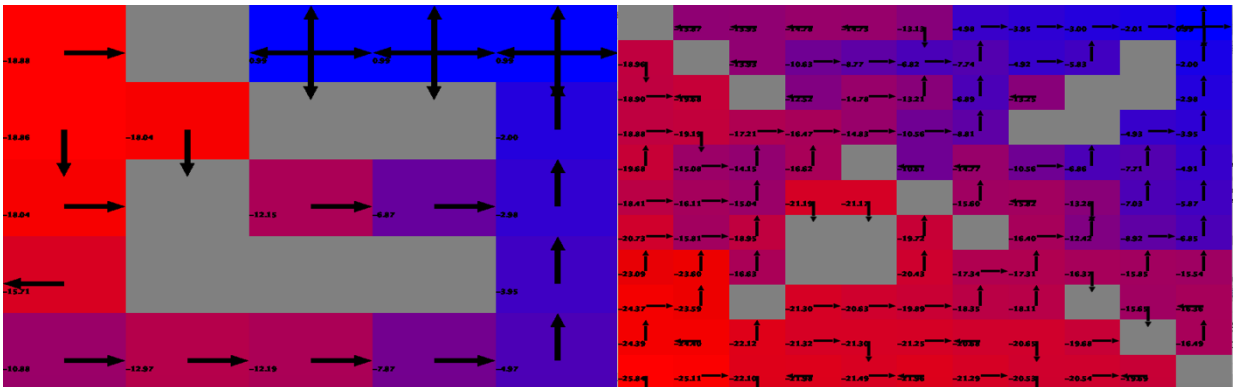
In this new reward structure, there is a definite shift from the previous two. The agent has no desire to hang around the starting area and pushes to move towards the terminal state. This is similar to the [100, -1] reward structure, but the gradient is much more pronounced.

Policy Iteration:



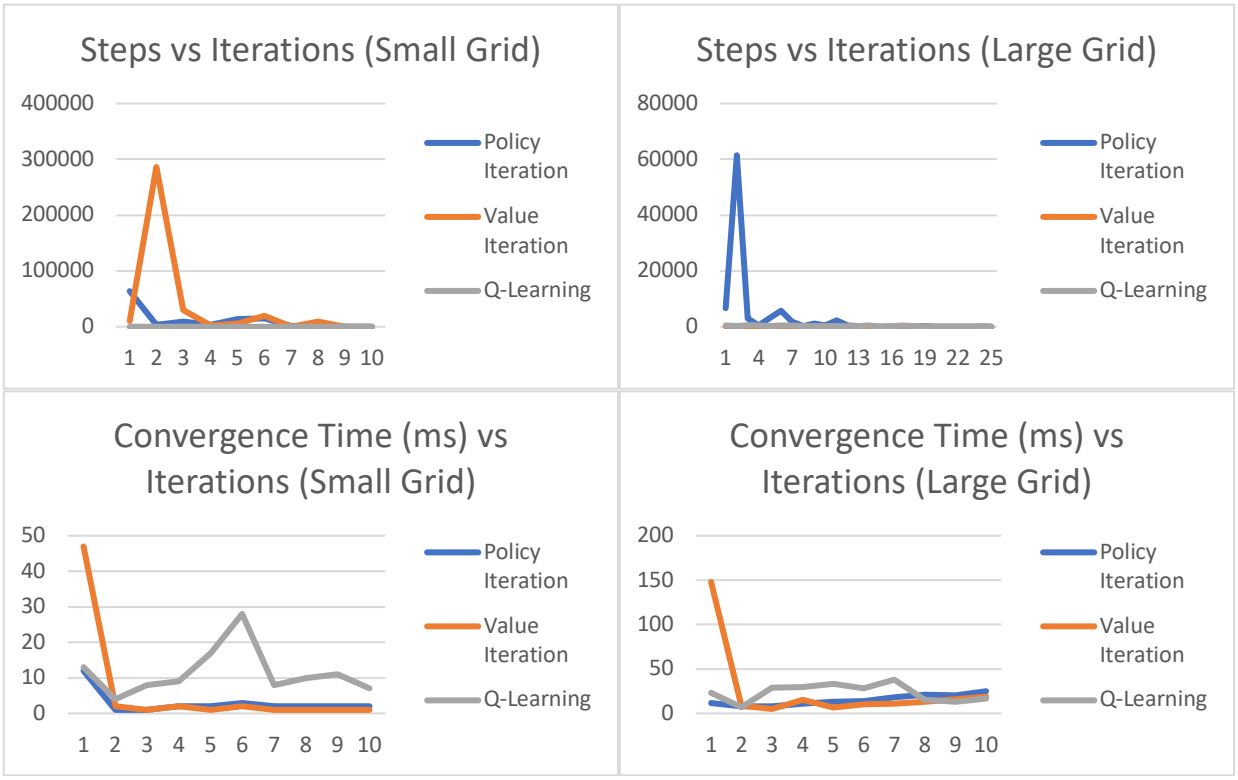
Similar to VI, PI had a much more pronounced gradient, indicating a stronger motive to move towards the goal state. The optimal policy found by PI and VI is effectively the same: move towards the goal immediately.

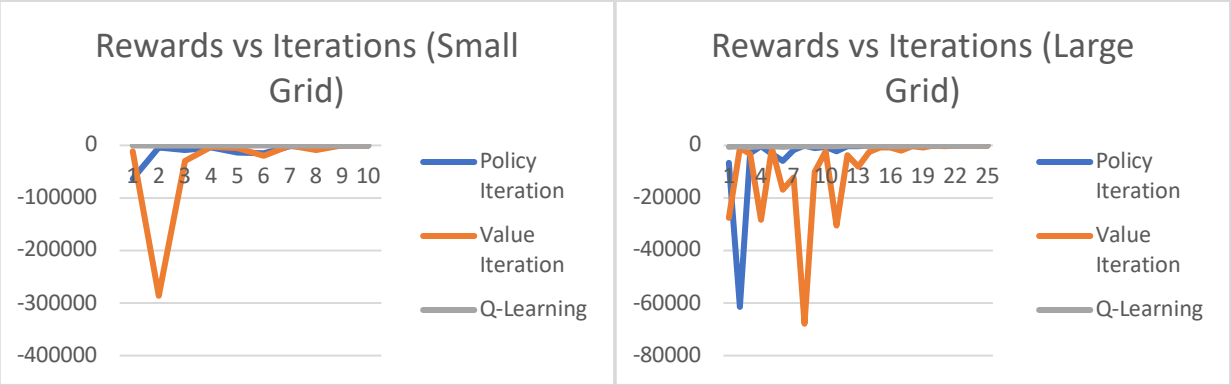
Q-Learning:



QL, in general, developed the same strategy as PI and VI: move towards the goal quickly. The primary difference comes in the way that the policy is expressed, with some policies being questionable. Some of them have reasonable utility value assignments, but the direction in which it is recommended to move is strange, indicating that the lack of domain knowledge really hurt QL in this instance.

Results:





Again, for these gridworlds, QL tended to outperform both VI and PI. The only case in which it did not was with convergence time on the small gridworld, which is likely a reflection of the strange policy decisions that were found in the upper-left portion of the grid. Even though that is the case, QL did much better with regards to the final rewards gained when compared to VI and PI. It is important to note that, even though it looks as if QL maintained a reward of effectively zero, we know this cannot be the case due to the constantly negative rewards that are to be found in a [-2, -1] reward structure. The fact that it looks that way only goes to show how poorly VI and PI did with this reward structure.

Conclusion:

In conclusion, QL was shown to be much more effective at solving these two MDPs regardless of reward structure. There were some odd cases, but I think that for the level of complexity presented by these two MDPs that QL is the optimal choice in terms of steps taken, convergence time, and total rewards. One key takeaway here is that QL did stumble a bit when it came to needless exploration. Because of the epsilon-greedy approach, there is potential for the agent to move back-and-forth throughout the gridworld needlessly, wasting time and rewards. Perhaps in gridworlds with more “rooms,” VI and PI would begin to overcome QL due to the way that they are able to look into the details of the model more than QL can. It is interesting that QL was relatively unaffected by the number of states. This was likely due to the fact that QL is model-free, so it cannot have the same “anticipatory anxiety” felt by VI and PI with their burden of knowing. PI and VI were very affected by the number of states found in each gridworld due to the way that they constantly build policies over the model they contain. This increases computational complexity, which then increases convergence time. While QL was the clear winner with these two MDPs, VI and PI are likely to shine in environments with more trickery or complex decision making involved, as their internal models will become a greater benefit to them.