1 - Early Stopping

Run this code with: python prob1.py

The following table describes the results for this problem:

Name	**Past Window**	**Rejection Rates**	**Optimal Stopping**
Baseline	1	N/A	36%
Equal Rejection	5	50%, 50%, 50%, 50%, 50%,	43%
Increasing Rejection	5	20%, 35%, 50%, 65%, 85%	28%

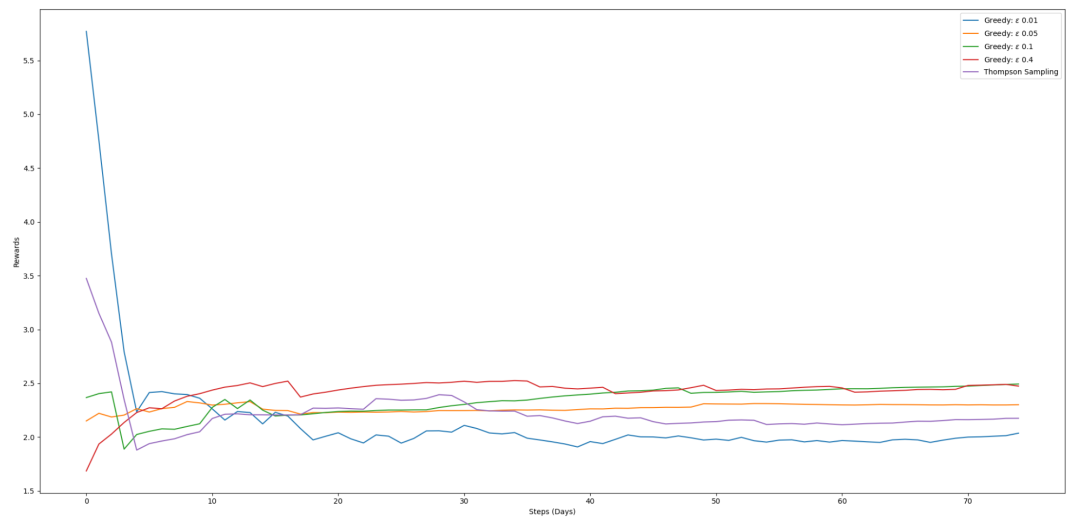
Having equal rejection across the board increases the percentage of the domain you can look through while still obtaining the optional result. I believe this to be because while you are not guaranteed to successfully hire the optimal person, you can do so with more information than you would have before. In the optimal scenario, you would move four people past the optimal, then make the choice, giving you more data.

When the rejection rates increase for each successive time stamp, the optimal stopping percentage actually decreases. Even though you can obtain more information, the high rejection rates end up overwhelming the information gained and resulting in a lower optimal stopping percentage.

2 - Explore and Exploit

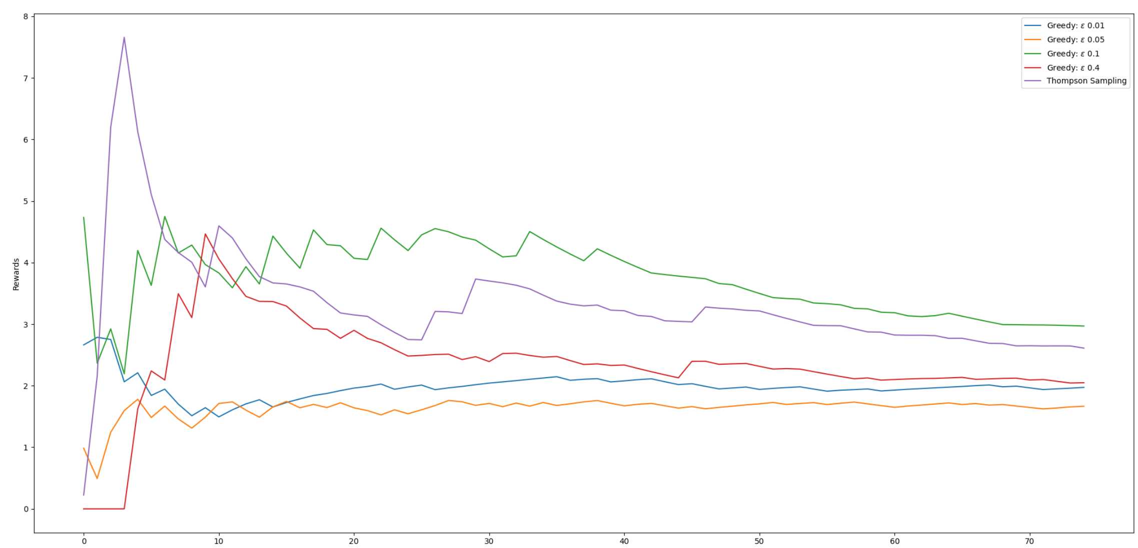
Run this code with: `python prob2.py`

For this problem I applied the same algorithms we used on homework two. Namely, epsilon-greedy and Thompson sampling. For the epsilon greedy approach I used the following values for epsilon: 0.01, 0.05, 0.1, 0.4. The following graph show all of the approaches as their rewards converged over a simulated 75 days:



As the graph makes apparent, They all end up converging near the same rewards range, with some of them converging there rather quickly. I would recommend making a decision after **25** days. Any fewer days than that and there wasn't enough data and they would often fluctuate on the chosen optimal location between rounds. For my final decision I opted to go with the Thompson sampling choice as it seemed to have the most stable behavior from round to round. The Thompson sampling chose **area 1.** The epsilon-greedy approaches usually followed suit, but it was common that at least one of them would choose area 3 instead.

When including the baseline sensor, the different approaches have a lot more difficulty converging:



Over the same timeframe of 25 days, Thompson struggled to pick a consistent area, often opting for 2 or 3. So I would recommend increasing the number of days before the choice is made to **50.** With this change, Thompson still settles on **areas 1.]**

3 - American Options

Run this code with `python prob3.py`

Below are averages obtained for each of the scenarios.

Name	European Value	American Value
Without Volatility Surge	$188,000	$150,000
With Volatility Surge	$189,000	$125,000

The surge in volatility had very little effect on the overall value when using a European call. I believe this is because over the entire time span the results tend to be the same with or without the surge in volatility. Because of this, the European call won't be affected much by the surge. In contrast, the American call had a significant decrease in value when applying the surge in volatility. This is likely due to the fact that including the surge means we might hit our strike price earlier on, but just barely. If that occurs, we immediately call instead of waiting to see if it continues to increase for the entire time-span like we would with the European call.

4 - Bonus

1. Better preparation for in-class coding
2. Clearer goals / expectation on the assignments
3. Clearer timeline on what topics will be covered in class and more transparency when those plans change. I don't think the schedule outline in the course has been updated to reflect changes made to curriculum