CS 747: Programming Assignment 1

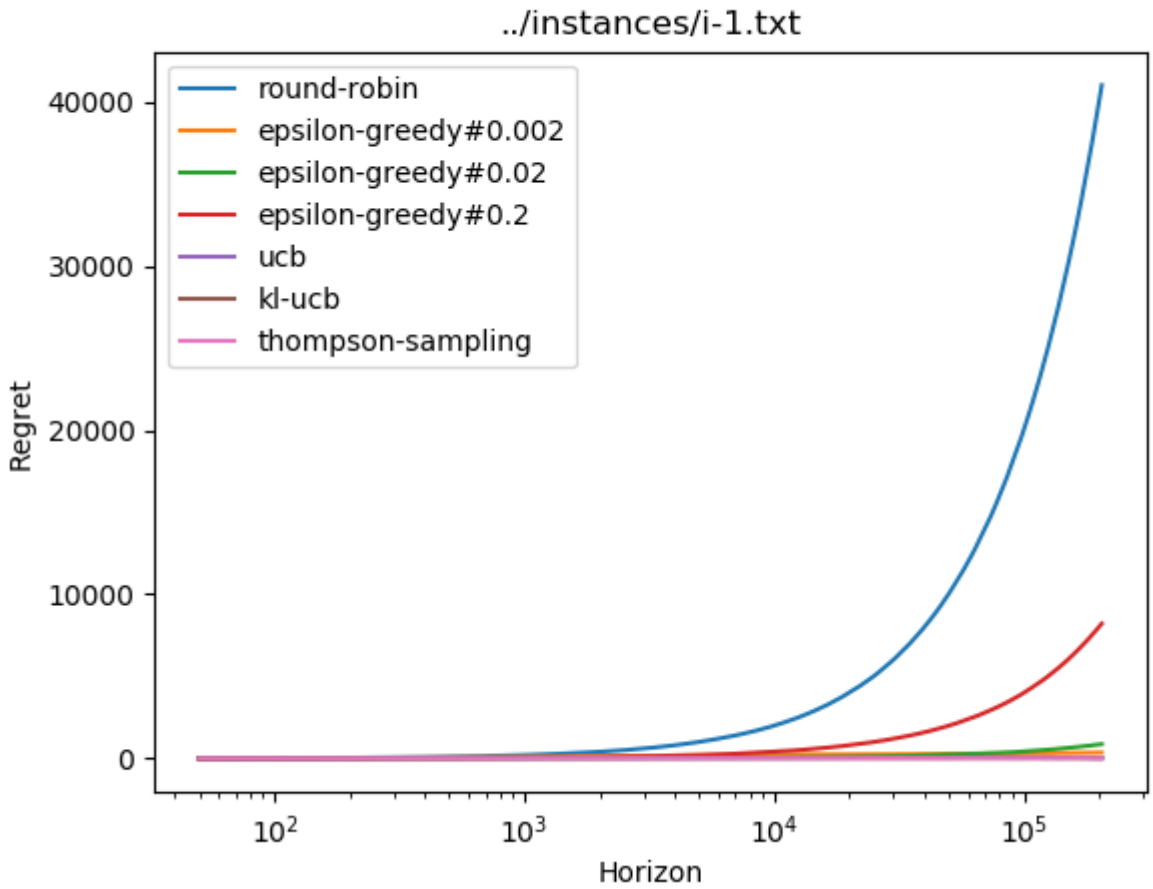
Name: Vamsi Krishna Reddy Satti **Roll Number:** 160050064

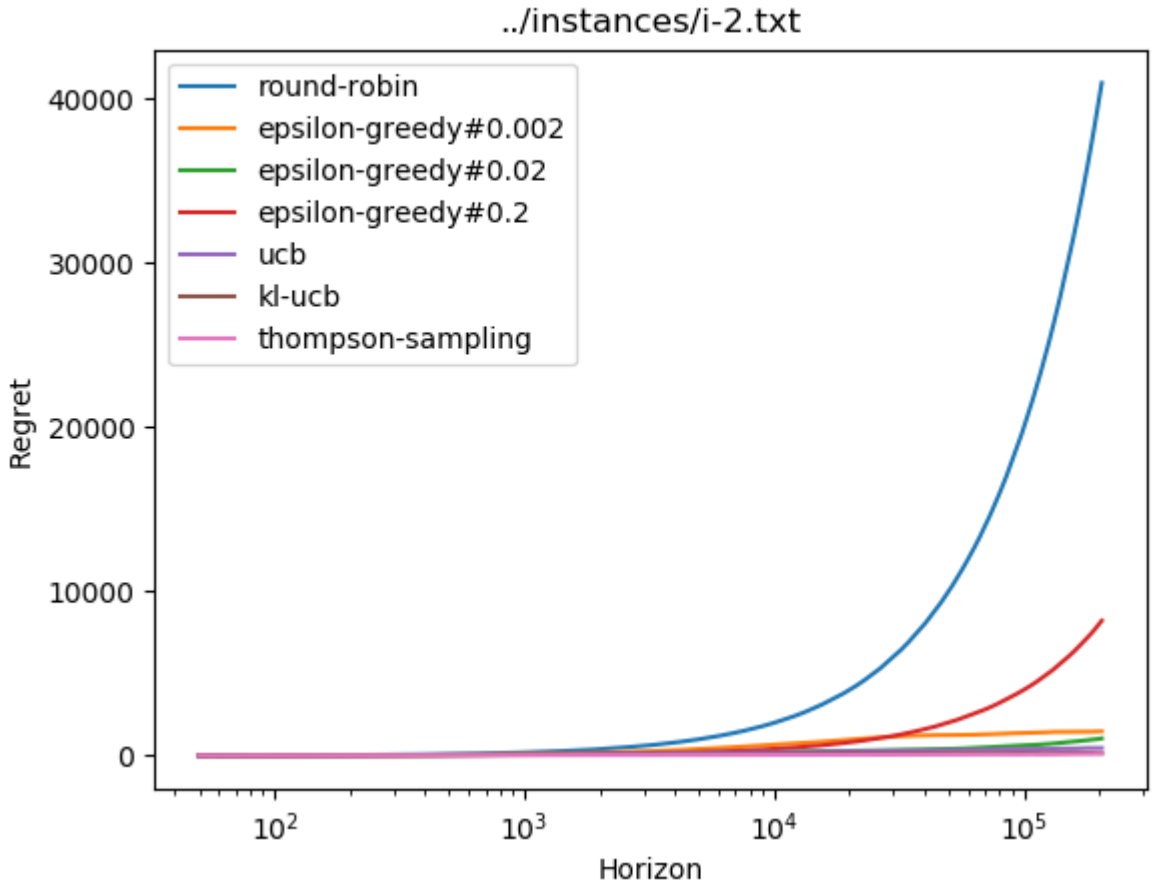
Assumption

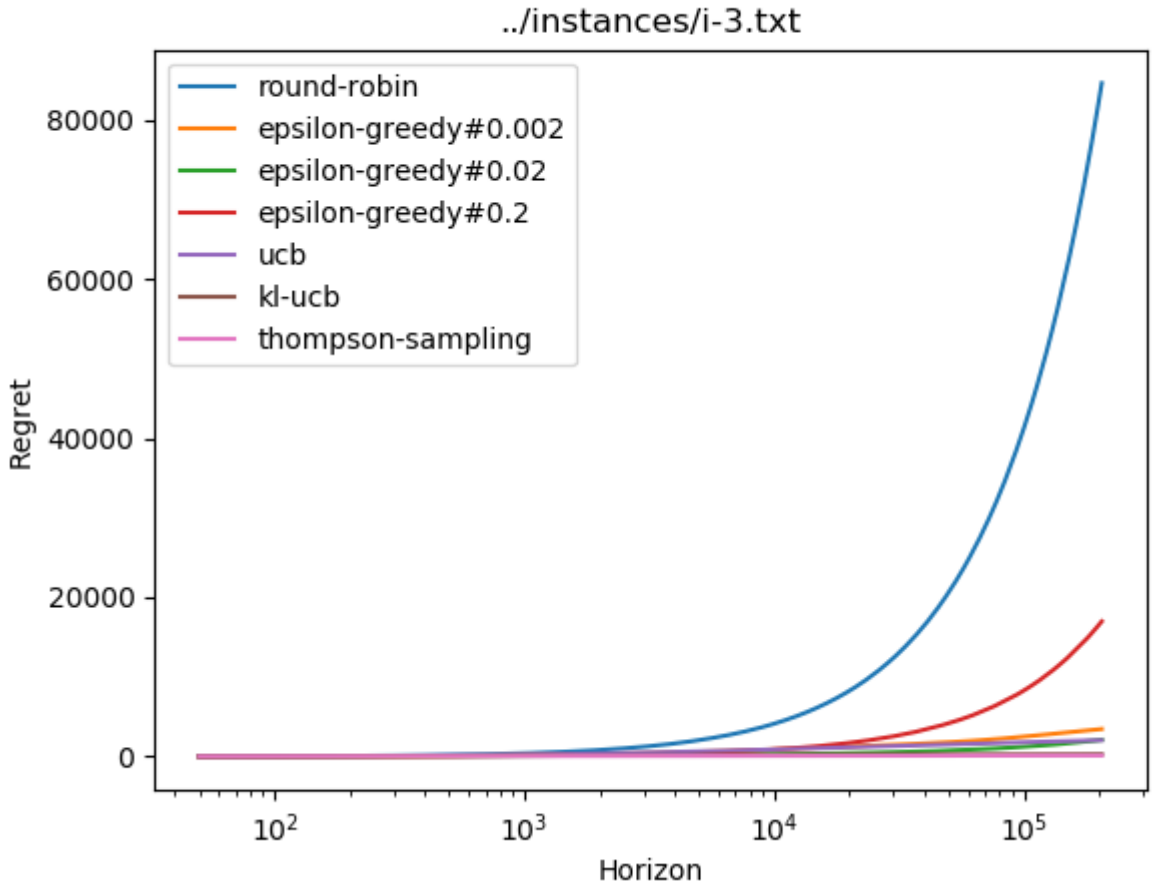
- All ties are broken randomly with uniform sampling.
- First *number_of_arms* pulls in UCB and KL-UCB algorithms are done once for each arm initially.
- Binary Search was used to find optimal *q_max* for each arm in KL-UCB algorithm with precision 10^{-4}.

Plots

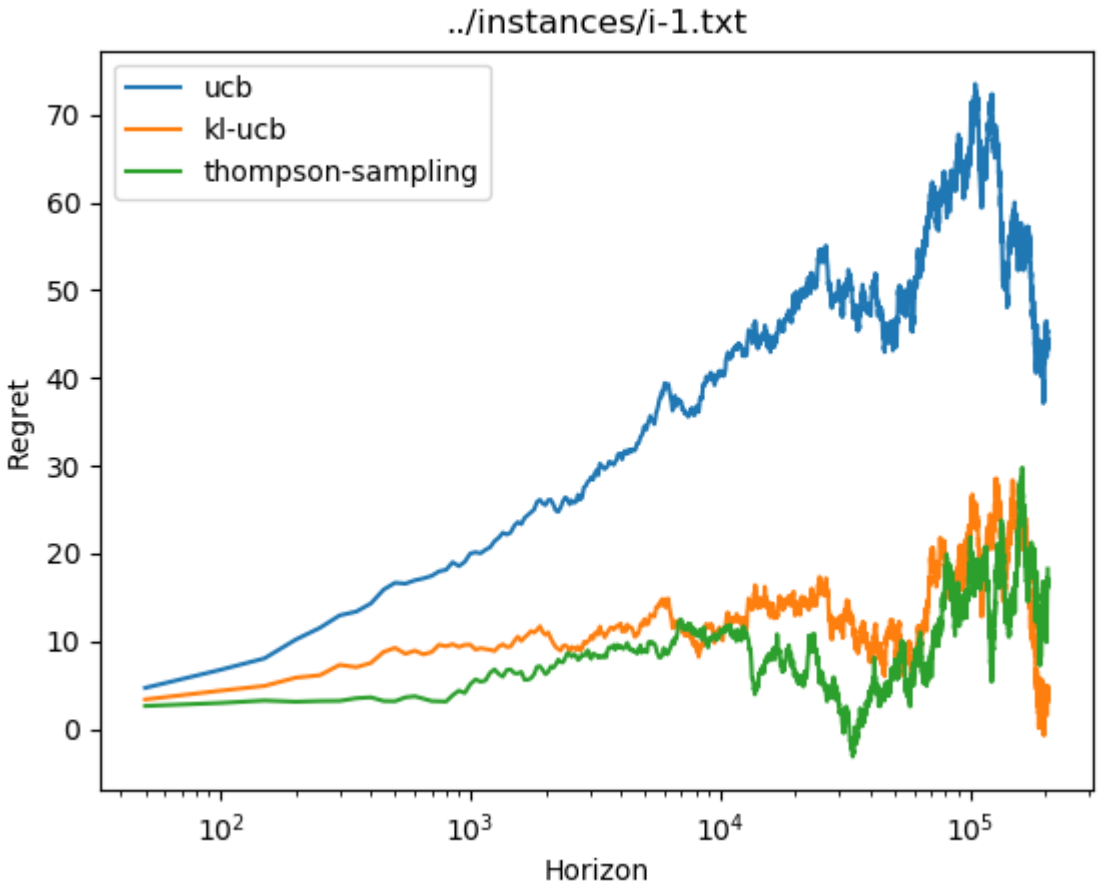
All the plots for three runs on different instances are as shown below.




../instances/i-2.txt
round-robin
epsilon-greedy#0.002
epsilon-greedy#0.02
epsilon-greedy#0.2
ucb
kl-ucb
thompson-sampling
Regret
Horizon
0
10000
20000
30000
40000
10^2
10^3
10^4
10^5

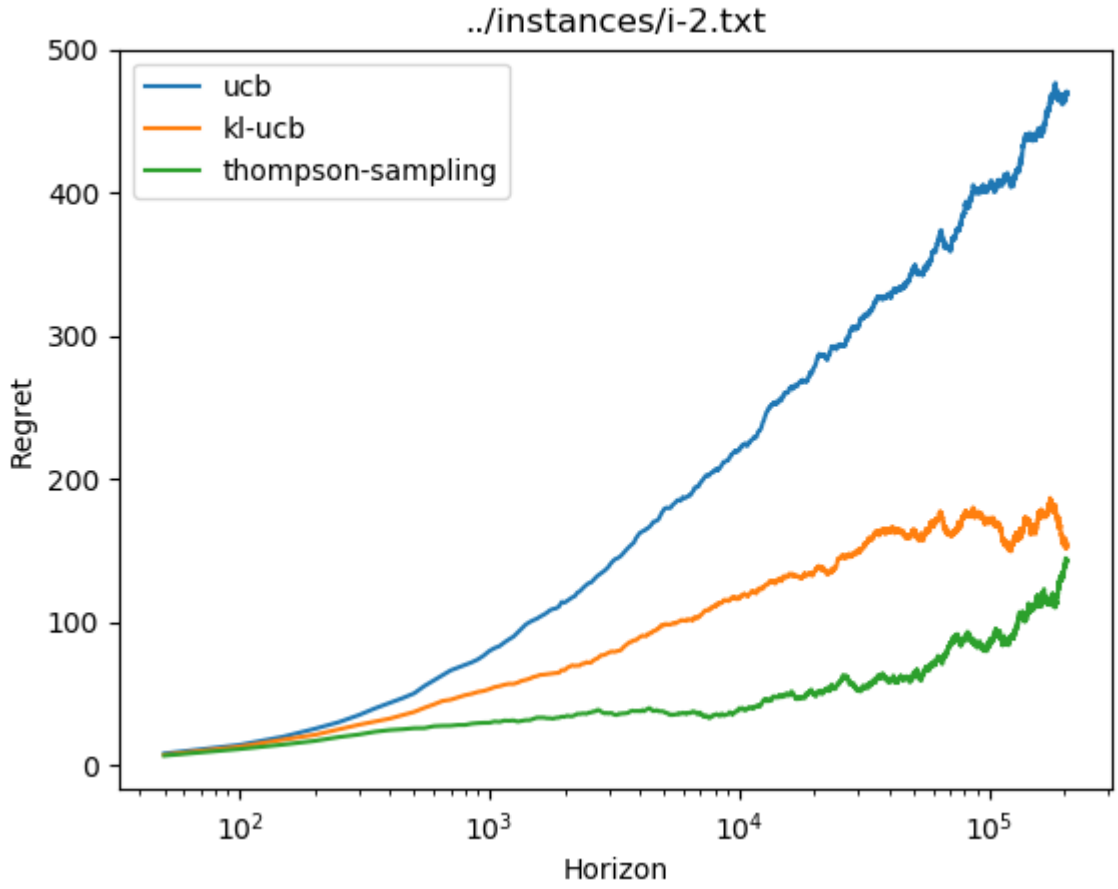

../instances/i-3.txt
round-robin
epsilon-greedy#0.002
epsilon-greedy#0.02
epsilon-greedy#0.2
ucb
kl-ucb
thompson-sampling
Regret
Horizon
0
20000
40000
60000
80000
10^2
10^3
10^4
10^5

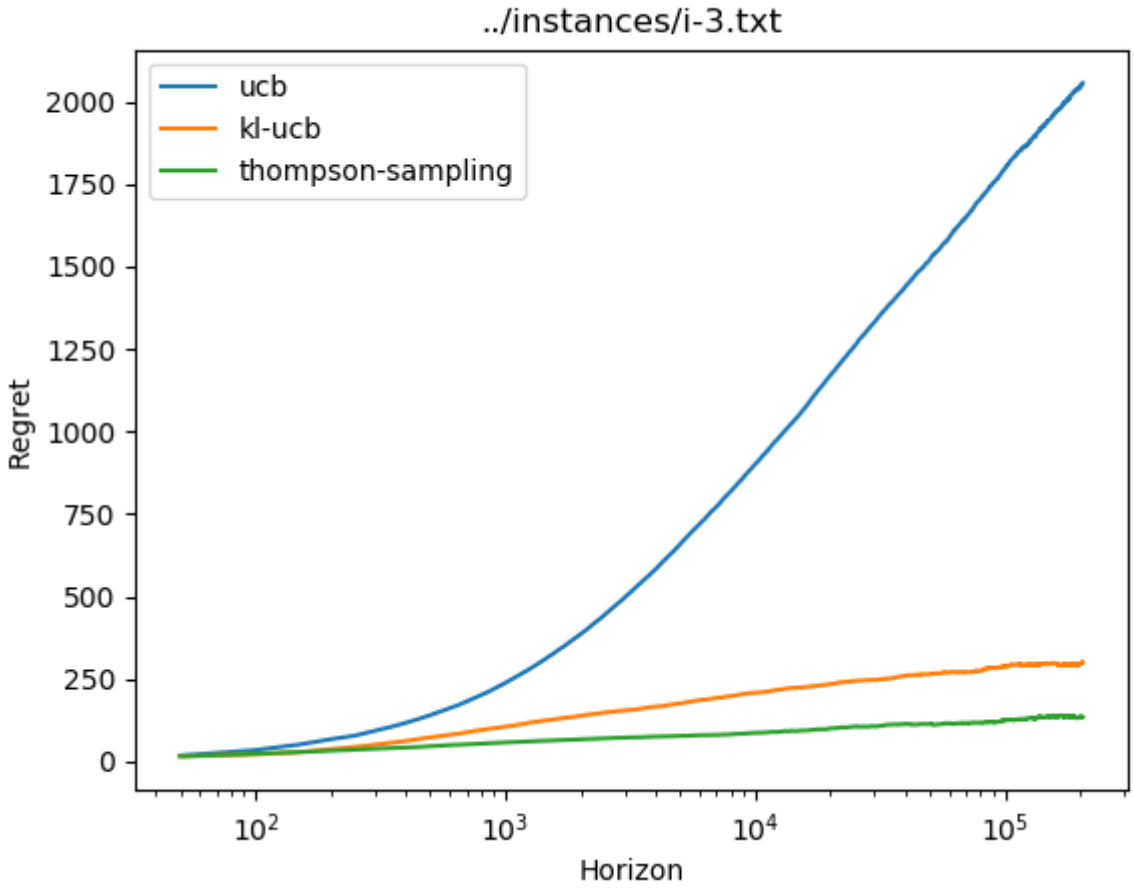
Since the plots for UCB, KL-UCB and Thompson Sampling were congested in the above plots, the follow are the plots containing only these three algorithms. Please be aware that the colors of the plot lines (algorithms) are not consistent with above plots.



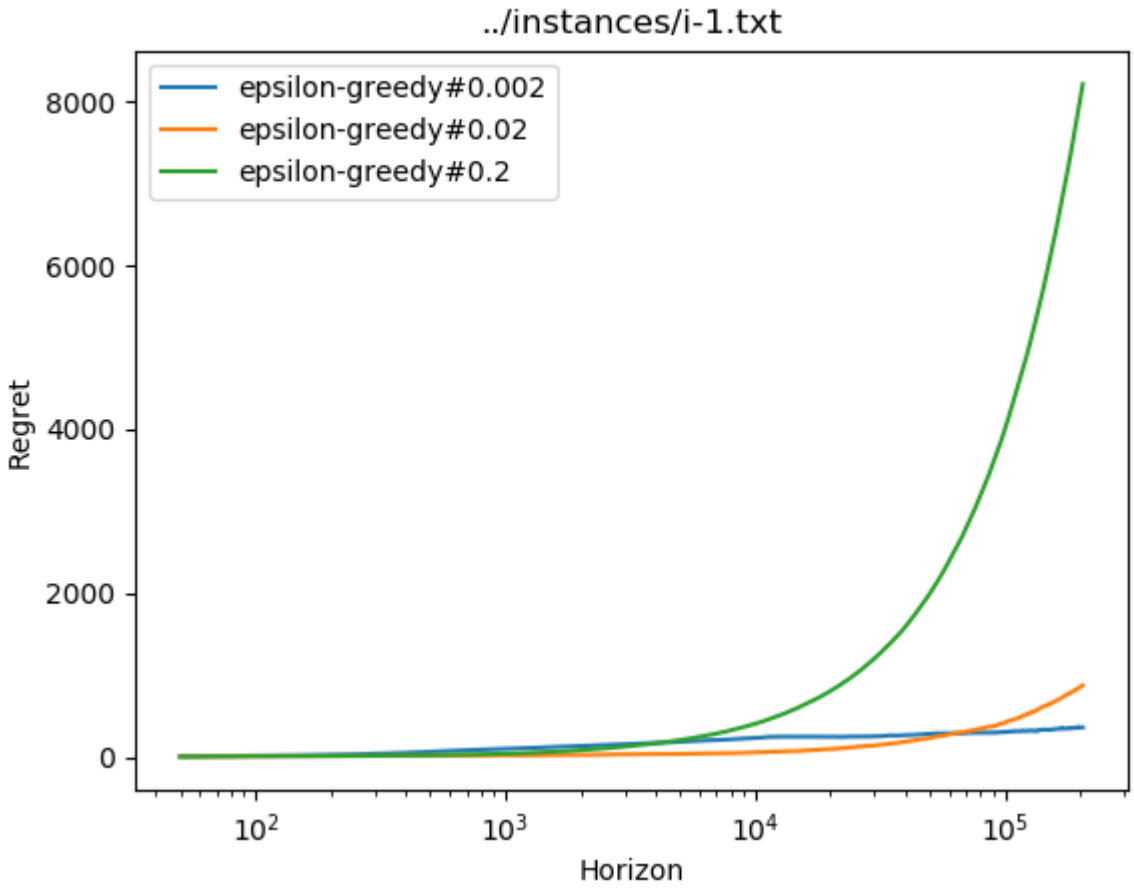
[This plot is not very stable probably because 50 seeds were not high enough for a good average. (This is likely due to larger gaps in between probabilities in arms)]

Continued on next page ...

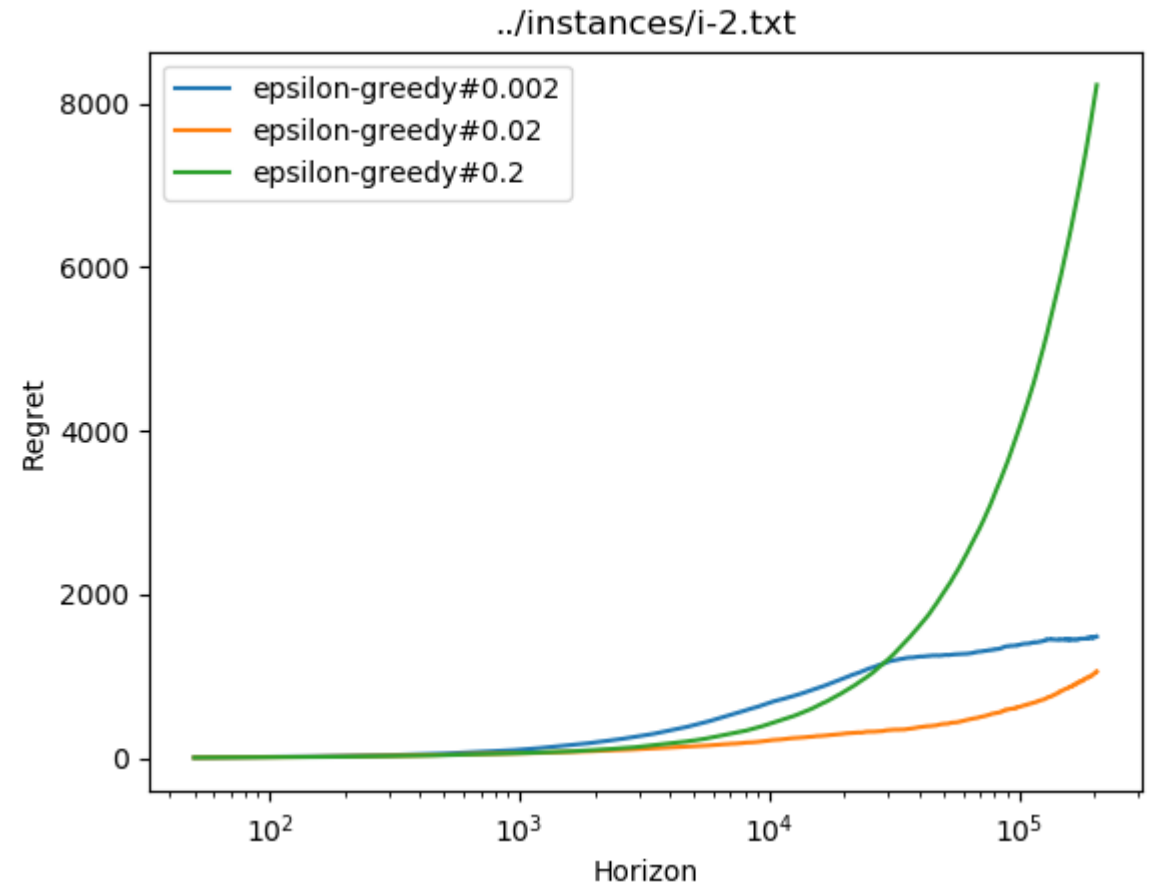

../instances/i-2.txt
ucb
kl-ucb
thompson-sampling
Regret
Horizon
0
100
200
300
400
500
10^2
10^3
10^4
10^5

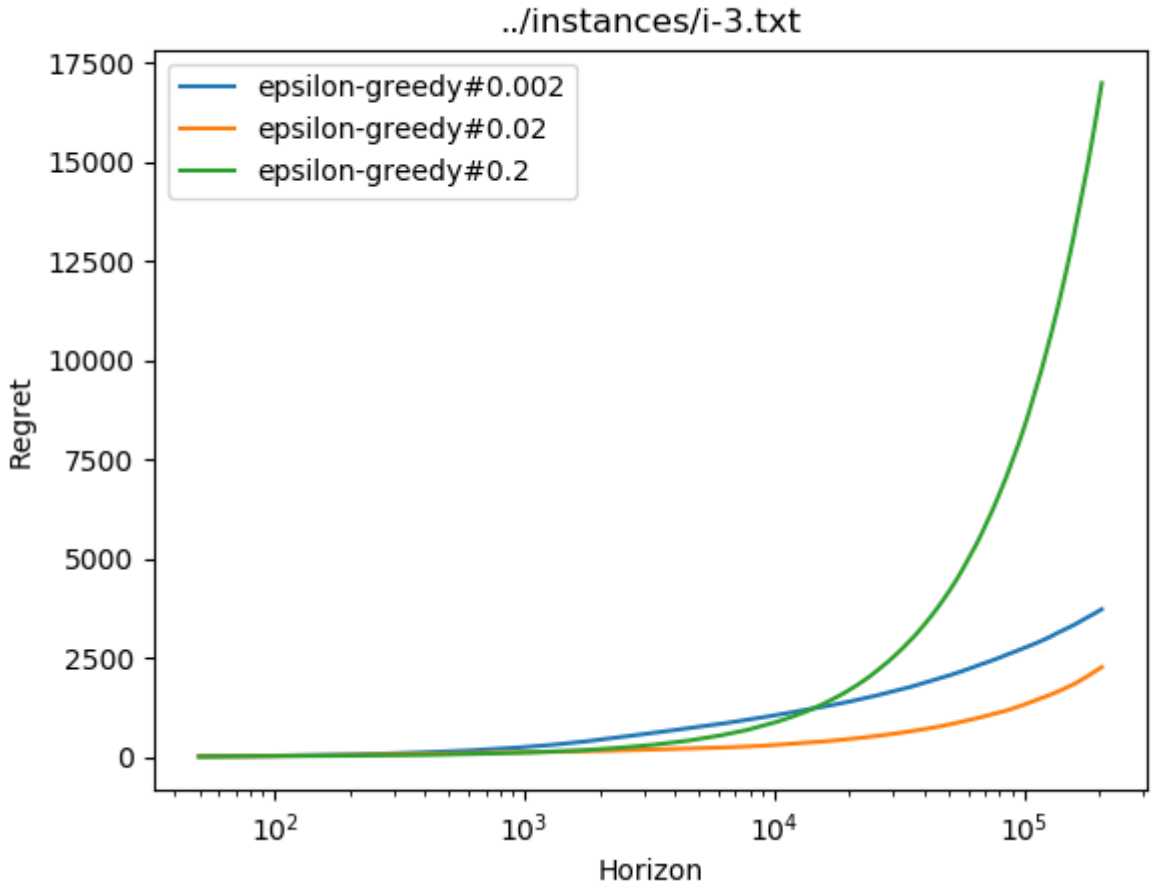

../instances/i-3.txt
ucb
kl-ucb
thompson-sampling
Regret
Horizon
0
250
500
750
1000
1250
1500
1750
2000
10^2
10^3
10^4
10^5

Similarly, the plots for Epsilon-Greedy with epsilon 0.002, 0.02, 0.2 are exclusively plotted below. Please be aware that the colors of the plot lines (algorithms) are not consistent with above plots.



Continued on next page ...


../instances/i-2.txt
epsilon-greedy#0.002
epsilon-greedy#0.02
epsilon-greedy#0.2
Regret
Horizon
0
2000
4000
6000
8000
10^2
10^3
10^4
10^5


../instances/i-3.txt
epsilon-greedy#0.002
epsilon-greedy#0.02
epsilon-greedy#0.2
Regret
Horizon
0
2500
5000
7500
10000
12500
15000
17500
10^2
10^3
10^4
10^5

Results & Observations

- Firstly, we observe how well each algorithm performs relatively for each instance: (lower the regret better the performance; ordering below is based on regret values)
    - *Instance 1*: round-robin > epsilon-greedy#0.2 > epsilon-greedy#0.02 > epsilon-greedy#0.002 > ucb > kl-ucb > thompson-sampling
    - *Instance 2*: round-robin > epsilon-greedy#0.2 > epsilon-greedy#0.002 > epsilon-greedy#0.02 > ucb > kl-ucb > thompson-sampling
    - *Instance 3*: round-robin > epsilon-greedy#0.2 > epsilon-greedy#0.002 > ucb > epsilon-greedy#0.02 > kl-ucb > thompson-sampling

- Thompson Sampling seems to work the best amongst all the algorithms across all instances. KL-UCB too performs pretty closely well, but takes more time to decide the pull for each run.

- As expected, Round-Robin is the worst performing algorithm, in fact the cumulative regret linearly increases with horizon in this case because the algorithm itself is dumb and doesn't even try to infer from the rewards values about the arms as theoretically expected, $O(T)$ is its expected cumulative reward.

- Theoretical upper bounds are proven for UCB, KL-UCB and Thompson Sampling are in order $O(log(T))$ which holds more and more better for higher values of horizon and the above plots confirm that their performance is indeed optimal.

- In all instances, KL-UCB performs better than UCB because of lower theoretical bounds on regret (better constant coefficients in upper bounds).

- Epsilon-Greedy with epsilon 0.2 seems to work well initially but miserably gets higher regrets for higher horizons. This is due to the fact that after a few horizons, since the exploration is pretty high (with probability 0.2), it probably finds out the best arm to pull soon. But, yet it tends to explore too much and hence unnecessarily explores more for higher horizons.

- Epsilon Greedy with epsilon 0.002 stabilizes its regret after some pulls because it could have figured out approximately the best arm till then, and anyway explores less thereafter. Thus, the further contribution to regret is less. Nevertheless, during the initial pulls, the regret does increase quite well due to sub-optimal pulls. In fact, more the arms, more the pulls it needs to stabilize its regret as supported by the above plots.

- From the plots above, Epsilon Greedy with epsilon 0.02 seems to be working better than other epsilon in general (specifically in instance 2, 3). This is in line with the Exploitation-Exploration tradeoff we are aware of. This epsilon seems to work well in the tradeoff for instance 2, 3 and for lower horizons in case of instance 1.