Preliminary Experiment

January 17, 2018

Bandit Prior/Instances that were considered (only for 10 arms):

- Uniform: Draw the mean rewards for the arms from [0.25, 0.75]
- "HeavyTail": We took the mean rewards to be randomly drawn from Beta($\alpha = 0.6, \beta = 0.6$). With this distribution it was likely to have arms that were at the extremes (close to 1 and close to 0) but also some of the arms with intermediate value means.
- Needle-in-haystack
    1. Low Mean
        (a) Low - 9 arms with mean 0.01, 1 arm with mean 0.02 (+ 0.01)
        (b) Medium - 9 arms with mean 0.01, 1 arm with mean 0.06 (+ 0.05)
        (c) High - 9 arms with mean 0.01, 1 arm with mean 0.21 (+ 0.20)
    2. Medium Mean
        (a) Low - 9 arms with mean 0.50, 1 arm with mean 0.01 (+ 0.01)
        (b) Medium - 9 arms with mean 0.50, 1 arm with mean 0.55 (+ 0.05)
        (c) High - 9 arms with mean 0.50, 1 arm with mean 0.70 (+ 0.20)
    3. High Mean
        (a) Low - 9 arms with mean 0.80, 1 arm with mean 0.81 (+ 0.01)
        (b) Medium - 9 arms with mean 0.80, 1 arm with mean 0.85 (+ 0.05)
        (c) High - 9 arms with mean 0.80, 1 arm with mean 1.0 (+ 0.20)

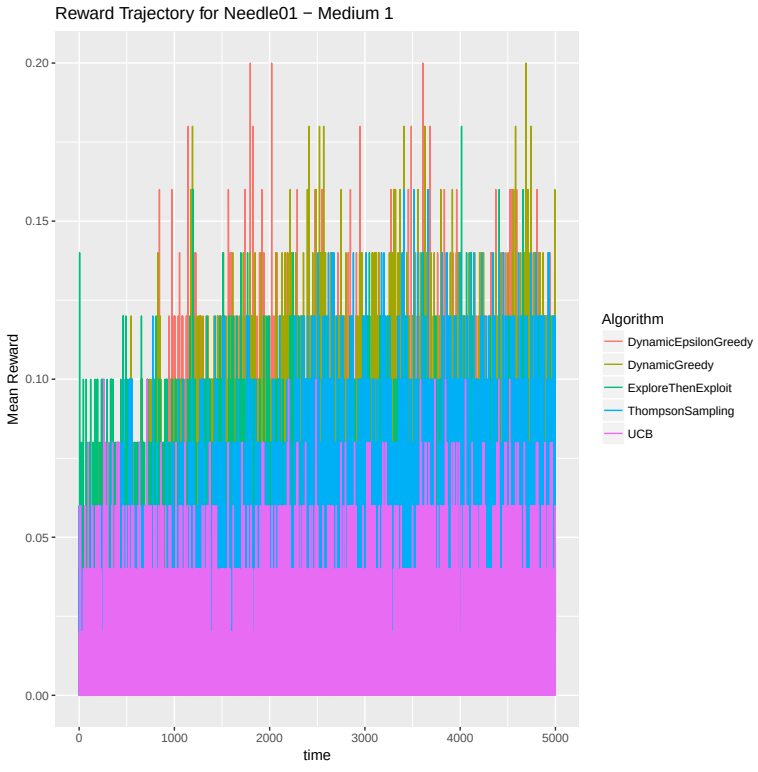
Algorithms considered:

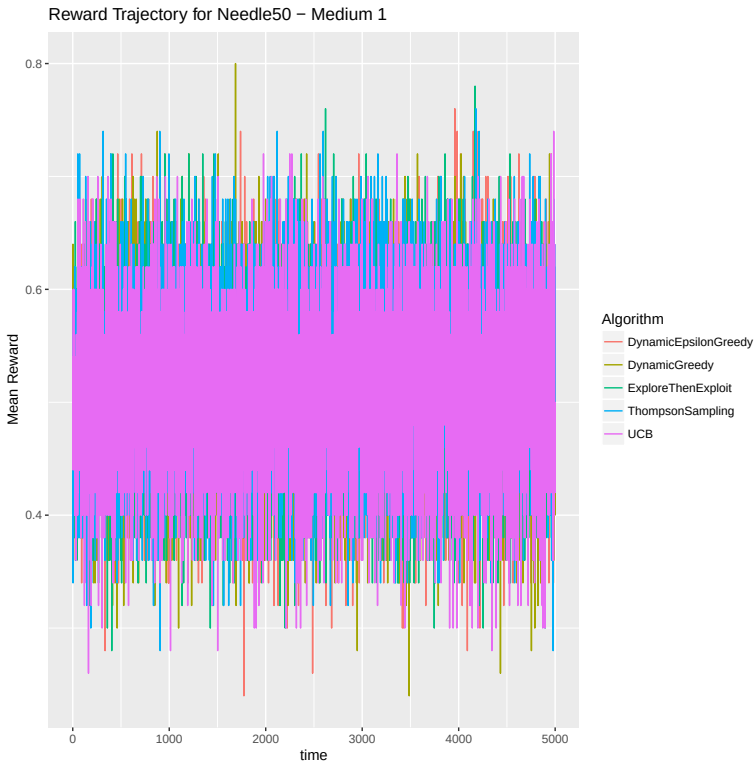
1. ThompsonSampling with priors of $Beta(5, 5)$ for every arm.
2. ExploreThenExploit with 100 exploration steps
3. DynamicGreedy with priors of $Beta(5, 5)$ for every arm
4. Dynamic ϵ-greedy with ϵ=0.05
5. UCB1

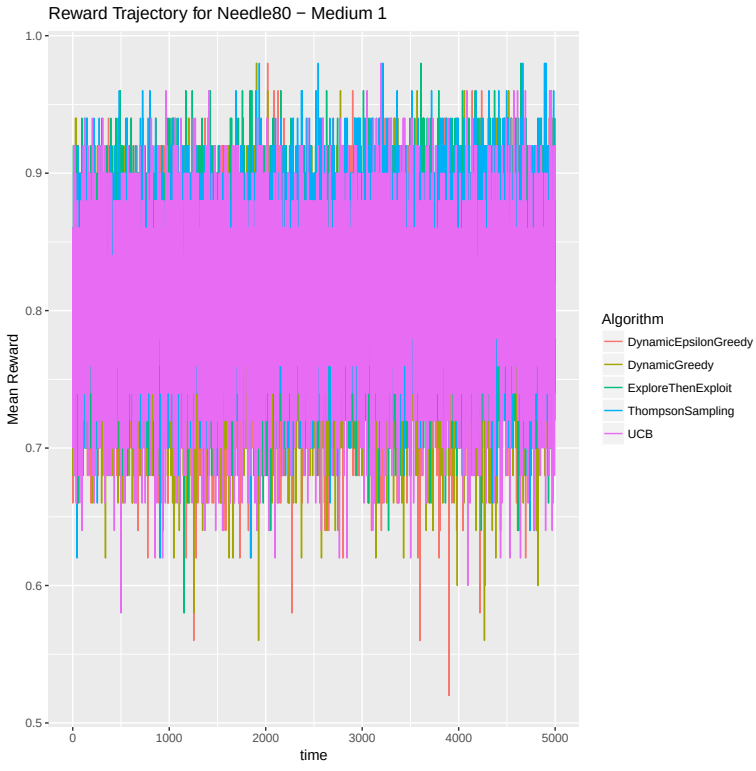
The first run considered $N = 50$ runs. The main use of this run was to help us detect any qualitative differences between the different needle-in-haystack regimes we Originally we considered three regimes of "needle-in-haystack" instances since we weren't sure if learning would be different between the different regimes. We considered so many needle-in-haystack instances to see if there would be meaningful qualitative differences between learning when the means were at extremes vs when most means were in the middle of the range of possible means.

The plots for all the algorithms are reported below. The "low" Δ in the needle-in-haystack appears uninteresting across all considered means, likely because the Δ of 0.01 is too small to be meaningful. For the "medium" instances it appears as though there is a qualitative difference between the "low" mean and the others. Though this may be partially due to low N, it appears as though there is some divergence particularly between UCB and Thompson Sampling. For the high Δ it the "low" mean and "medium" mean seem to have qualitatively similar results. The "high" mean has starker differences in terms of the learning rates of the algorithms.

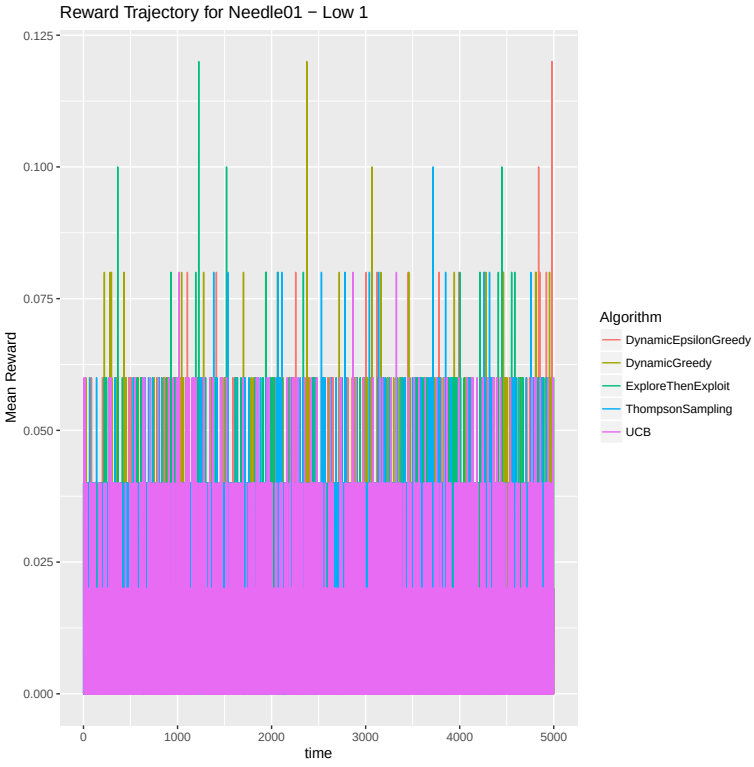
First, looking at the medium range:

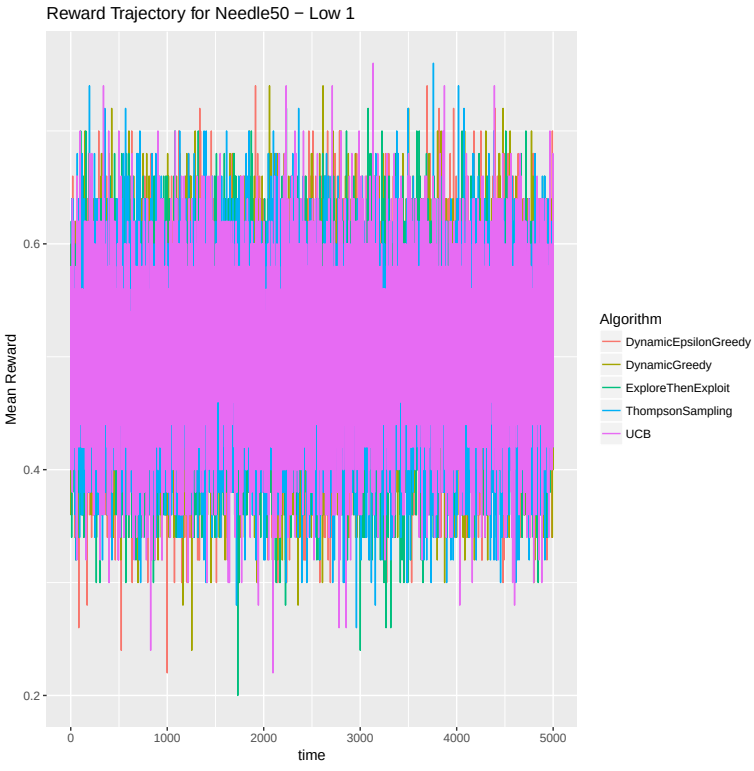


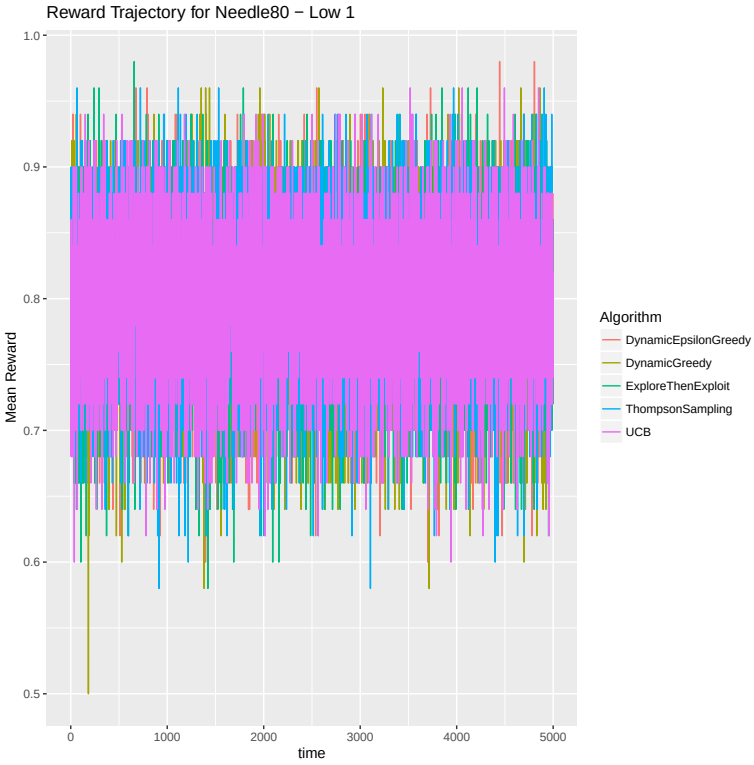




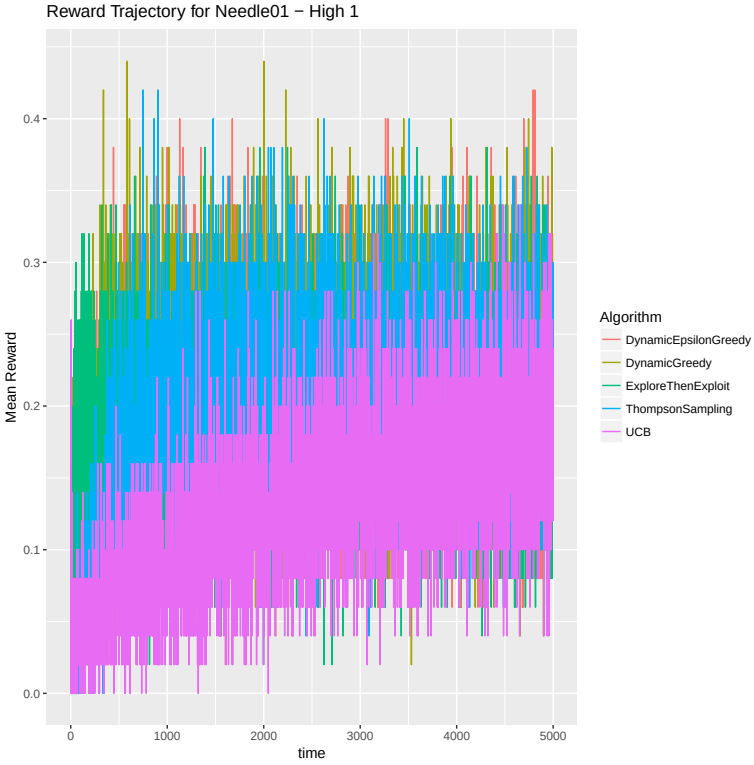
The low range:

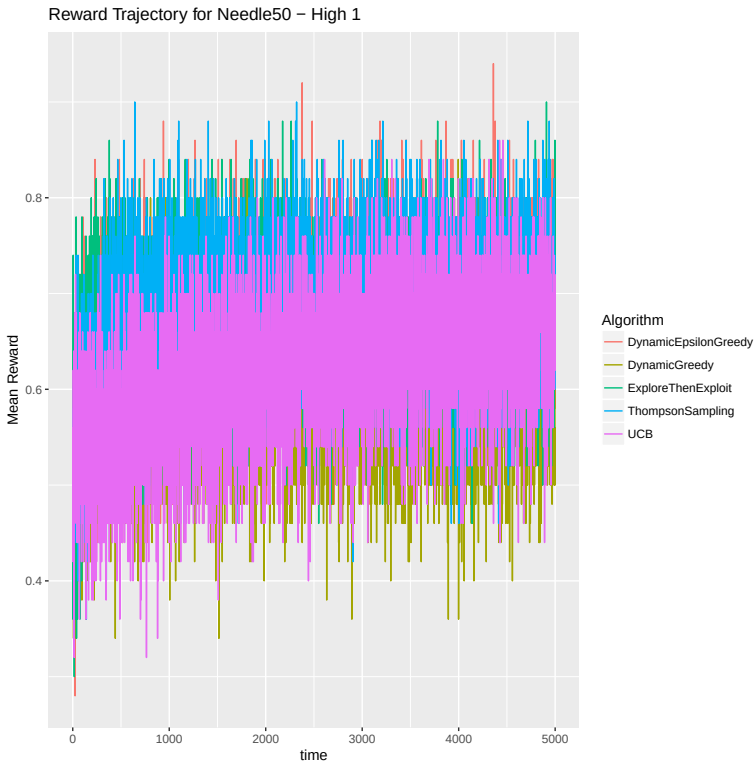

Reward Trajectory for Needle01 − Low 1
Mean Reward
0.125
0.100
0.075
0.050
0.025
0.000
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB

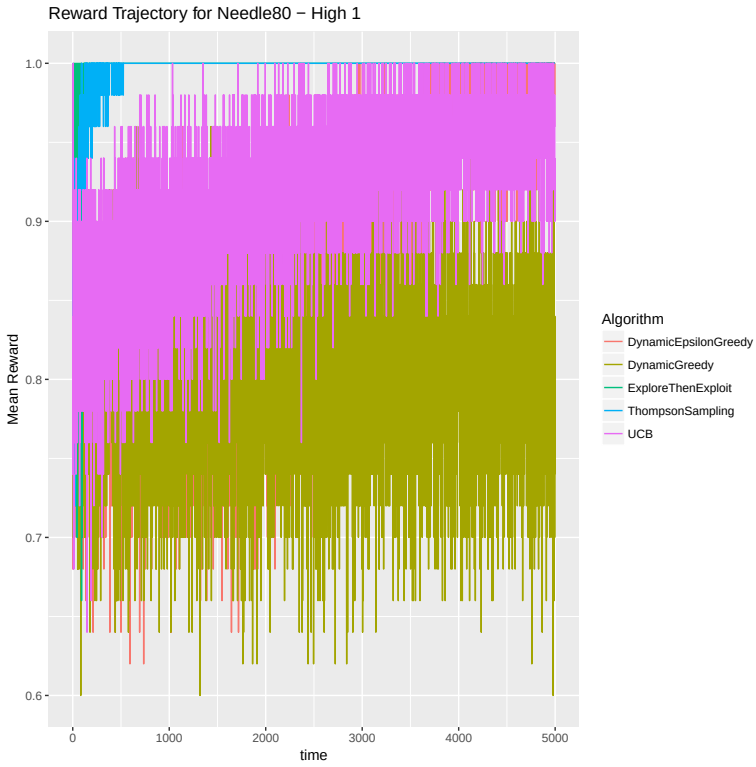

Reward Trajectory for Needle50 − Low 1
Mean Reward
0.6
0.4
0.2
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB



The high range:



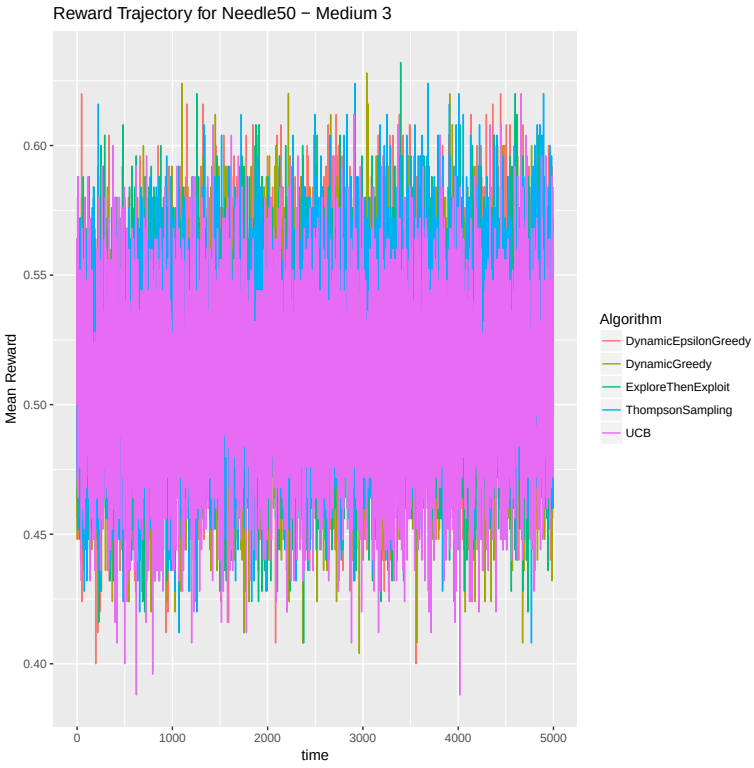


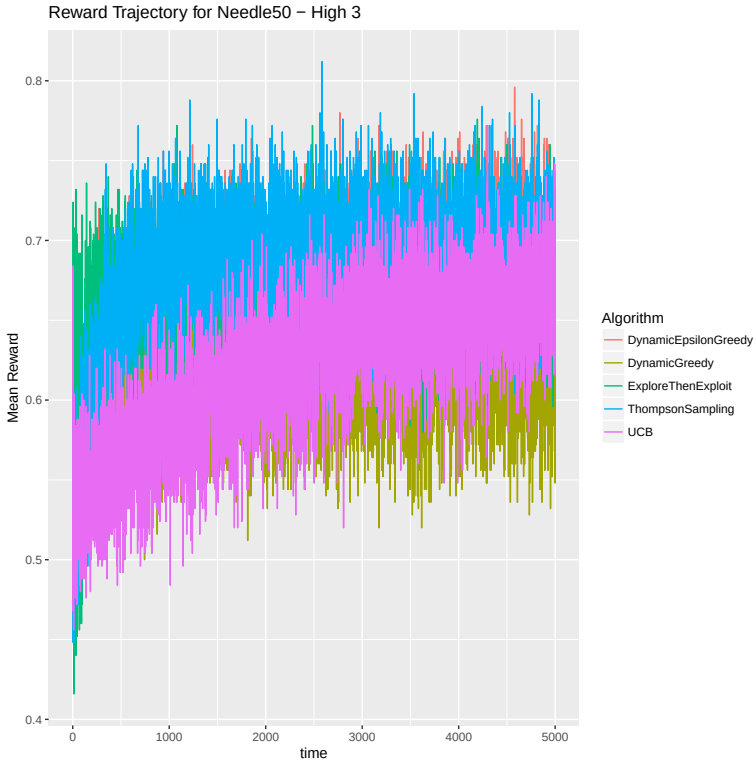


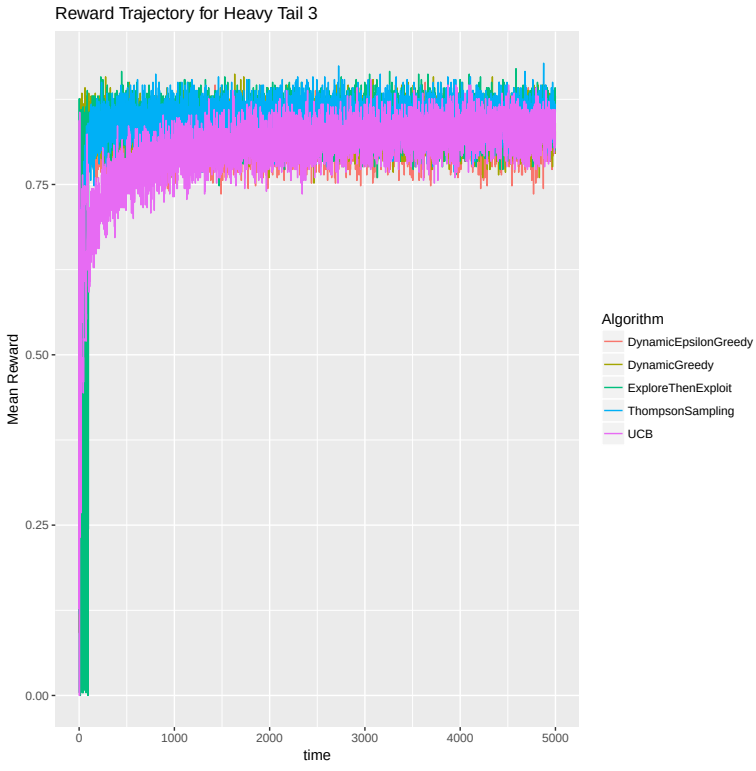
Surprisingly with $N = 50$ there appeared to be considerable variability in the trajectories and so the experiment was reran with $N = 250$ and $T = 5000$. To reduce the amount of time this

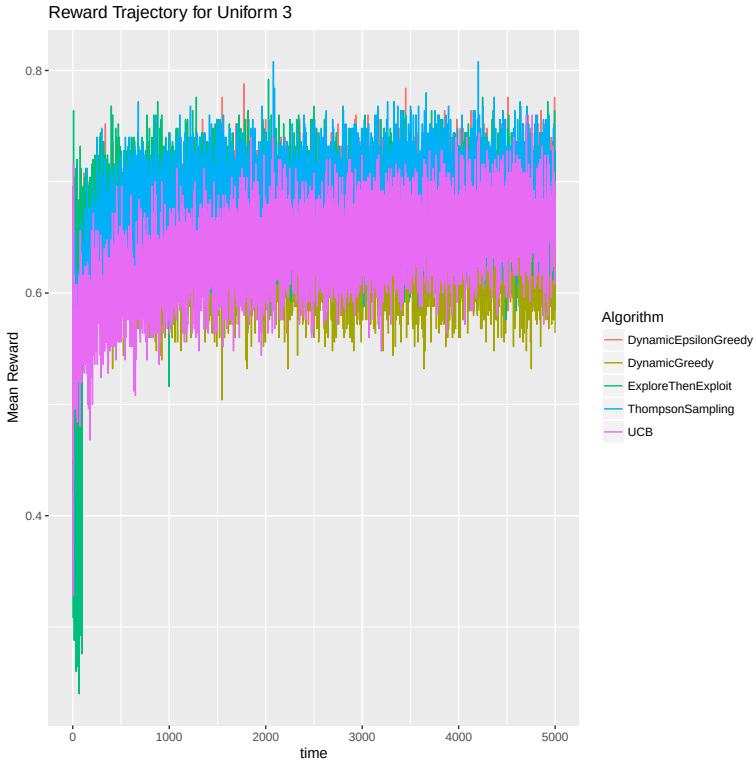
took, only Uniform, HeavyTail, Needle50 - Medium, and Needle50 - High were considered. Though there were some qualitative differences between the different needle in haystack instances discussed above, they were omitted here but we can run this again with those if we decide we want some more fine-grain information.

The plots with all the algorithms are reported below:

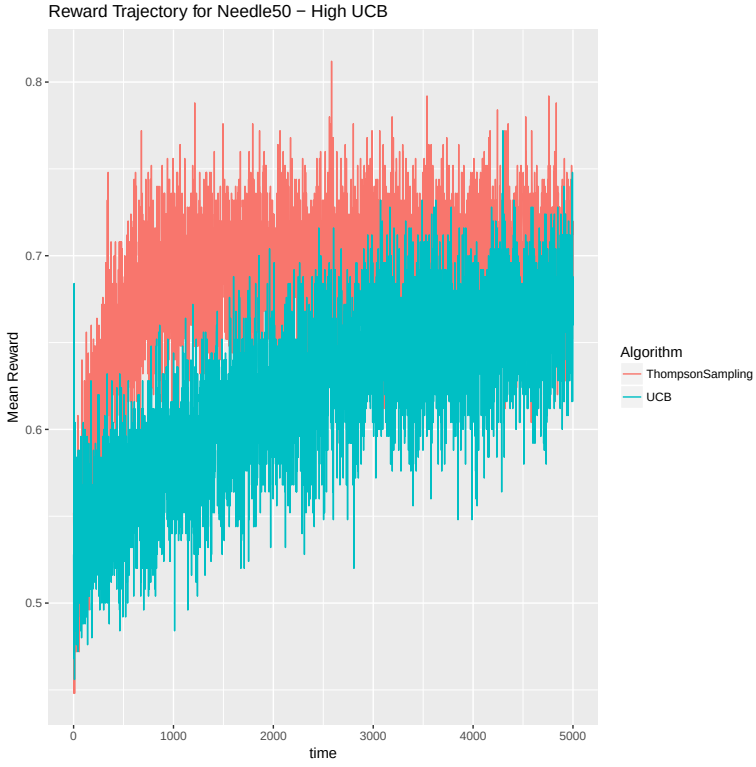


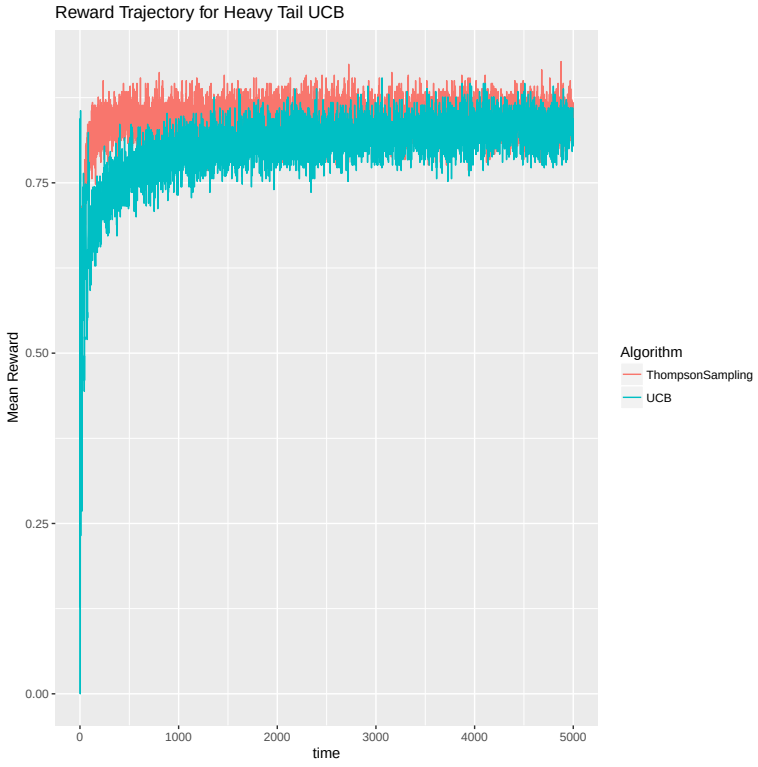

Reward Trajectory for Needle50 − High 3
Mean Reward
0.8
0.7
0.6
0.5
0.4
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB


Reward Trajectory for Heavy Tail 3
Mean Reward
0.75
0.50
0.25
0.00
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB

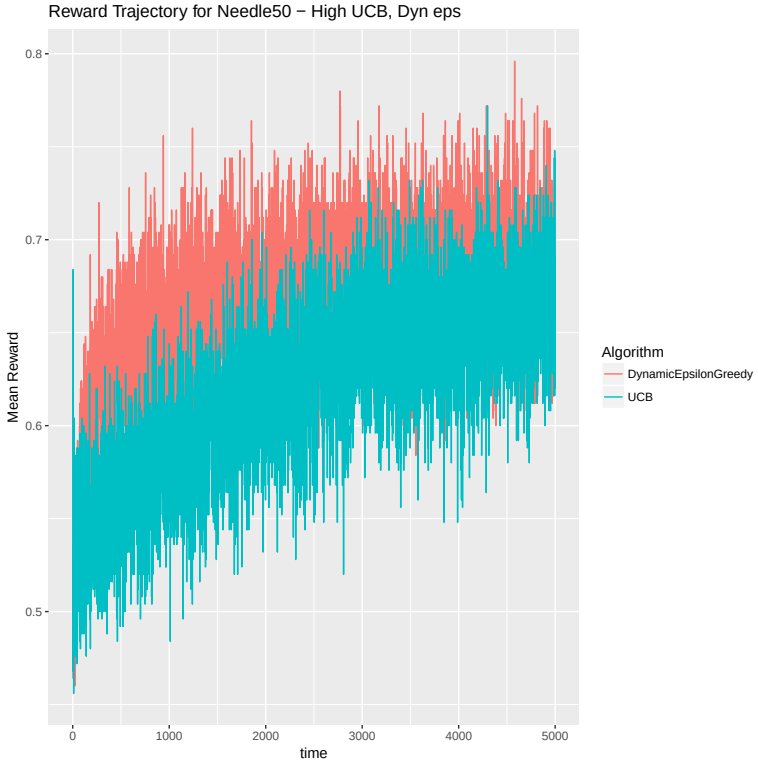


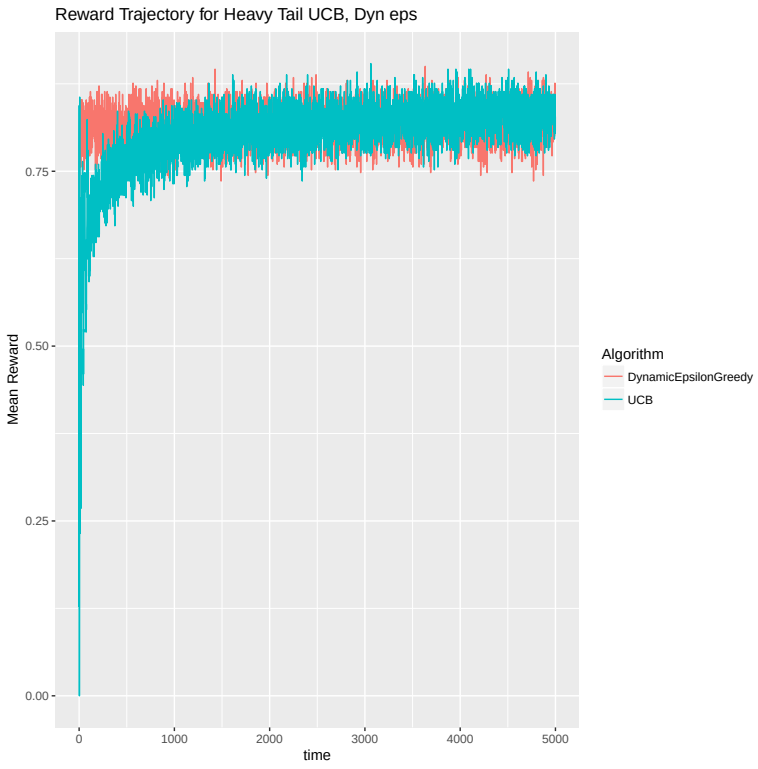
A few particular striking comparative plots:





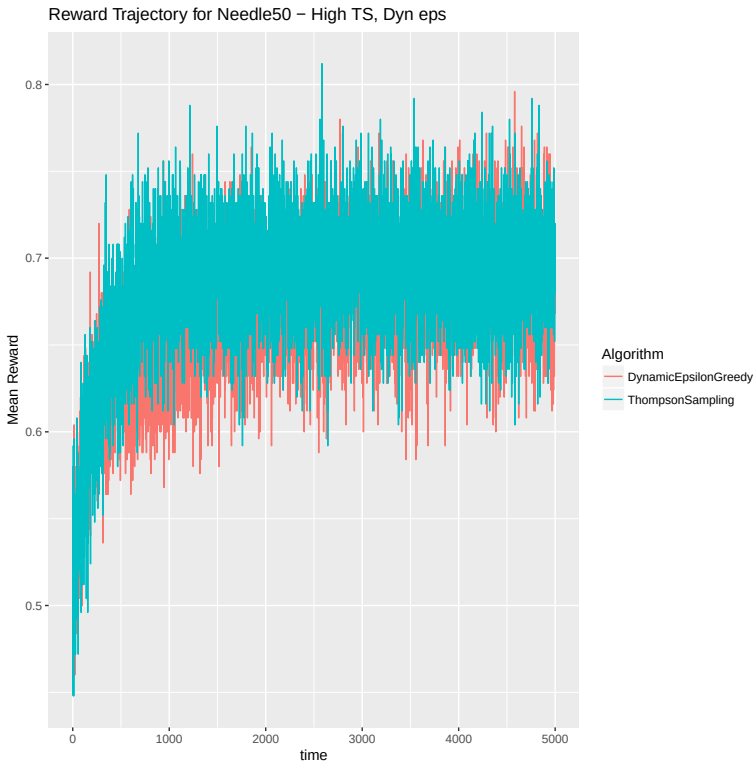
How does UCB compare against Dynamic ϵ-greedy in these cases?

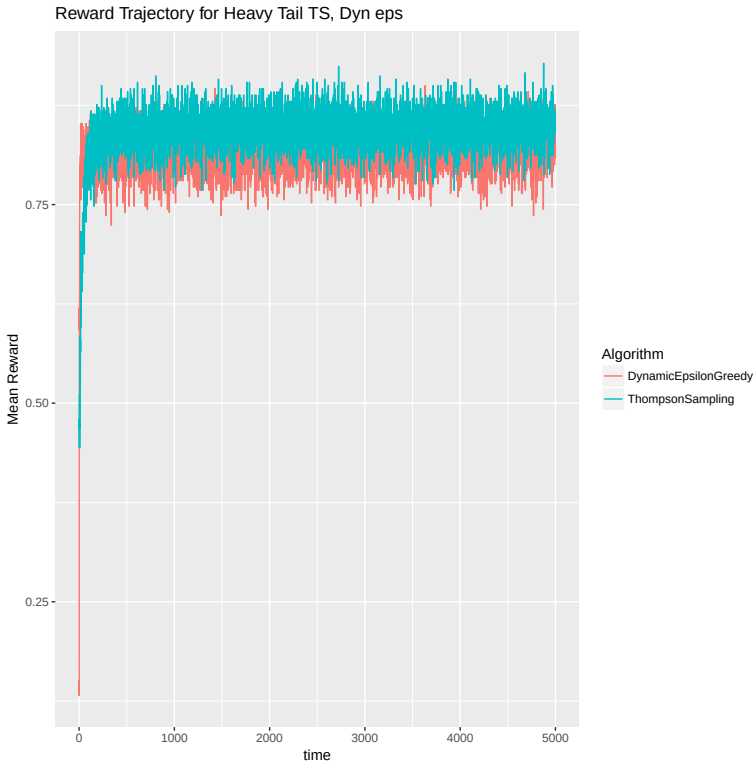




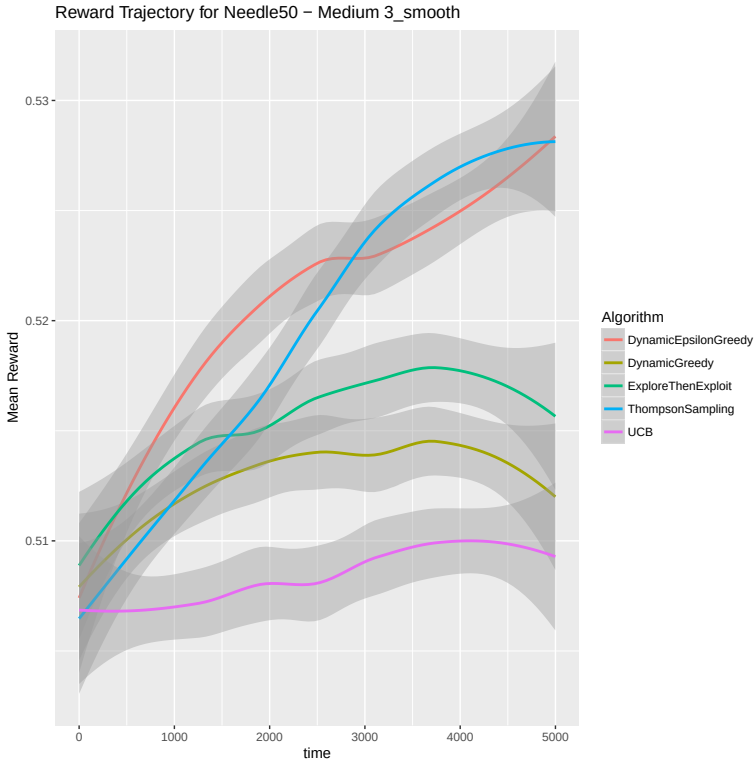
The rate of learning for UCB seems slower than for the other algorithms. Is this expected (or do we need to tune the UCB1 constant - I don't think so)? This potentially explains the results we saw that UCB1 loses in the competing bandits game because it's rate of learning growth for these bandit instances is slower than the other algorithms.

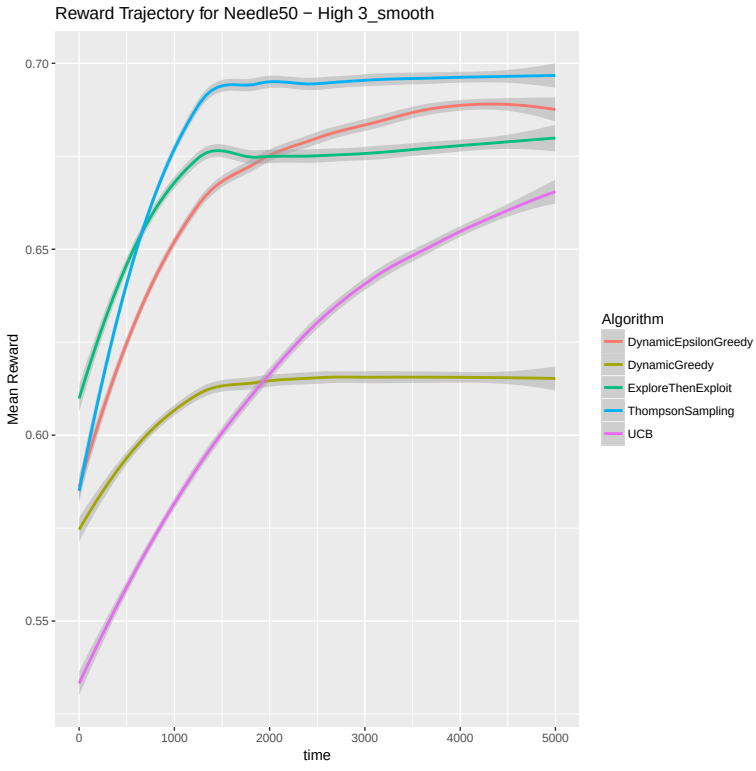
Comparing ThompsonSampling to Dynamic ϵ-greedy:

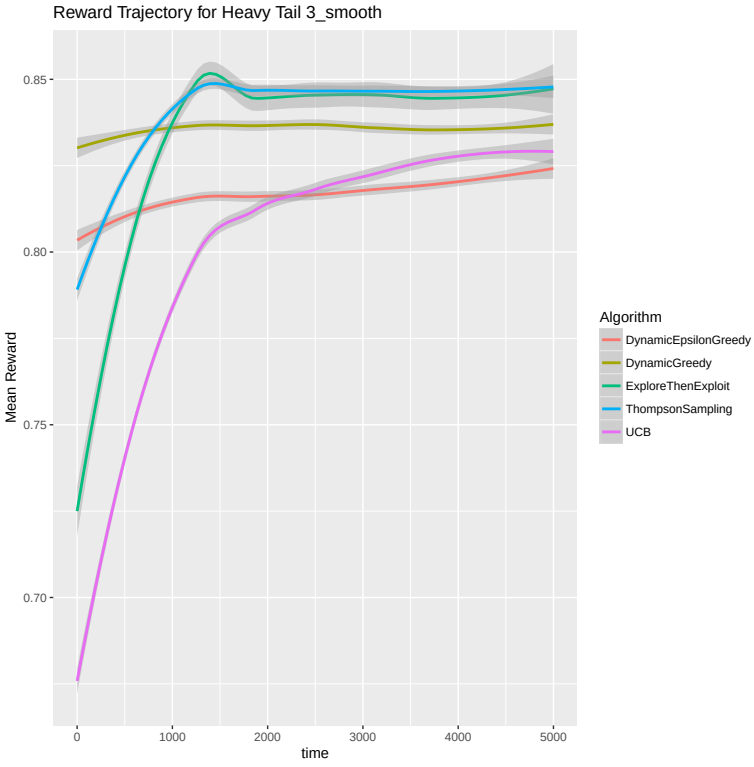


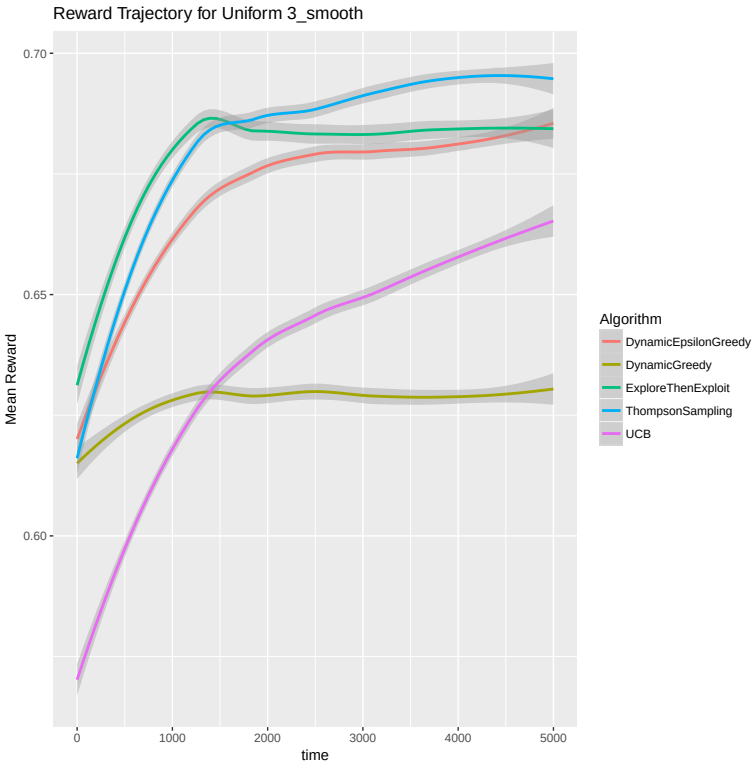


As well, a version of the plots that smooths the trajectory:


Reward Trajectory for Needle50 − Medium 3_smooth
Mean Reward
0.53
0.52
0.51
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB


Reward Trajectory for Needle50 − High 3_smooth
Mean Reward
0.70
0.65
0.60
0.55
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB


Reward Trajectory for Heavy Tail 3_smooth
Mean Reward
0.85
0.80
0.75
0.70
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB


Reward Trajectory for Uniform 3_smooth
Mean Reward
0.70
0.65
0.60
0
1000
2000
3000
4000
5000
time
Algorithm
DynamicEpsilonGreedy
DynamicGreedy
ExploreThenExploit
ThompsonSampling
UCB