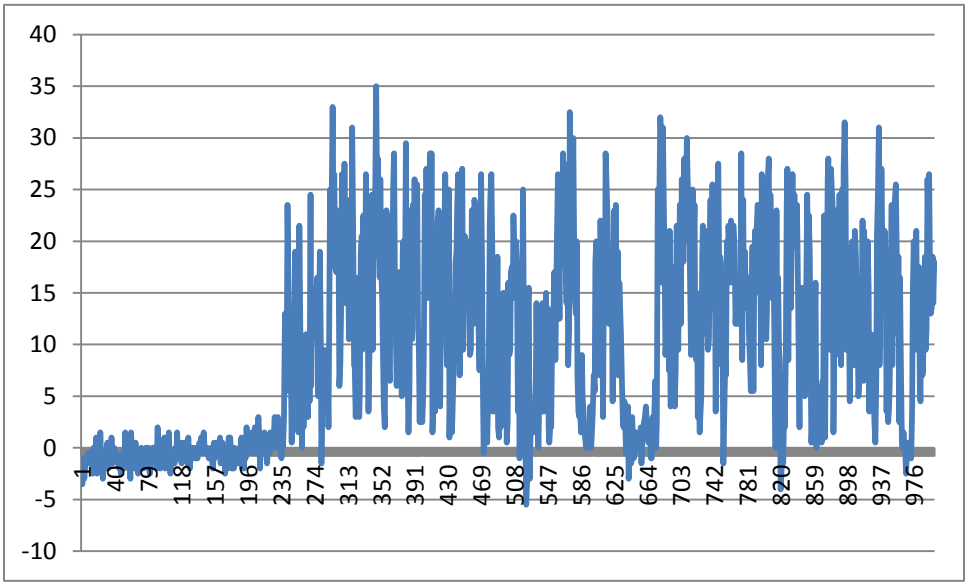
Miles Teh

Mteh2

CS440 MP 1 Part 2

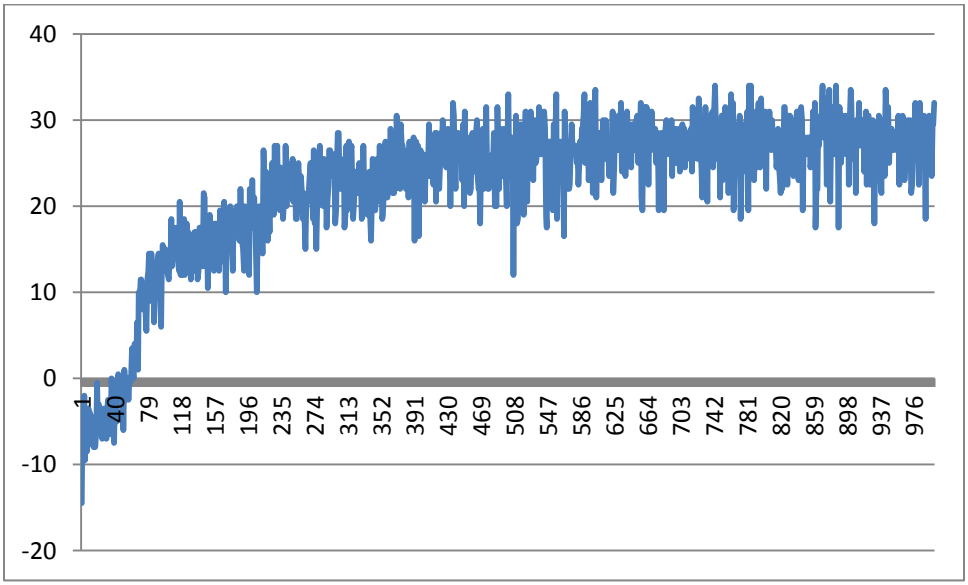
1. WorldWithoutThief
    a. For the first 1000 episodes of 1000 steps without a randomness coefficient with the exception of a few cases at the beginning all of the episodes had a result of 0. With the first few it's most likely because there are a few choices of equal QValue at the beginning. After that based on the fact there was no randomness once one path was created the bot kept going down that same path thus one gets the same score over and over again.
    b. For the first 1000 episodes of 1000 steps with an epsilon coefficient of 0.1 there are a few notable things. The average reward is definitely not stuck at 0. And for a while, while the Agent is learning the scores, the result is rather lackluster, it takes so long because of the epsilon coefficient. After that of course the reward is much higher than in in the first one because the epsilon coefficient allows the Agent to explore new routes rather than always take one.




    c. As one increases the value of epsilon the average value of the reward decreases. I saw this after testing values of epsilon 0.2, 0.3, 0.4, and 0.5. Apart from epsilon 0.0 which is an odd case for 0.1 – 0.5 the average value of the reward decreases from about 15 to 0. This is probably caused by the fact of having a greater randomness coefficient the agent is more likely to mess up during its episode and thus reduces the reward in general.
2. WorldWithThief
    a. The performance in the WorldWithThief without knowing the thief is absolutely terrible. The average reward is negative, this is probably because without knowing the thief the

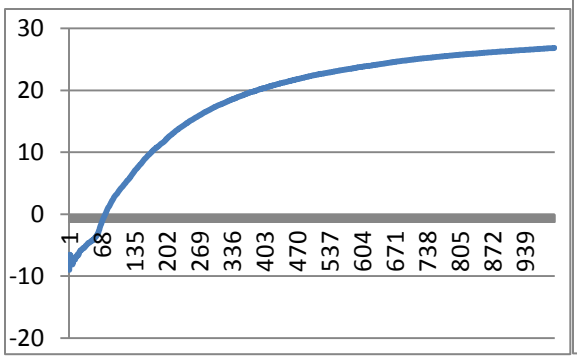
bot is unable to adapt to the movements of any thieves and thus has no better chance at any time to dodge the thieves.

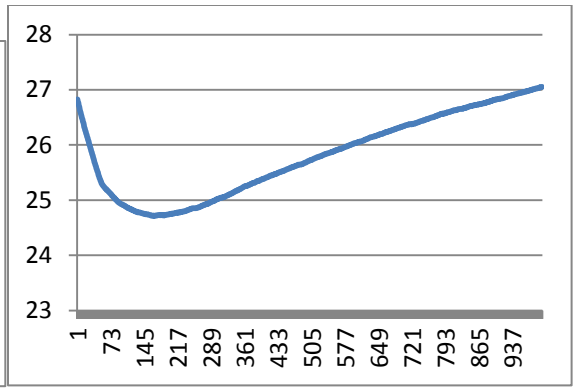
b. The performance of the bot while knowing the thief is excellent. The value of the rewards constantly increases till it seems to hit the ceiling at a reward of approximately 30. The performance difference is mainly because the agent can now change its movements based on the thief's movement allowing it to efficiently dodge the thief. The path it takes also becomes optimized for dodging thieves because the paths that are more likely to catch paths will generally have a lower Q value.

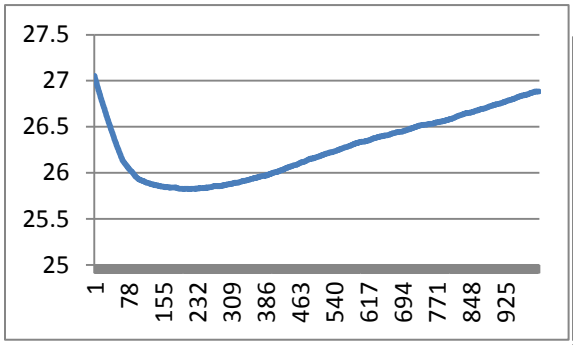


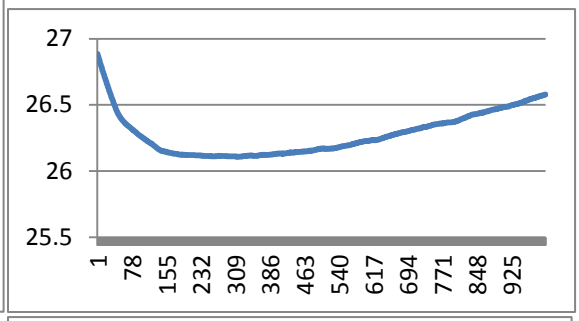

c. After a little testing I found the best values of learning rate and epsilon to be 0.2 and 0.02 respectively. The way I found this is first I went through a list of learning rates from 0.05 to 0.5 going up by a value of 0.05. At certain points the value climbed faster but if one increased the learning rate too much the average reward would decrease. So finding the balance was necessary and that balance was found at 0.2. After that I tested different values of epsilon. Those values included 0.001, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5. For most values the average value would decrease too much certain values of epsilon. The best was at 0.02 where the value increased to a little over 30.

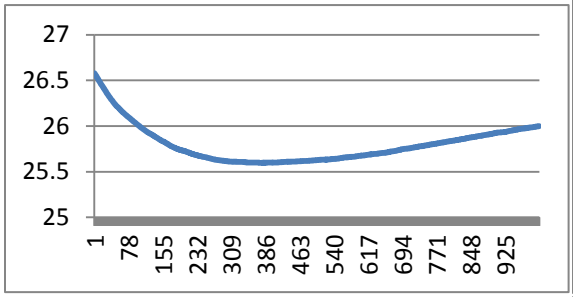
3.

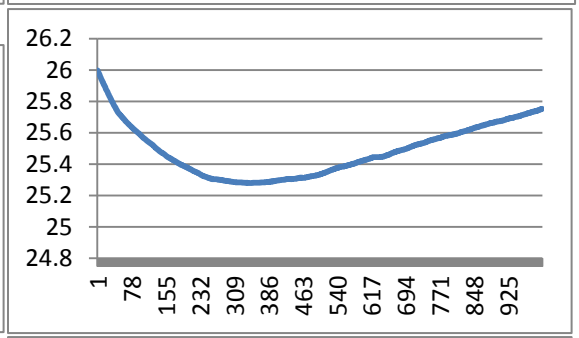

30
20
10
0
-10
-20
1
68
135
202
269
336
403
470
537
604
671
738
805
872
939

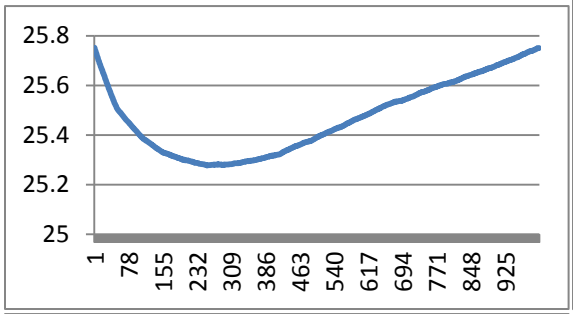

28
27
26
25
24
23
1
73
145
217
289
361
433
505
577
649
721
793
865
937

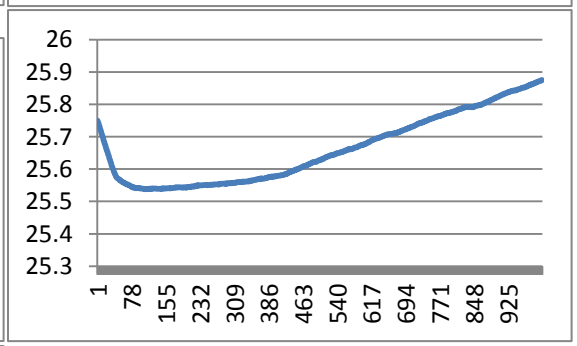

27.5
27
26.5
26
25.5
25
1
78
155
232
309
386
463
540
617
694
771
848
925

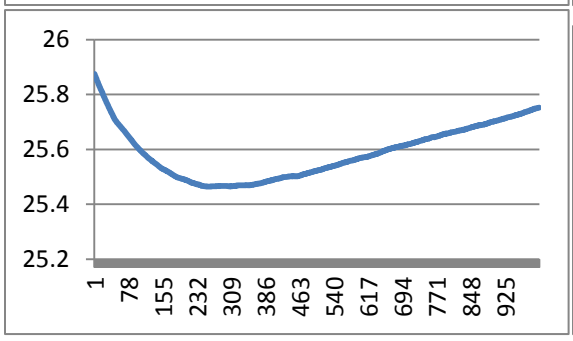

27
26.5
26
25.5
1
78
155
232
309
386
463
540
617
694
771
848
925

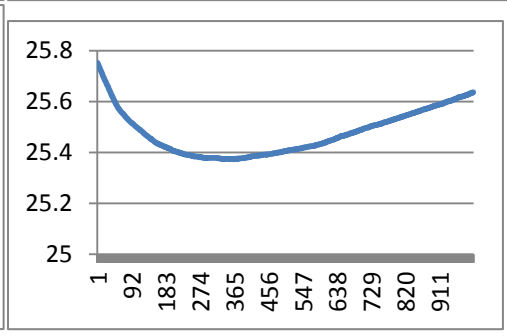

27
26.5
26
25.5
25
1
78
155
232
309
386
463
540
617
694
771
848
925


26.2
26
25.8
25.6
25.4
25.2
25
24.8
1
78
155
232
309
386
463
540
617
694
771
848
925


25.8
25.6
25.4
25.2
25
1
78
155
232
309
386
463
540
617
694
771
848
925


26
25.9
25.8
25.7
25.6
25.5
25.4
25.3
1
78
155
232
309
386
463
540
617
694
771
848
925


26
25.8
25.6
25.4
25.2
1
78
155
232
309
386
463
540
617
694
771
848
925


25.8
25.6
25.4
25.2
25
1
92
183
274
365
456
547
638
729
820
911

The simulations are in order from top to bottom on the left then right 1,2,3,4,5,6,7,8,9,10. The major thing to notice was that after the  first two runs or so the average started decreasing again. At simulation 1 and 2 it gets up to ~27 average reward and after that it comes down to maybe ~25.5 expected reward. Honestly I'm not completely sure why this happens I would assume as we increase the value we get closer and closer to the actual average reward. And the first two were simply slight outliers whose average was higher than normal.