Assignment 1

Parth Bhardwaj

Part 1

```
policy iteration:
  gamma = 0.95: iterations = 20, calls = 290400, mean reward = 47, std = 21.8
  gamma = 0.5: iterations = 25, calls = 110000, mean reward = 32, std = 57.24
  gamma = 0.3: iterations = 23, calls = 91200, mean reward = 32, std = 57.24
value iteration:
  gamma = 0.95: iterations = 31, calls = 86800, mean reward = 47, std = 21.8
  gamma = 0.5: iterations = 27, calls = 75600, mean reward = 32, std = 57.24
  gamma = 0.3: iterations = 17, calls = 47600, mean reward = 32, std = 57.24

We expect both algorithms to converge to one of the optimal policy and hence give
the same results on evaluation, as observed

policy in both algorithms is same at gamma = 0.5, 0.95 but different at gamma =
0.3 - although the reward statistics are identical. That is because due to low
gamma the unimportant states for away from centre have no impact on the
performance, so we might as well choose right instead of left at those states,
which doesn’t make difference in evaluation but policy becomes different

Change in gamma: improves convergence speed but reduces reward- due to
discounting at each stage. Convergence is improved because effect of states do
not propagate as far as in high gamma, because gamma ^ n << 1 for small gamma

Non Stationary value iteration: calls = 336000
Time dependent policy results: mean = 37.7, std = 39.6
Time independent policy results: mean = 23.65. Std = 44.71

GIFs are in the folder ( seed = 20): for stationary part both algorithms give
same output.gif. For non-stationary, time independent gif is output_time_ind.gif
and time dependent is output_non.gif

Modified Value iteration: the strategy is to visit those states more often whose
value function is much farther from true optimal value function. To do that, we
use a priority queue with priorities of states determined by the change in value
function in the previous iteration: higher change corresponds to farther distance
from optimal value function. We keep peeking the priority queue and updating
using Bellman Updates until the max delta is less than tolerance.
```

```
Calls with this = 30219, a reduction by a factor of 1/3. Time taken is also much
lesser
```

Part 3: Portfolio optimization

State space: holding, cash, price of asset

We did not take time into state space because in general market there is no concept of end of market like in our case, aim was to build time agnostic policy

We also assume the following probability of price at next time given price currently:

Find Normal distribution centered at current price with standard deviation as a model parameter. Then the possible next prices are assumed to be integers in the range: [max(0,p−max_jump),min(max_price,p+max_jump)]

Where max jump (10) and max price(100) are model parameters.

So we find probability at these points and then normalise them to sum up to 1

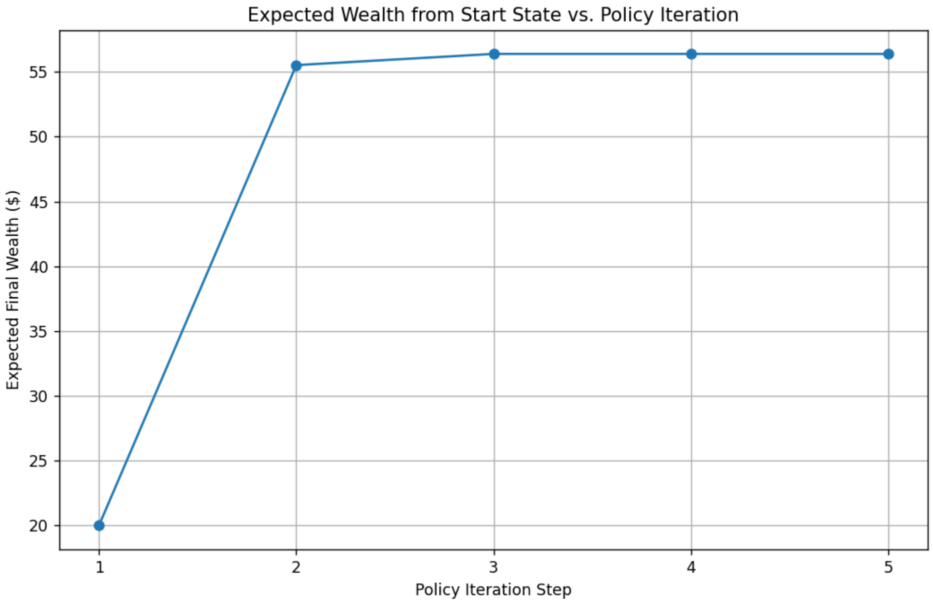
Reward: transaction cost and terminal reward

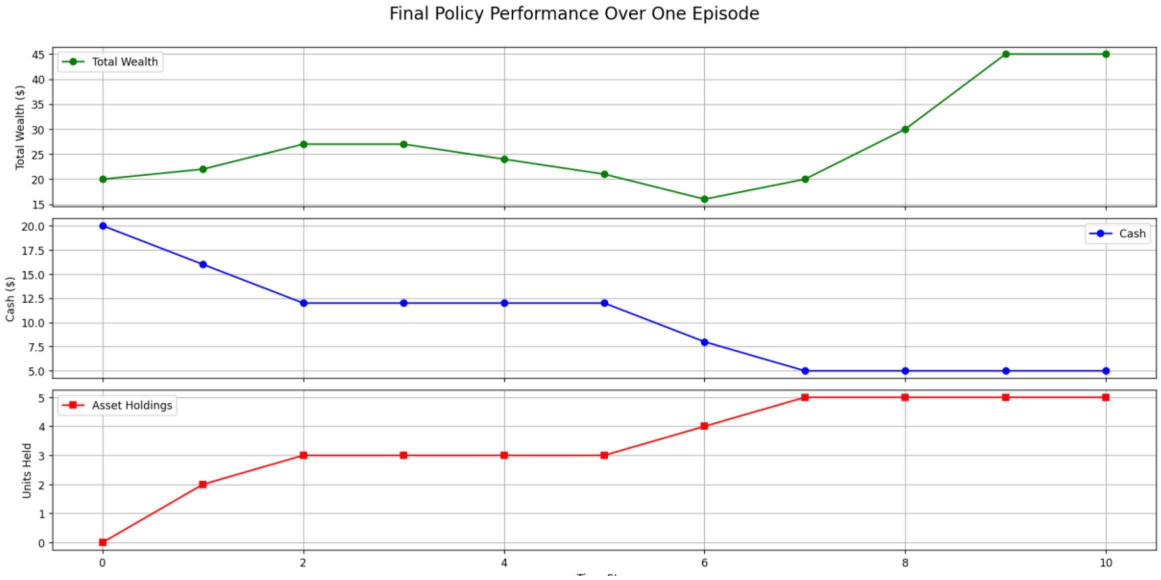
First we initialise value function to be c + h*5 at any state (h,c,p) and initialise policy using heuristic: sell 1 if price and holding high, buy 1 if price low and cash high

Algorithm: We run trial-based asynchronous value iteration: Choose action greedily, visit the next state, update the value function using bellman step and repeat till we reach terminal state. Repeat this episode using new value function.

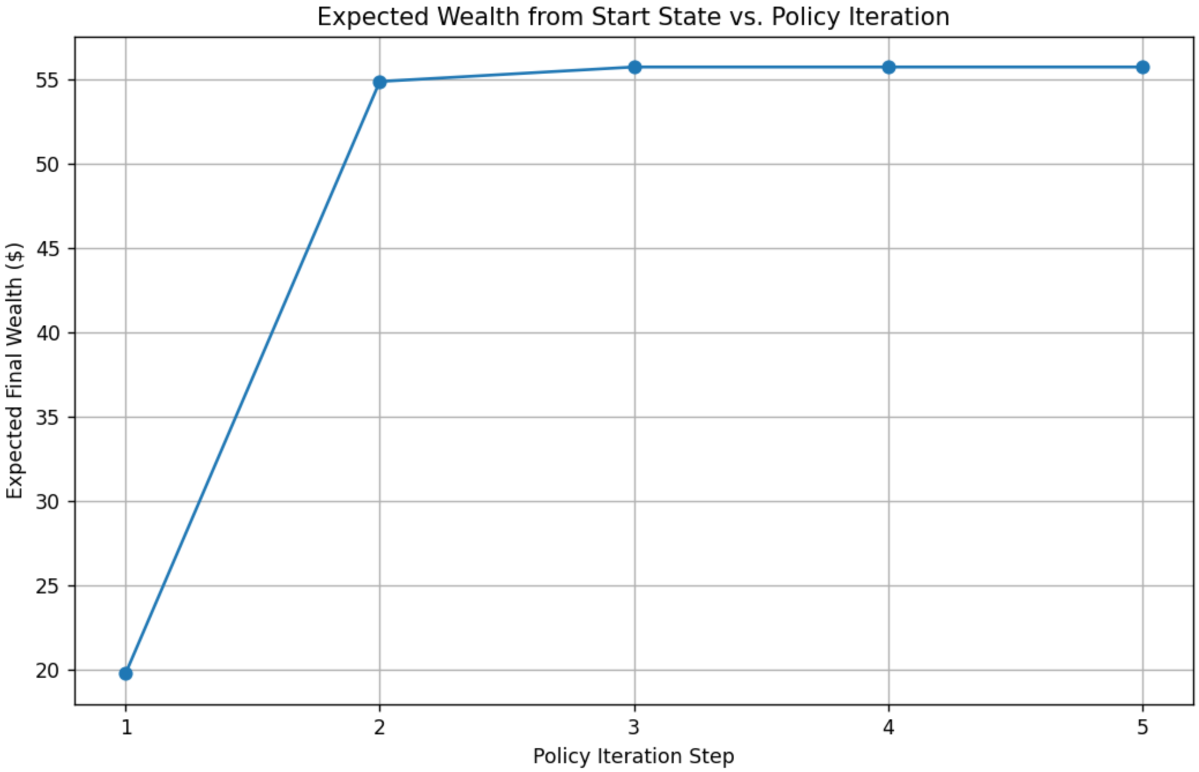
Policy Iteration:

Config 1 (gamma = 1): Policy converges in 5 steps. Total time = 225.45 seconds

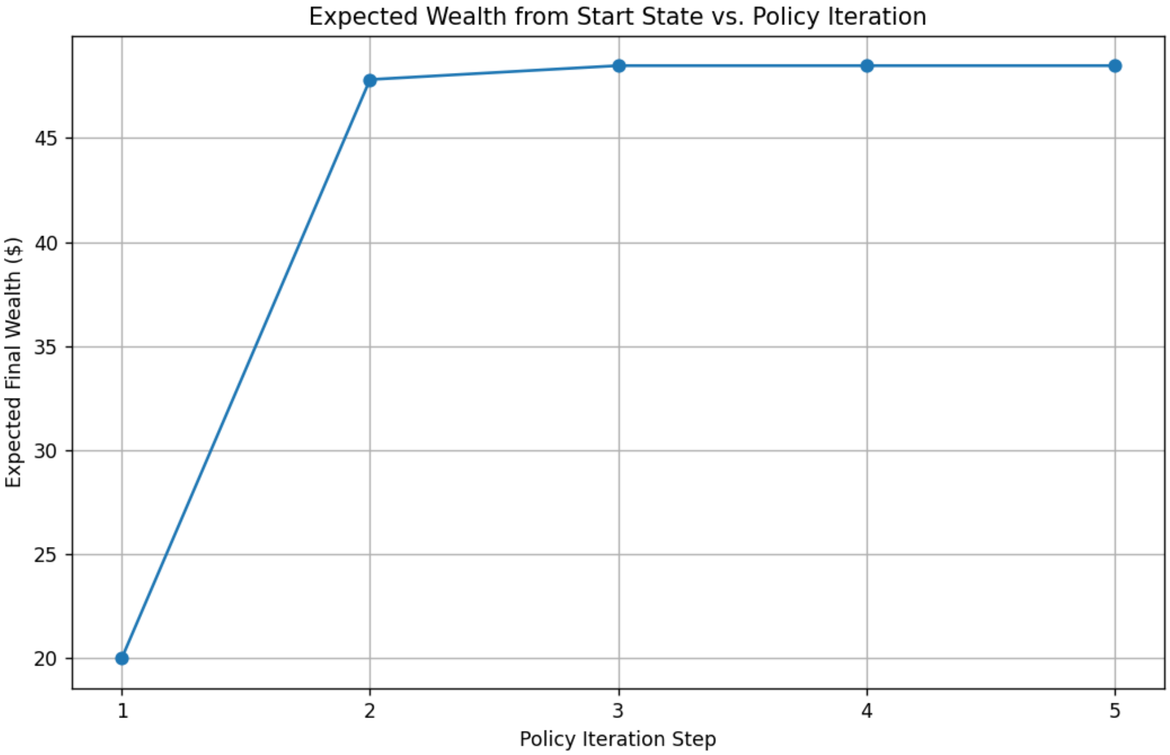


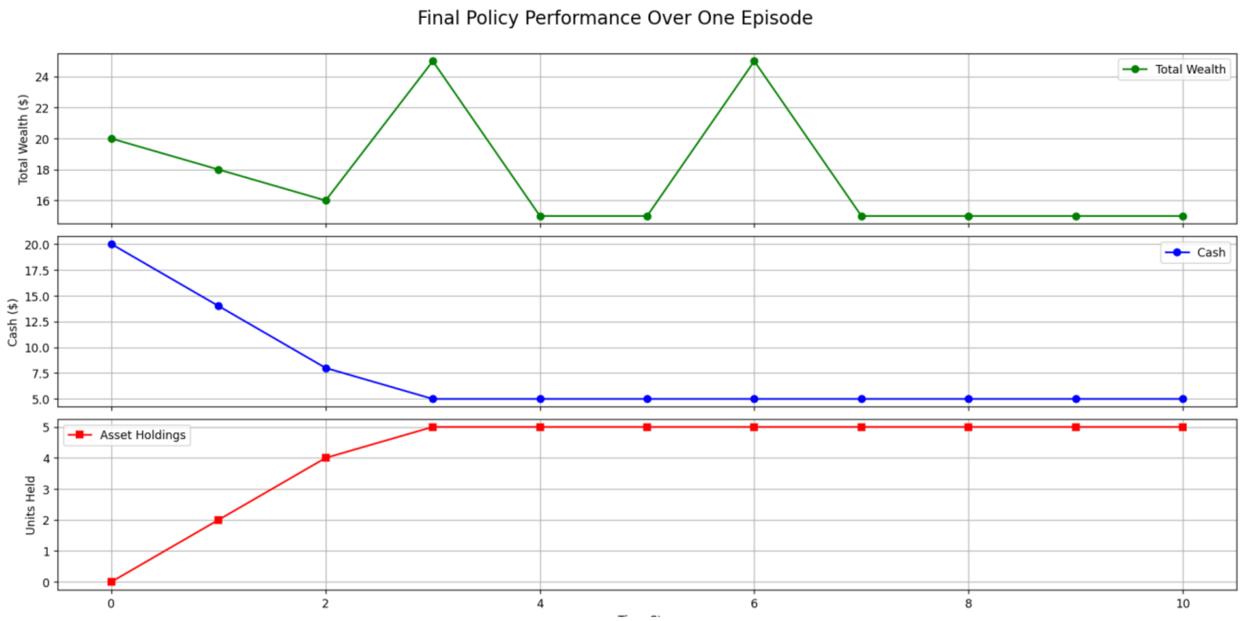


Gamma = 0.999: 235.18 seconds

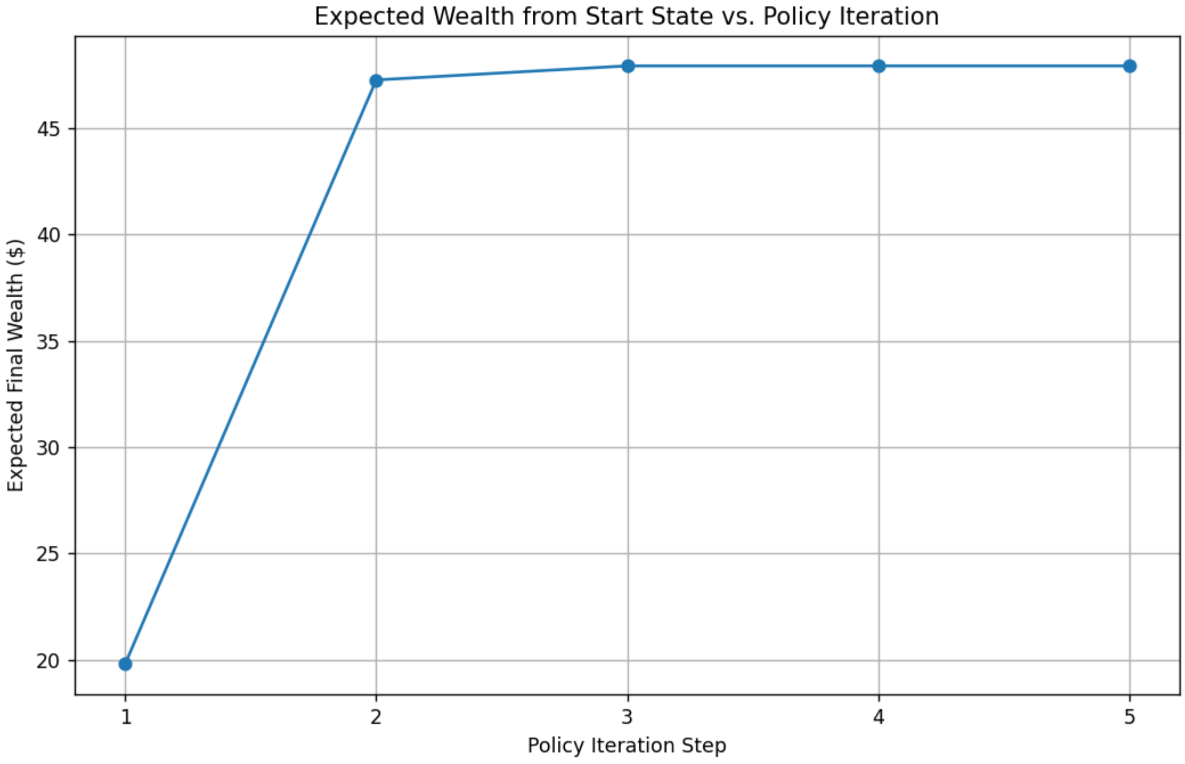


Config 2 (gamma = 1): Policy converges in 5 steps. Total time = 234.57 seconds

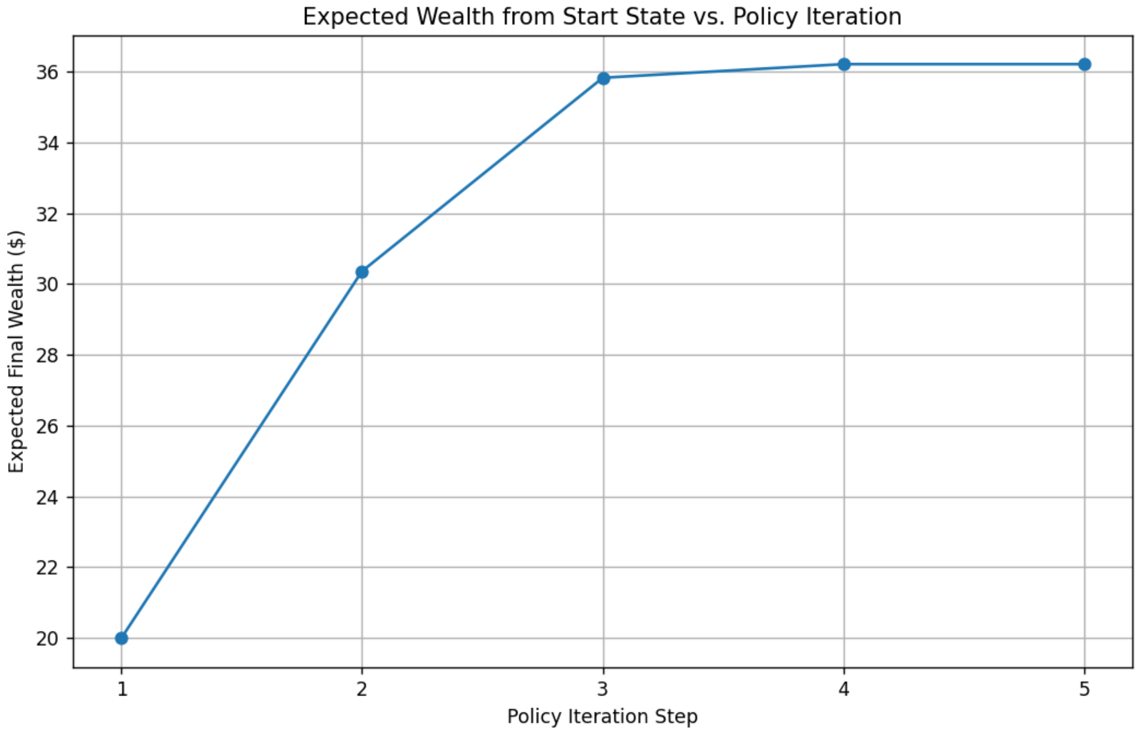


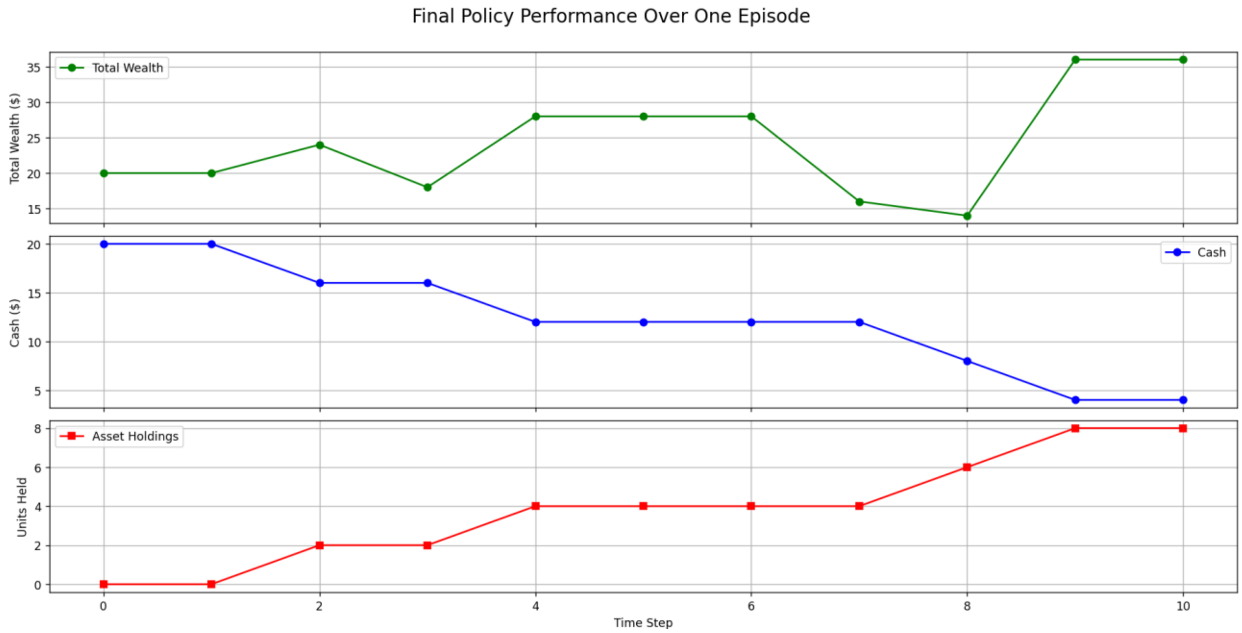


Gamma = 0.999: time 231.88 seconds

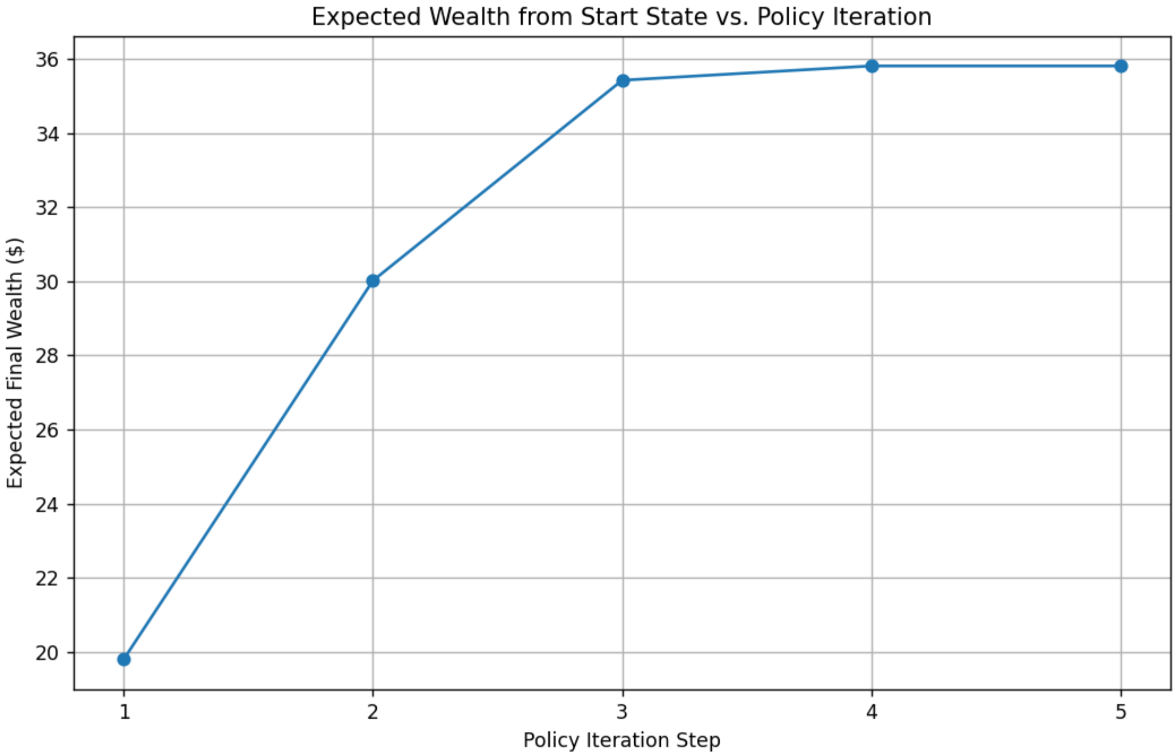


Config 3 (gamma = 1): total time = 251.47 seconds and iterations = 5

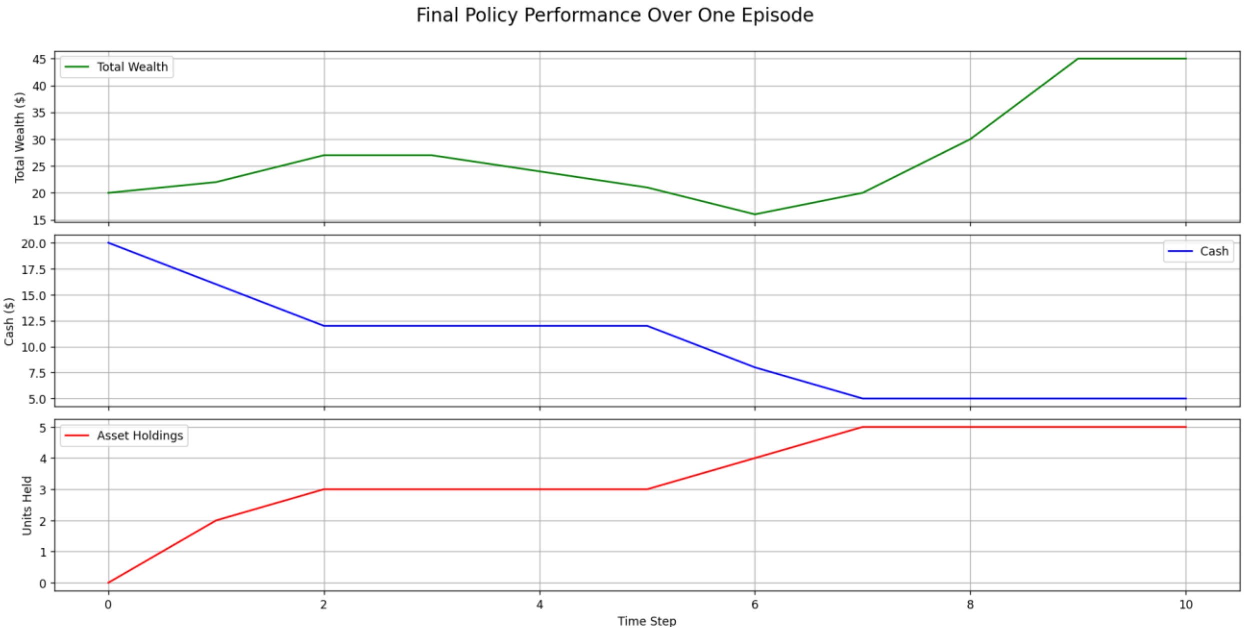




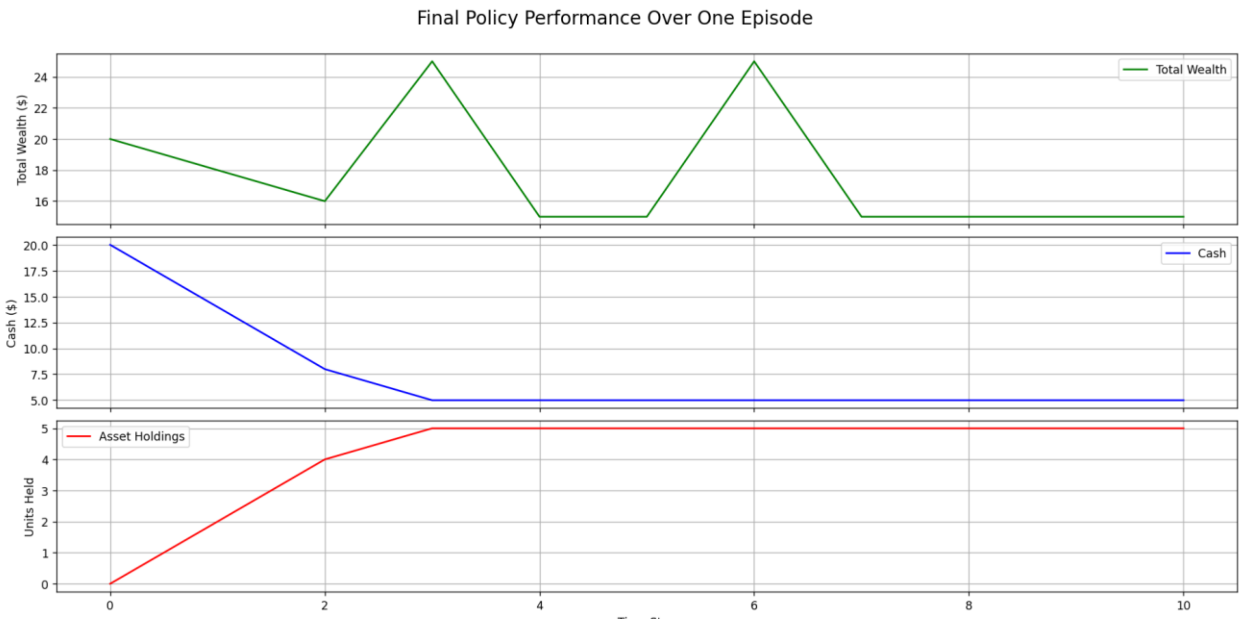
Gamma = 0.999: time = 224.51 seconds



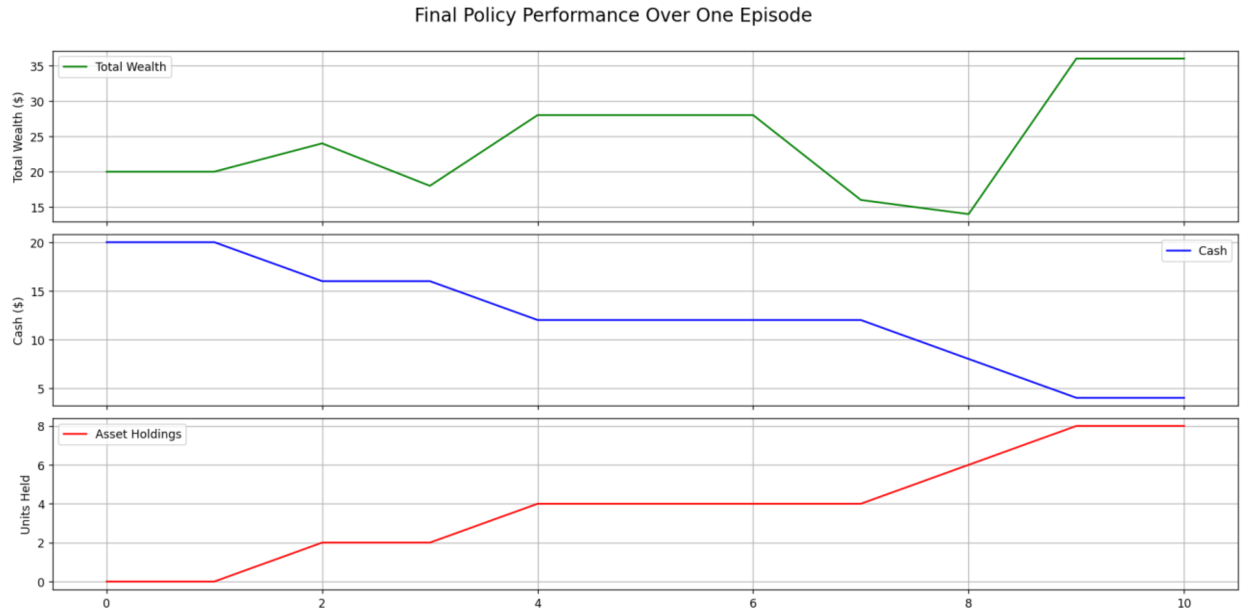
Config 1 value iteration time = 33.32 seconds:



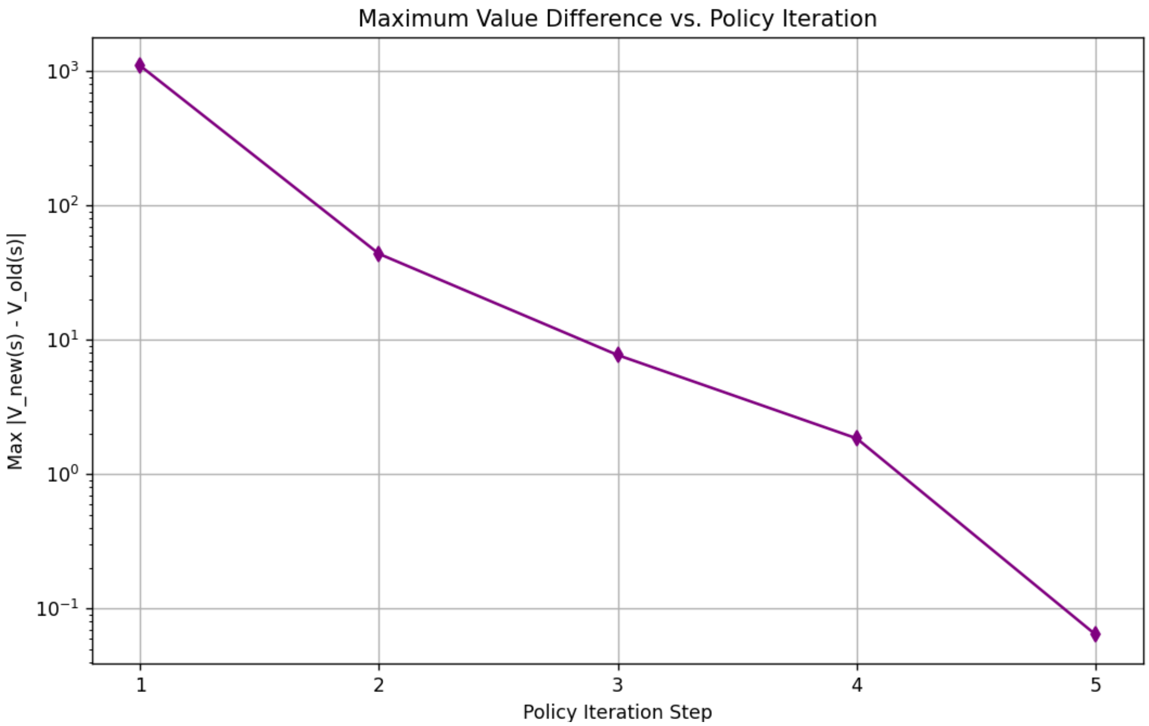
Config 2 value iteration time = 33.9 seconds:

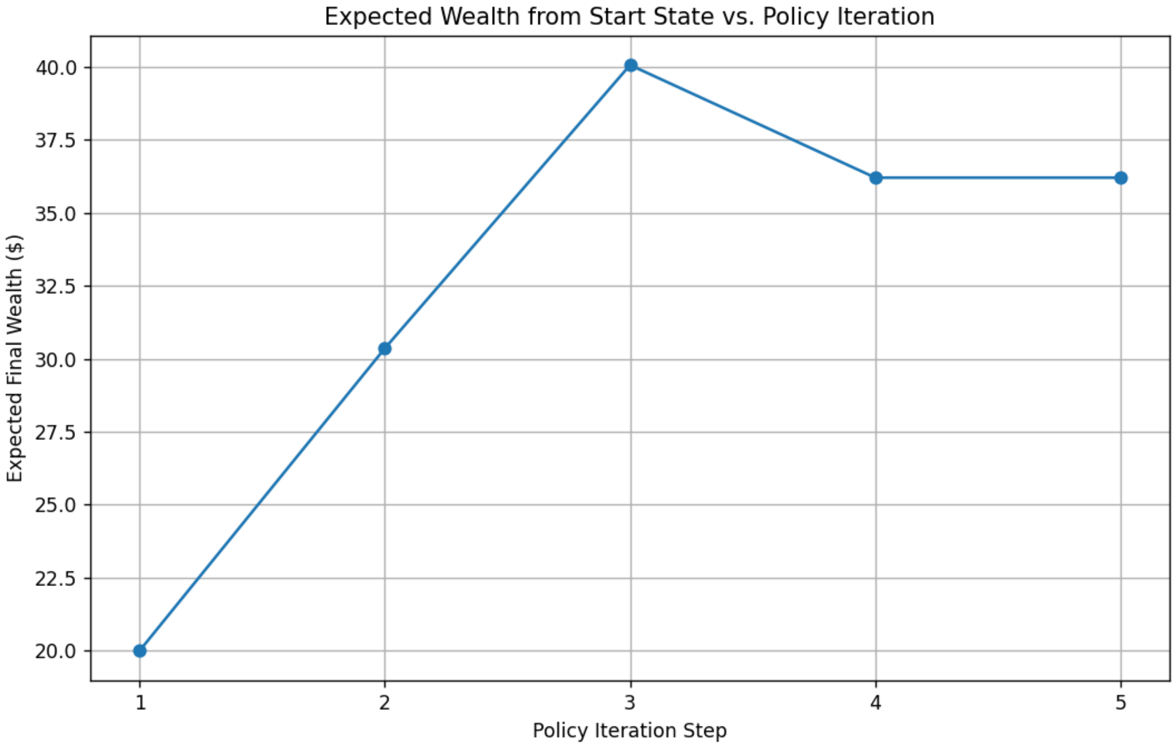


Config 3 value iteration time = 33.94 seconds:



Variance = 1: Policy Iteration converges in 240.19 seconds





Part 2:

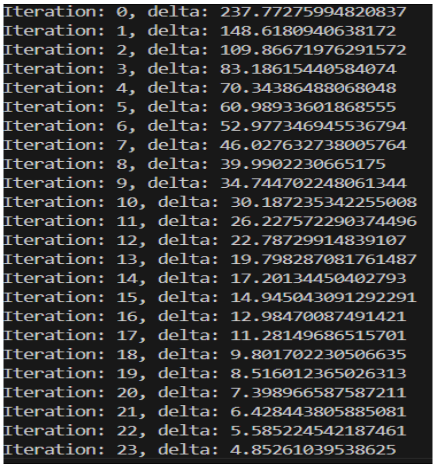
State Space: item, current_weight, time step

We assume uniform probability distribution for the upcoming item hence expected value function of future is just the average of value functions of all the possible future states

Generated a GIF for the heatmap because state is a function of time for timestep = 10, 50. Time = 500 was not supported by laptop RAM

GIFs: output_vi.gif, output_vi_10.gif, output_pi.gif

Convergence of value iteration(done till 60 iterations for delta < 0.02):



```
Iteration: 0, delta: 237.77275994820837
Iteration: 1, delta: 148.6180940638172
Iteration: 2, delta: 109.86671976291572
Iteration: 3, delta: 83.18615440584074
Iteration: 4, delta: 70.34386488068048
Iteration: 5, delta: 60.98933601868555
Iteration: 6, delta: 52.977346945536794
Iteration: 7, delta: 46.027632738005764
Iteration: 8, delta: 39.9902230665175
Iteration: 9, delta: 34.744702248061344
Iteration: 10, delta: 30.187235342255008
Iteration: 11, delta: 26.227572290374496
Iteration: 12, delta: 22.78729914839107
Iteration: 13, delta: 19.798287081761487
Iteration: 14, delta: 17.20134450402793
Iteration: 15, delta: 14.945043091292291
Iteration: 16, delta: 12.98470087491421
Iteration: 17, delta: 11.28149686515701
Iteration: 18, delta: 9.801702230506635
Iteration: 19, delta: 8.516012365026313
Iteration: 20, delta: 7.398966587587211
Iteration: 21, delta: 6.428443805885081
Iteration: 22, delta: 5.585224542187461
Iteration: 23, delta: 4.85261039538625
```

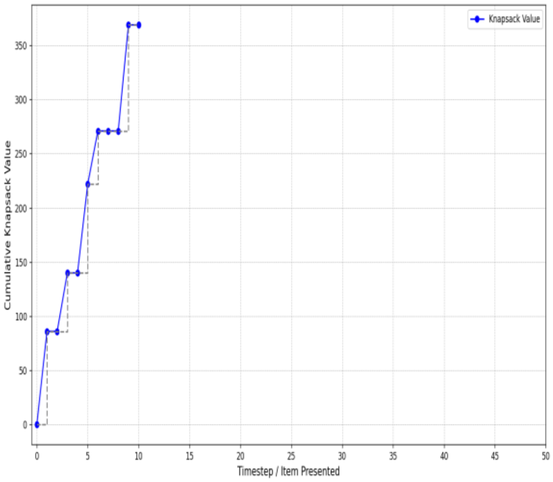
Seed 0: final value = 367, final weight = 164

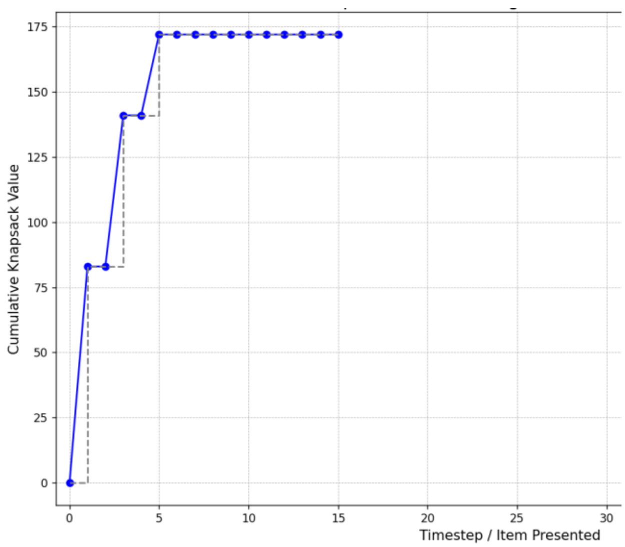
Seed 1: final value = 369, final weight = 176

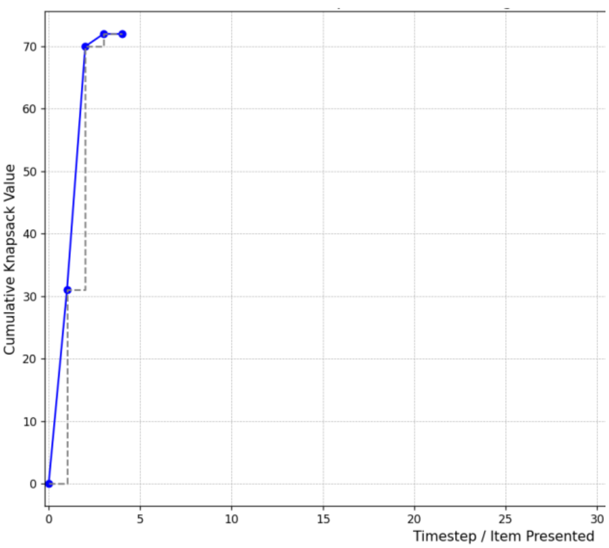
Seed 2: final value = 172, final weight = 182

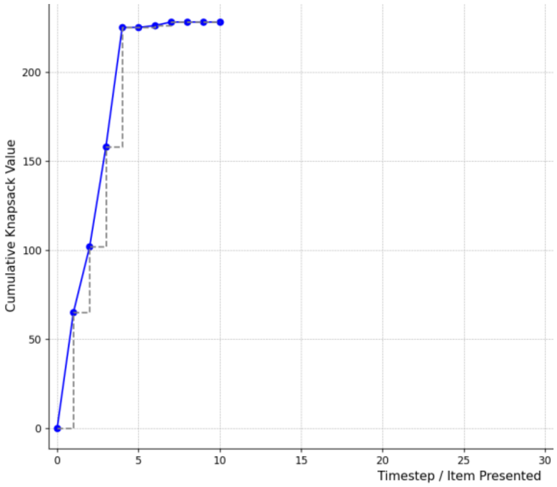
Seed 3: final value = 72, final weight = 171

Seed 4: final value = 228, final weight = 185









Policy iteration: took much more time so we couldn't go till convergence. We evaluated for 100 policy evaluation steps

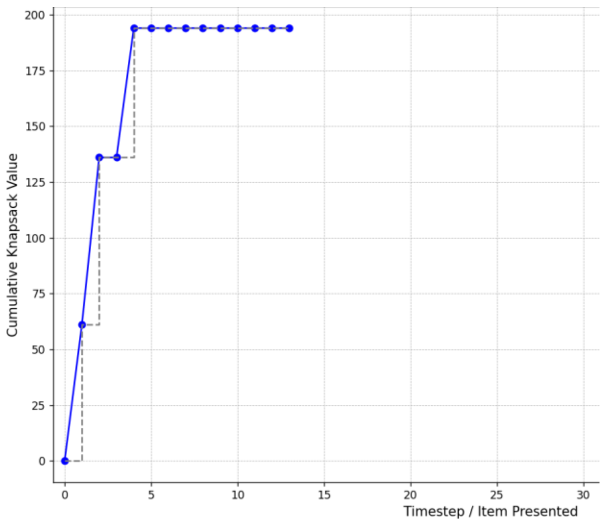
Seed 0: final value = 194, final weight = 153

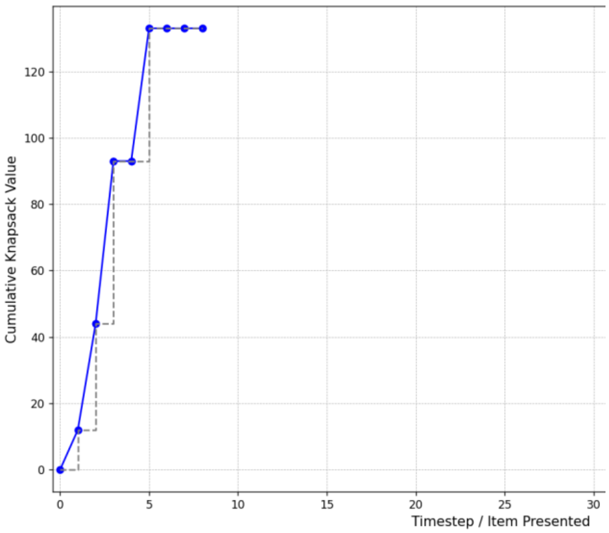
Seed 1: final value = 133, final weight = 142

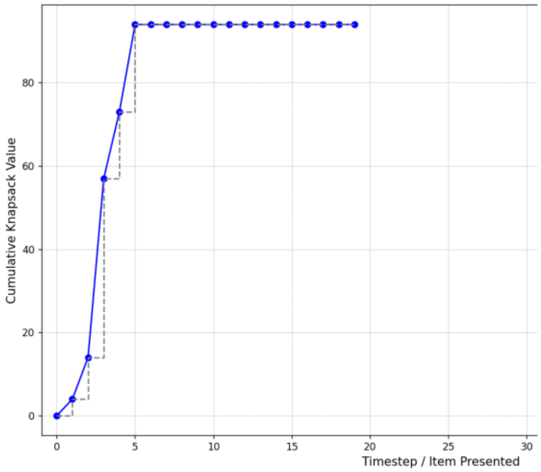
Seed 2: final value = 94, final weight = 184

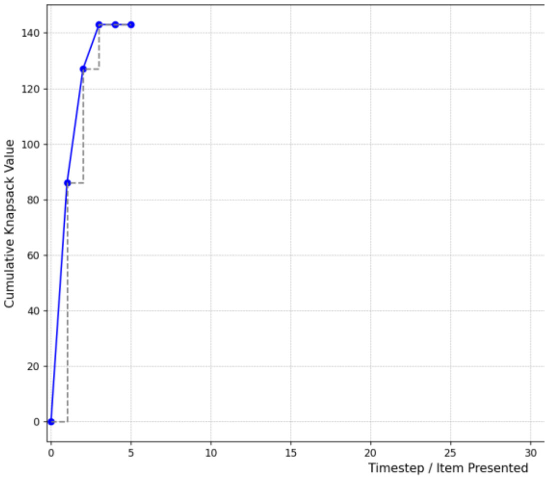
Seed 3: final value = 143, final weight = 153

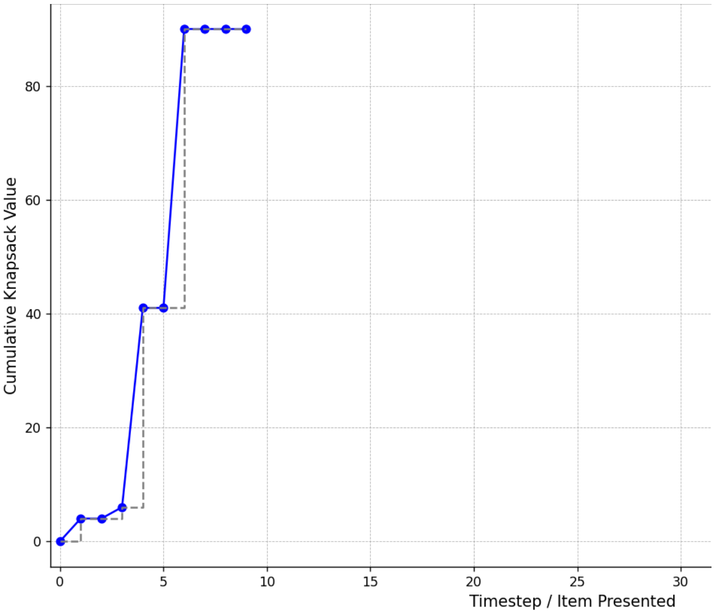
Seed 4: final value = 90, final weight = 145


200
175
150
125
100
75
50
25
0
Cumulative Knapsack Value
0
5
10
15
20
25
30
Timestep / Item Presented


120
100
80
60
40
20
0
Cumulative Knapsack Value
0
5
10
15
20
25
30
Timestep / Item Presented


80
60
40
20
0
Cumulative Knapsack Value
0
5
10
15
20
25
30
Timestep / Item Presented


140
120
100
80
60
40
20
0
Cumulative Knapsack Value
0
5
10
15
20
25
30
Timestep / Item Presented



Cumulative Knapsack Value
0
20
40
60
80
0
5
10
15
20
25
30
Timestep / Item Presented