# N-Model, Threshold & Greedy Policy

Guo Yuanxin

CUHK-Shenzhen

December 4, 2019

# N-model: System Description

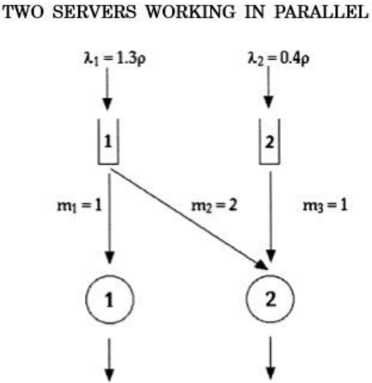



Figure: N-model

# N-model: System Description

- Arrival: Poisson process with rate $\lambda_i$
- $\lambda_1 = 1.3\rho, \ \lambda_1 = 0.4\rho$
- Service: Exponential with mean $m_i$
- $m_1 = 1, m_2 = 2, m_3 = 1$
- Goal: Minimize holding cost: $g(X_1, X_2) = 3X_1 + X_2$
- The only decision to be made is when server 2 is idle, whether to take a job of class 1 or class 2.

# Threshold Policy

- Class 1 job has a higher arrival rate, so it is more likely to be congested.
- Idea: Whenever jobs in buffer 1 exceeds a certain **threshold** $\alpha$ (or buffer 2 is empty), server 2 takes a class 1 job. Otherwise take a class 2 job.
- Problem becomes choosing the best threshold $\alpha$.

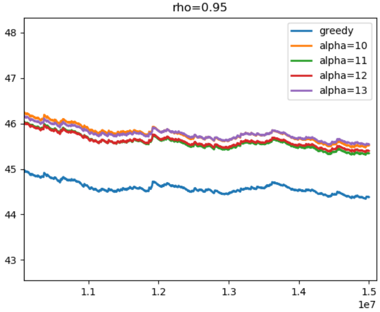
# Greedy Policy

- We can use **relative value iteration** to find a greedy optimal policy.
- The value function of regular value iteration diverges in this case, so we subtract the value function of a *reference state* at each iteration, which makes the iterates converge.

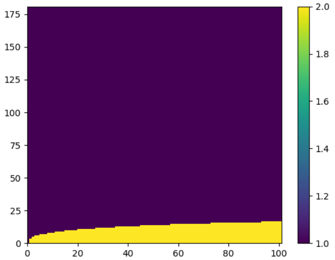
# Truncating the state space

- However, we have infinite state space $\mathcal{S} = \mathbb{Z}_+^2$. It is impossible for us to do RVI on it.
- We truncate the state space, discard the part that is unlikely to occur.
- When $\rho = 0.95$, we truncate the space to be $0 \leq X_1 \leq 180$, $0 \leq X_2 \leq 120$.
- Only 1% of the states are out of the truncated state space.
- When $\rho = 0.8$, we truncate the space to be $0 \leq X_1 \leq 45$, $0 \leq X_2 \leq 45$.
- Only 0.001% of the states are out of the truncated state space..

# Comparison of policies ($\rho = 0.95$)



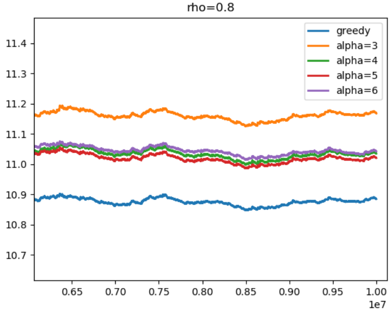

(a) Long-run average value function



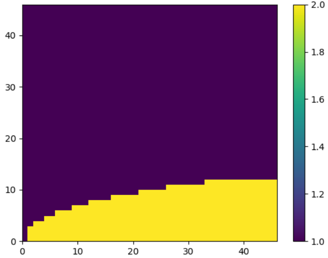

(b) Greedy Policy

Figure: $\rho = 0.95$

# Comparison of policies ($\rho = 0.8$)




(a) Long-run average value function




(b) Greedy Policy

Figure: $\rho = 0.8$

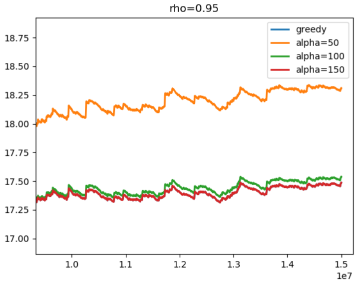
# Results

- The greedy policy does better than all threshold policies.
- The 'threshold' is increasing over $X_2$, which is in accordance to our intuition.

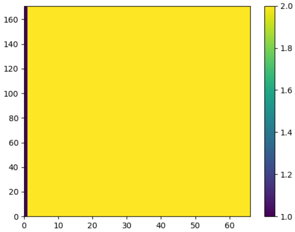
## Changing the holding cost

- We now consider the holding cost to be $g(X_1, X_2) = X_1 + X_2$.
- Since $m_2 > m_3$ while the holding cost for the two kinds of job are the same, it is always favorable to take a job from buffer 2 whenever it is not empty.
- The greedy policy thus reduces to the **last-buffer-first-serve** (**LBFS**) policy.

# Comparison of policies ($\rho = 0.95$)




(a) Long-run average value function

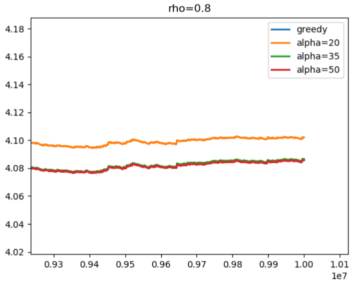



(b) Greedy Policy

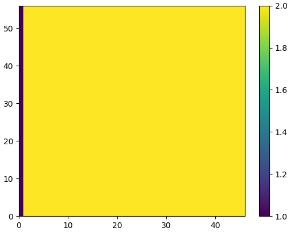
Figure: $\rho = 0.95$

- The blue line almost coincides with the red one in (a).

# Comparison of policies ($\rho = 0.8$)




(a) Long-run average value function




(b) Greedy Policy

Figure: $\rho = 0.8$

- The blue line almost coincides with the red one in (a).

# Sample generation does make a difference

- Using different random seeds may produce unwanted results. In the following example, it seems that threshold policy is better.

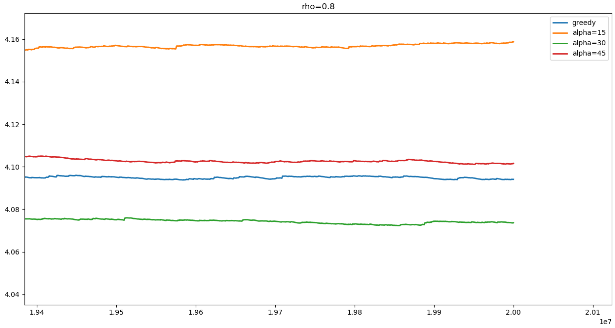



Figure: Using different seeds produces opposite result