DDPG for Optimal Execution of Portfolio Transaction

Zeya Yu

7385661781

1. Introduction

The execution of portfolio transactions are transactions that move a portfolio from a given starting composition to a specified final composition within a specified period of time. Our aim is to minimize a combination of volatility risk and transaction costs arising from permanent and temporary market impact during the execution of portfolio transactions.

Traditionally, this problem, also called optimal liquidation problem, can be solved by using the Almgren-Chriss market impact model. And it is usually solved by greedy algorithm like DP, which is inefficient. However, for the transaction of each day, what we need to consider are the amount of shares we left(state) and the amount of stocks we choose to sell(action), which means by giving each action with corresponding reward, this process can be considered as a MDP. Therefore, we can use RL algorithm to increase efficiency of the minimizing problem.

2. Problem Statement

i. States

The optimal liquidation problem entails that we sell all our shares within a given time frame. Therefore, our state must contain some information about the time remaining, or what is equivalent, the number trades remaining. We will use the latter to define our state vector.

Let $m_k = \frac{N_k}{N}$ denotes the number of traders remaining at time $\mathrm{t_k}$ normalized by the total number of traders.

Let $i_k = \frac{X_k}{X}$ denotes the number of shares remaining at time t_k normalized by the total number of shares.

We also need the information of historical stock prices, let's define them as $S_k, S_{k-1}, S_{k-2}, \ldots$.. By Almgren and Chriss model, the stock price follows:

$$S_k = S_{k-1} + \sigma\tau^{\frac{1}{2}}\xi_k - \tau g\left(\frac{n_k}{\tau}\right)$$

Almgren and Chriss model also defines the impacted price which means what you can actually get from the market in this transaction:

$$\tilde{S}_k = S_{k-1} - h(\frac{n_k}{\tau})$$

ii. Actions

Since the optimal liquidation problem only requires us to sell stocks, it is reasonable to define the action u_k to be the number of shares to sell at time t_k.

iii. Rewards

In Almgren and Chriss model, it is derived that an optimal execution strategy：

$$minimize\ implementation\ shortfall \qquad XS_0 - \sum n_k\tilde{S}_k$$

$$subject\ to \qquad V(x) \le V^*$$

Which is equivalent to minimizing the utility function

$$U_{util}(x) = \lambda_u V(x) + E(x).$$

Where:

$$E(x) = \sum_{k=1}^{N} \tau x_k g(v_k) + \sum_{k=1}^{N} n_k h(n_k/\tau)\mathrm{i}$$

$$V(x) = \sigma^2 \sum_{k=1}^{N} \tau x_k^2$$

Since the value of function V and E at each time can be calculated by Almgren and Chriss model, by defining a reasonable risk aversion λ, we can calculate the value of U_{util} at each time t_k. Thus the Reward function can be defined as the difference $R_t = U_t(x_t^*) - U_{t-1}(x_{t-1}^*)$, where x_t^* denotes the optimal trajectory at time t.

iv. RL model

Since the action space and policy space are both continuous in this problem, we can choose DDPG(Deep Deterministic Policy Gradient) algorithm. The pseudocode of this algorithm looks like this:

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

3. Related Literature.

One important paper is *Reinforcement Learning for Optimized Trade Execution*, written by Yuriy Nevmyvaka, Yi Feng and Michael Kearns in 2006. In this paper they combined the real-world orders and artificial orders generated by their strategies with an efficient RL algorithm carefully crafted to take advantage of the structural features of execution. Another important paper is *A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution*, written by Dieter Hendricks and Diane Wilcox in 2014. In this paper, they formulated the optimal trading problem as MDP and conducted a revised Q-learning update equation to solve it. *Double Deep Q-Learning for Optimal Execution* and *Practical Deep Reinforcement Learning Approach for Stock Trading* extended the optimal problem to DQN and DDPG. *Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis* introduced multi-agent deep reinforcement learning model to better captures high-level complexities.

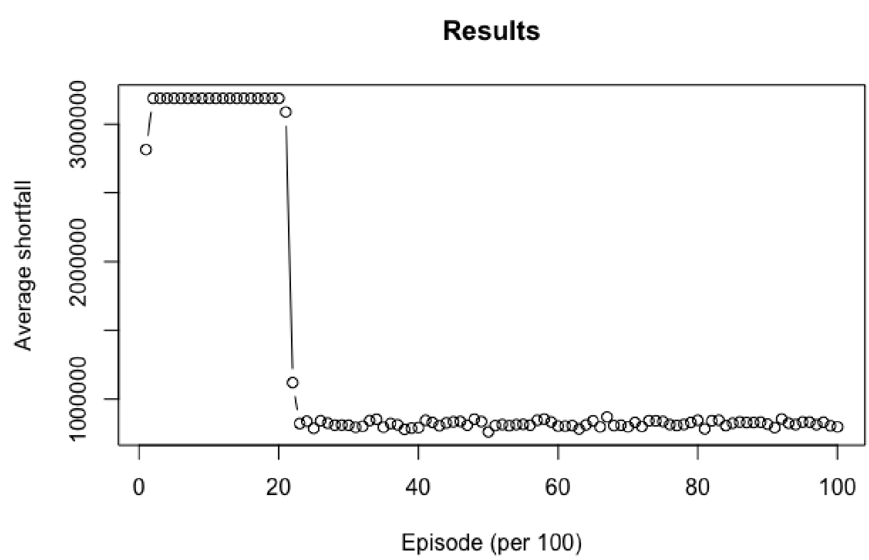
4. Project Results.

At first we need to set some basic financial and Almgren and Chriss model parameters:

Annual Volatility	10%
Bid-Ask Spread	1/8
Daily Volatility	0.6%
Daily Trading volume	4,000,000
Total number of shares to sell	1,000,000
Stock price per share	$30
Trader's risk aversion	1e-05
Number of days to sell all shares	2 months (60 days)
Number of Trades	60

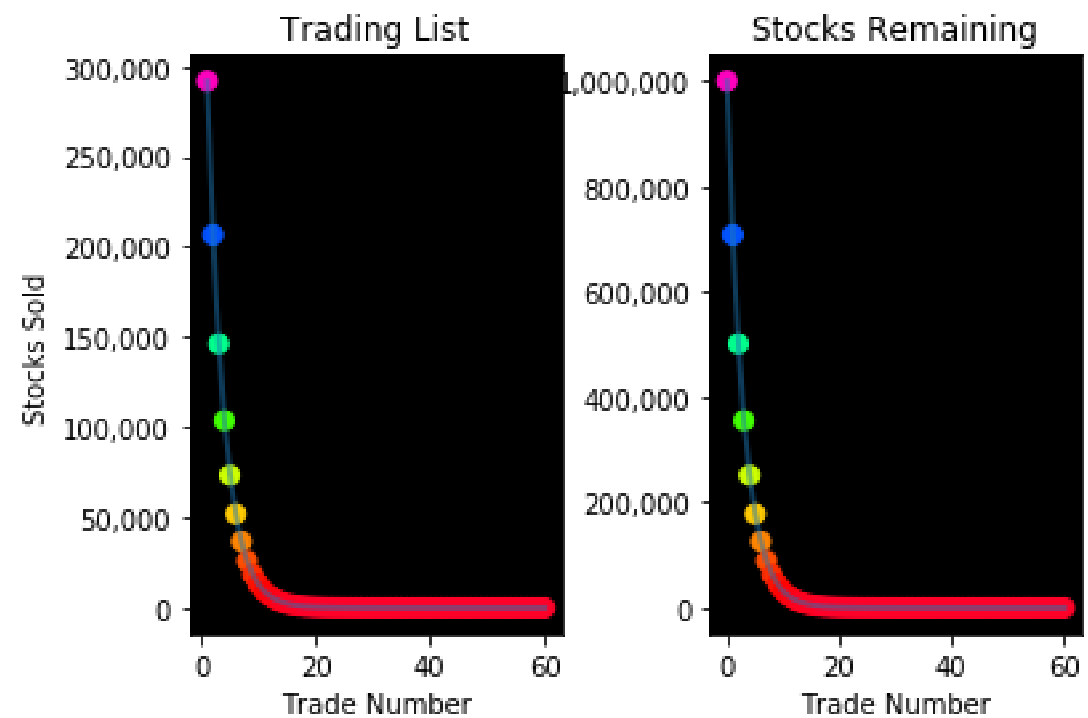
In DDPG algorithm, we set 2 neutral network in both actor optimizer and critic optimizer. We use the Almgren and Chriss model to calculate the transition vector and store it in experience replay memory.

Our target is to minimize the average implementation shortfall. The trend of average shortfall in every 100 episode looks like this:

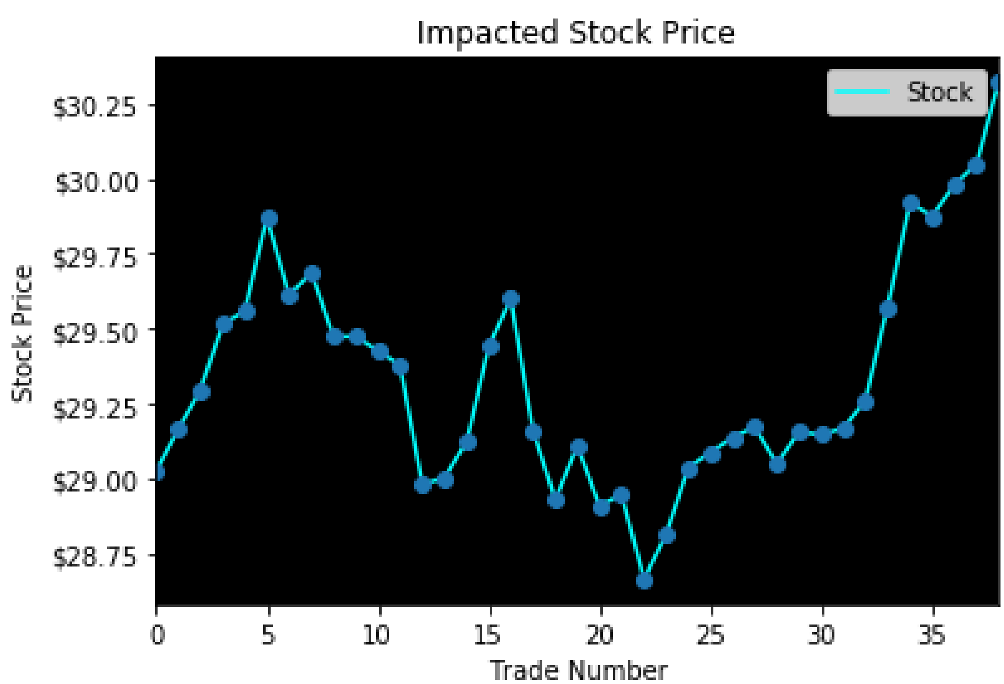


which means the DDPG converges around 2200 episode.

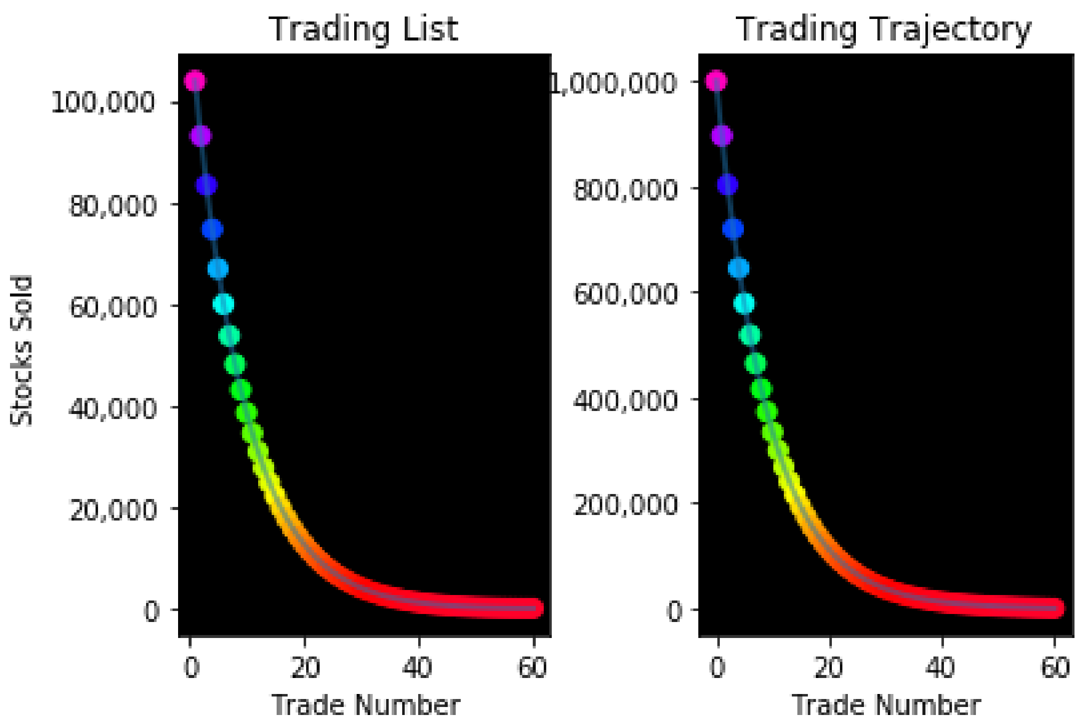
The next figure shows the trading list and stocks remaining after each transaction:

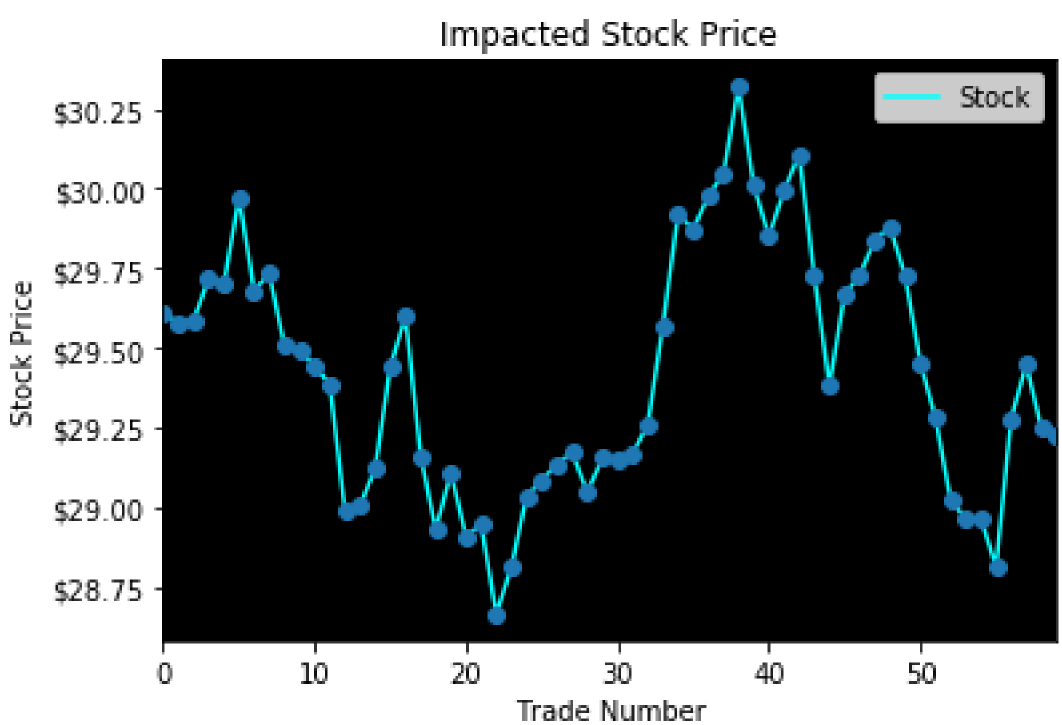


Actually we finished execution process in first 40 transaction, and in each transaction the impacted stock price is:



When we set the parameter Traders’ risk aversion as 1e-06, which means a lower risk execution process. The results have slight difference:





The results suggest that the trader who has a higher risk aversion will sell a larger percent of shares in current stocks remaining than the one who has a lower aversion. And the trader with a higher risk aversion will have cost fewer number of trades to finish the execution process. But the trader with lower risk aversion has a higher average impacted price, which means if you can afford a higher risk, you can earn more from the market. The experiment results also indicate that the optimal execution strategy has the method that the number of shares to sell at each transaction is monotone decreasing.

5. References

[1] Almgren, R. and N. Chriss (2000). Optimal execution of portfolio transactions. J. Risk 3 (2), 5–39.

[2] Robert Almgren, Neil Chriss, "Optimal execution of portfolio transactions" J. Risk, 3 (Winter 2000/2001) pp.5–39

[3] Nevmyvaka Y , Feng Y , Kearns M . [ACM Press the 23rd international conference - Pittsburgh, Pennsylvania (2006.06.25-2006.06.29)] Proceedings of the 23rd international conference on Machine learning, - ICML \"06 - Reinforcement learning for optimized trade execution[J]. 2006:673-680.

[4] Hendricks D, Wilcox D. A reinforcement learning extension to the Almgren-Chriss model for optimal trade execution[C]// Computational Intelligence for Financial Engineering & Economics. 2014.

[5] Bao W , Liu X Y . Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis[J]. 2019.