REINFORCEMENT LEARNING FOR OPTIMIZED TRADE EXECUTION

Authors: Yuriy Nevmyvaka, Yi Feng, and Michael Kearns
Presented: Saif Zabarah
Cs885 – University of Waterloo – Spring 2020

OPTIMIZED TRADE EXECUTION

Does not decide on what to invest on and when.

Instead, if you do decide to Buy/Sell

- How to execute the order:
  - Within a specific horizon.
  - While Maximizing revenue (SELL) earned or minimizing cost(BUY).

LIMIT ORDER MARKETS AND MICROSTRUCTURE

Modern financial Markets such as NASDAQ

Buyers and sellers not only choose

quantities they also choose their price.

Order book of a specific share

lists both sell and buy orders

INET home system stats help

NVDA

GET STOCK NVDA go

Aggregate by Price

LAST MATCH		TODAY'S ACTIVITY	
Price	27.2200	Orders	59,437
Time	16:00:41	Volume	1,336,147

BUY ORDERS		SELL ORDERS	
SHARES	PRICE	SHARES	PRICE
9	27.1900	713	27.2200
100	27.1800	1,000	27.2700
9	27.1800	640	27.2700
109	27.1300	500	27.4300
1,843	27.0400	500	27.4500
100	26.7500	20	27.8000
700	26.4900	700	27.8200
1,000	25.0000	1,000	27.9300
700	24.8000	400	28.0000
2,300	24.4800	75	28.0000

Figure 1. Snapshot of NVDA order book.

Nasdaq

RELATED WORK AND CONTRIBUTIONS

Related work:

- (Bertsimas and Lo, 1998), (Chan et al, 2001), (Tesauro and Bredin, 2002), and (Kim and Shelton, 2002)

Contributions:

- Demonstrate that RL approaches are well suited for optimized Execution
- Take advantage of the order book trade execution (Microstructures).
- Study of the value of a variety of market variables.
- An analysis of the policies learned by RL.

EXPERIMENTAL METHODOLOGY

Identification of state variables:

- Private and Market variables

The application of a customized RL Algorithm which exploits these variables.

Comparing this RL algorithm to a variety of natural baseline executions.

STATE BASED STRATEGY

There are two conventional ways sell/buy V shares in horizon H

- Sell/Buy using available market price
- Submit and leave

State-based strategy offers a good middle ground

- During the horizon H readjust your prices to maximize your capital
- Real-world traders are too busy to do that.

STATE VARIABLES

Each state is a vector of attributes.

Although it is a partially observable state. It will be considered fully observable.

They explore multiple state representations.

$$xm = \langle t,i,o1,o2,\ldots..,or \rangle$$

Private variables:

- Elapsed time t
- Remaining inventory i

Market variables:

- Using limit order books and recent activity of the stock. (o)

PRIVATE VARIABLES

How will i and t be represented.

We don't want a very large state space!

Therefore I and T must be chosen.

- These are the resolutions of i and t.

For example:

We have V = 10,000 shares and want to completely sell them in H = 2min.

I = 4 and T=4

Our remaining inventory can be represented in batches of 10,000 /4 = 2500 shares.

Our remaining time can be represented in 2min /4 = 30 sec increments.

If we are in t = 1 and i = 2 then 30sec have passed and 5000 shares left.

ACTIONS

An action a is:

- Withdrawing all unexecuted limit orders
- Changing limit order prices to
  - For selling:
    - ask = ask-a
  - For buying
    - Bid = bid + a
  - A = 0 : no change in price
  - Positive a means “crossing the spread”
  - Negative means “deeper within out books”

REWARDS

Each action may produce an immediate reward.

- Selling: outflow of cash
- Buy :inflow of cash

When H is reached everything left must be executed.

In all cases we define the trading cost as the underperformance in comparison to the idealized price.

the idealized price: mid-spread price at the beginning of the episode

- (Ask + bid) /2

ALGORITHM (MARKOVIAN ASSUMPTION)

"Approximately" Markovian nature of trade executions:

- An action at any state in time is independent of any previous actions
- Meaning when t = T(no time remaining.) Independent of all

This way we can move backwards in time:

- We assign optimal actions for all states with t=T
- We then move backwards in time with enough information to assign optimal action for all states in t = T-1
- And so on…

ALGORITHM (MARKET VARIABLES ASSUMPTION)

Our actions will not affect market variables

Makes the problem simpler

Which can also be exploited to reduce overfitting.

FINDING THE OPTIMAL ACTION

Similar to Q-learning:

- In every state encountered we try all actions.
- Update the cost associated to each action
- Following the optimal strategy

Each action results in an immediate payout and a new state

Since we move backwards the new state has already been optimized.

Therefore our cost update rule:

$$c(x,a) = n/(n+1)\ c(x, a) + 1/(n+1)\ [c_{im}(x,a) + \text{argmax}\ c(y,p)]$$

y → new state, p → action taken in y

x → current state

C_{im} → immediate 1 step cost

ALGORITHM

```
Optimal_strategy (V, H, T, I, L)
    For t = T to 0
        While (not end of data)
            Transform (order book)  o1 … oR
            For i = 0 to I
              For a = 0 to L
                Set x = {t, i, o1 … oR}
                Simulate transition x → y
                Calculate cim(x, a)
                Look up argmax c(y, p)
                Update c(<t,v,o1 …or>, a)
    Select the highest-payout action argmax c(y, p)
     in every state y to output optimal policy
```

DATA SET

They used historical records from INET

Accounts for a significant volume of NASDAQ stock trading

They developed a simulator:

- Combines real orders with our artificial orders

Training/testing split 12/6 months respectively

EXPERIMENTAL METHODOLOGY

Investigated every combination of the following:

- Stocks: AMZN, NVDA, and QCOM
- V = 5000 or 10,000 shares
- H = 2 or 8 minutes

I and T resolution were generally kept small

- To keep a reduced state space

Market Variables summarized information from the order books

- Reduced their resolution.
- For example Executed Market Volume is reduced to high, medium, and low

Partitioned training data into episodes

- If data is 1 years worth and H is chosen to be 2 min then it was partitioned into 45,000 episodes.

Policies learned were compared with several baseline strategies

EXPECTATIONS (TO KEEP IN MIND)

With all other factors kept constant the following will cause an increase in cost:

- Stock Liquidity
  - Less liquid means more expensive to trade.
  - NVDA is the least liquid stock.
- Larger orders are more costly
- Smaller remaining time results in higher costs

EVALUATION

Evaluating model using different state space configuration :

- Using private variables only
- Using private variables + market variables

EVALUATION

With only the private variables

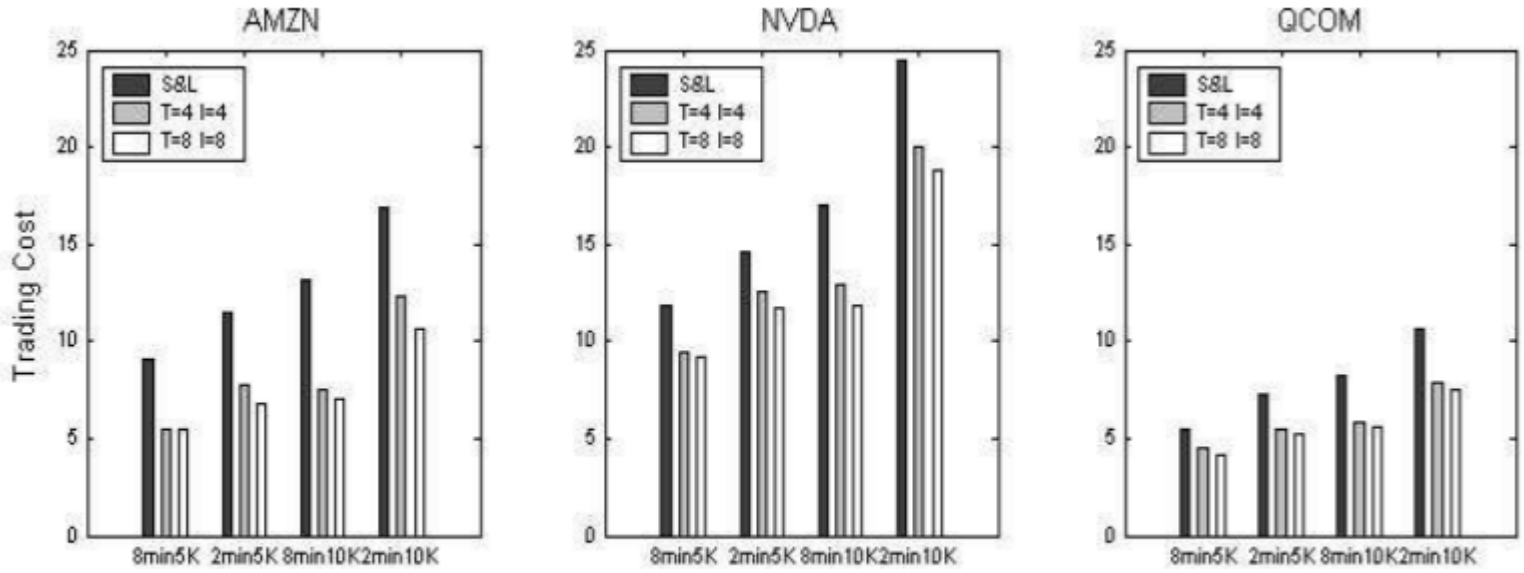

Figure 3. Expected cost under S&L and RL: adding private variables T and I decreases costs

EVALUATION

With only the private variables

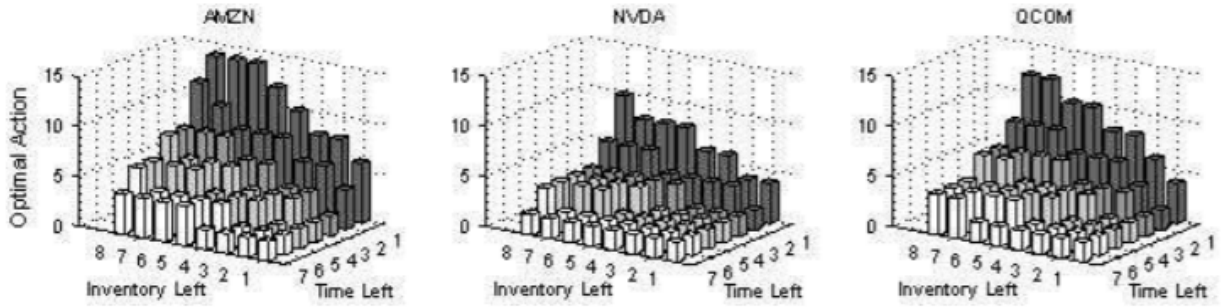

Figure 4. Visualization of learned policies: place aggressive orders as time runs out, significant inventory remains

EVALUATION

With only the private variables

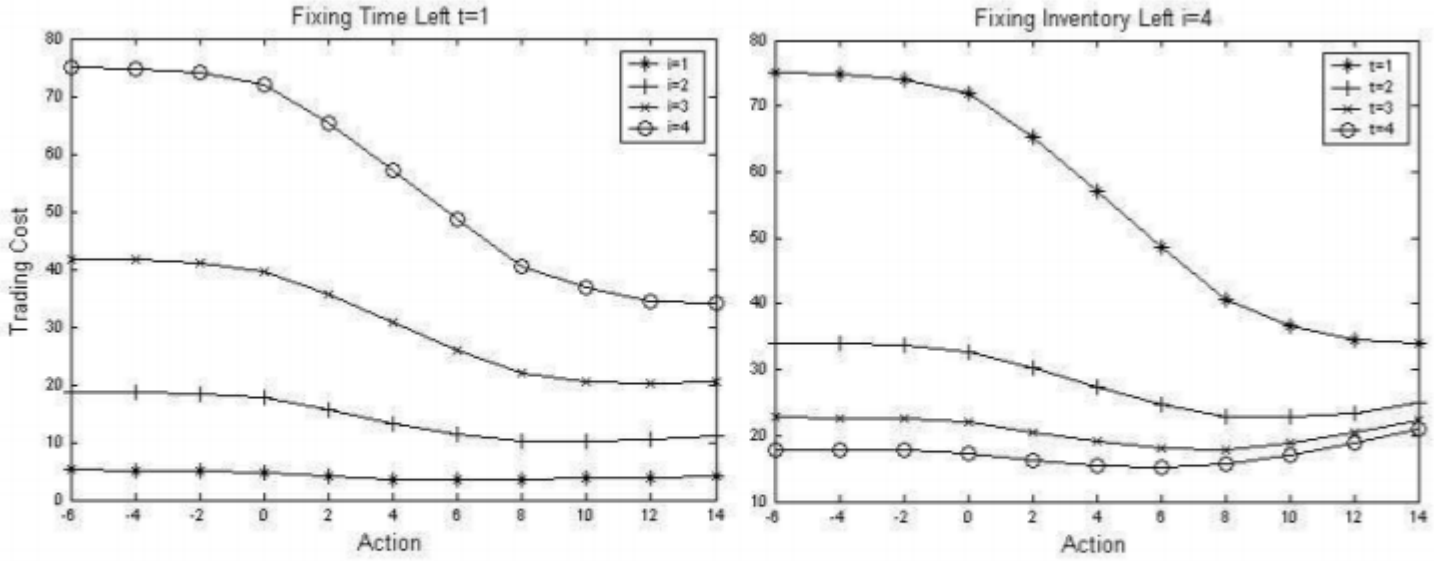

Figure 5. Q-values: curves change with inventory and time (AMZN, T=4, I=4)

EVALUATION

With both private variables and different combination of market variable

Bid-Ask Spread	7.97%
Bid-Ask Volume Misbalance	0.13%
Spread + Immediate Cost	8.69%
Immediate Market Order Cost	4.26%
Signed Transaction Volume	2.81%
Spread+ImmCost+Signed Vol	12.85%

Table 1. Additional trading cost reduction when introducing market variables

EVALUATION

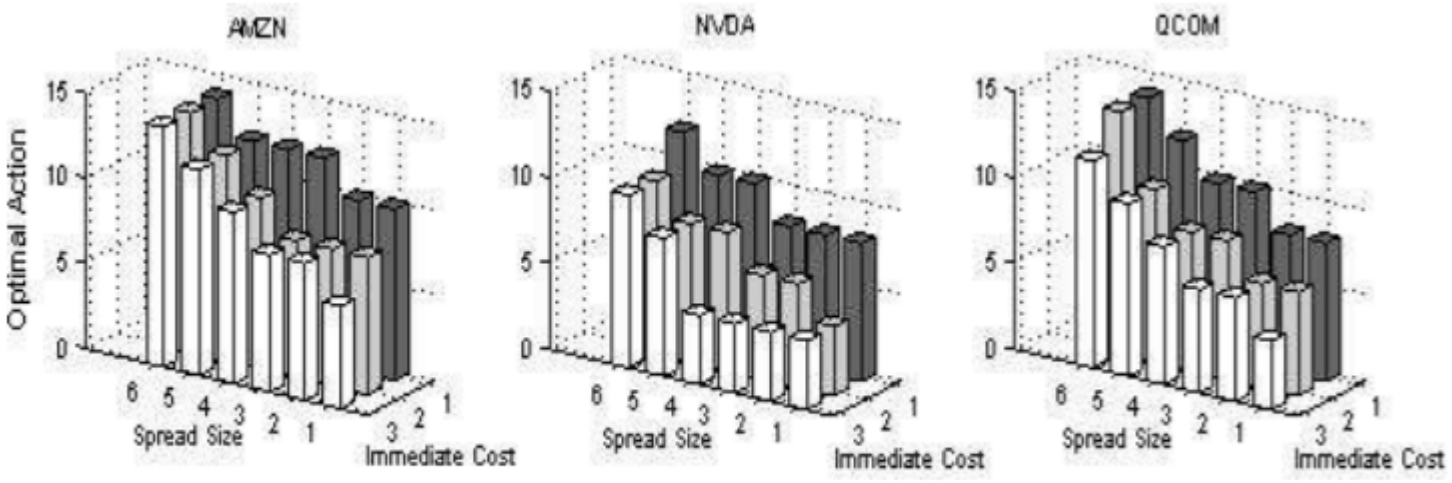

Figure 6. Large spreads and small market order costs induce aggressive actions

EVALUATION

With both the private variables and market variables

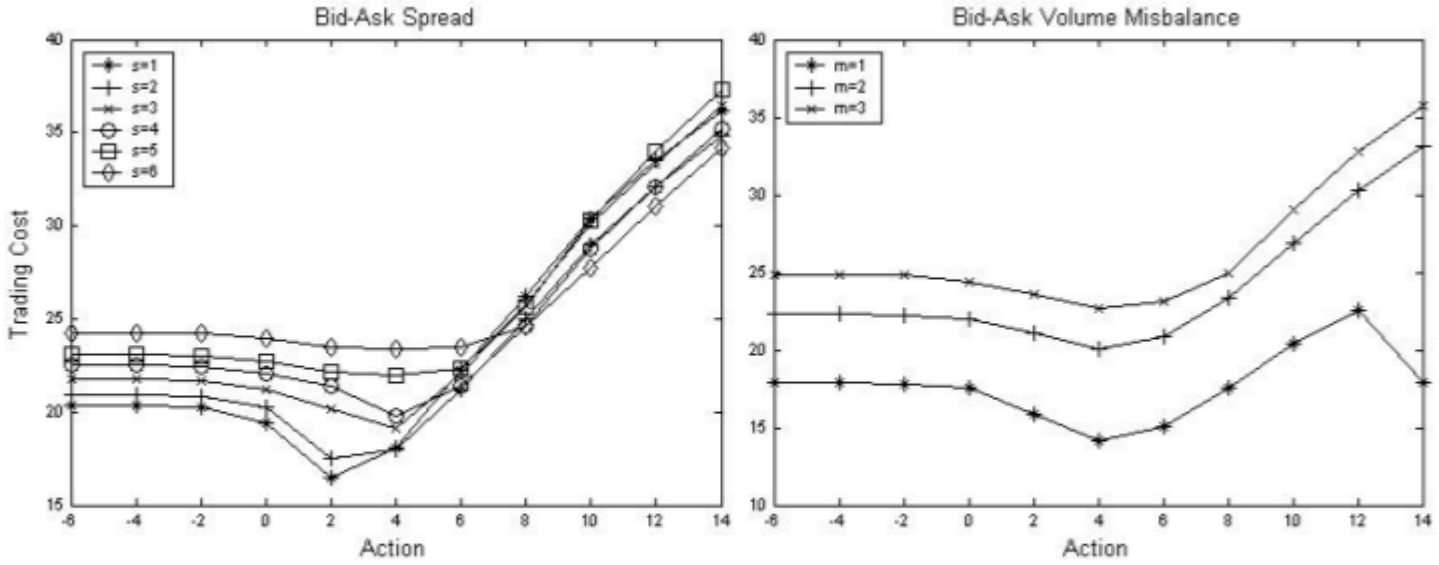

Figure 7. Q-values: cost predictability may not affect the choice of optimal actions

CONCLUSION AND FUTURE WORK

First large-scale application of RL in optimized trade execution

Used the newly available limit order markets micro-structures

Provided improvements of up to 50% or more in comparison to S&L.

Adapter this work to other precisely-defined finance problems .

REFERENCES

Bertsimas, D., A, Lo, A., Optimal Control of Execution Costs. Journal of Financial Markets 1, 1-50, 1998.

Chan, N., Shelton, C., Poggio, T., An Electronic MarketMaker. AI Memo, MIT, 2001.

Tesauro, G., and Bredin, J., Strategic Sequential Bidding in Auctions Using Dynamic Programming. Proceedings of AAMAS-02.

Kim, A., Shelton, C., Modeling Stock Order Flows and Learning Market-Making from Data. AI Memo 2002-009, MIT, 2002.