# A Reinforcement Learning Extension to the Almgren-Chriss Model for Optimal Trade Execution

Diamor Marke
BMLL Technologies Ltd, 5 Fleet Place, London, UK, EC4M 7RD

## Transaction Costs vs Volatility

When considering the optimal execution of portfolio transactions, one may aim to minimise the average transaction cost and the volatility of transaction costs. However, these goals oppose each other; one can minimise volatility by executing all orders immediately, but this maximises expected cost because large orders go though several levels of the limit order book. One can also minimise average transaction cost by executing many equally sized small orders, but this maximises volatility because of the uncertainty of price movements over time.

## Data Pipeline

Vodafone, Tesco and Barclays were selected to reproduce the results, as they are all large cap stocks. 5 levels of order book data were collected of every 5 mins from Jan 2018 to Mar 2018. This includes bid-ask spread, volume and mid-market price data.

## Optimal Trade Execution

Almgren and Chriss specify a model for optimal trade execution [1] which aims to minimise $E(x) + \lambda Var(x)$, a combination of expected transaction cost and volatility, weighted based on a trader's risk aversion, $\lambda$. Here $x$ is implementation shortfall ($IS$), a measure of the difference between transaction cost and reference cost as defined by Perold in [3].

## The Almgren-Chriss Model

The Almgren Chriss model suggests that after execution $j$, the recommended amount of the total inventory $X$ to be holding is given by:

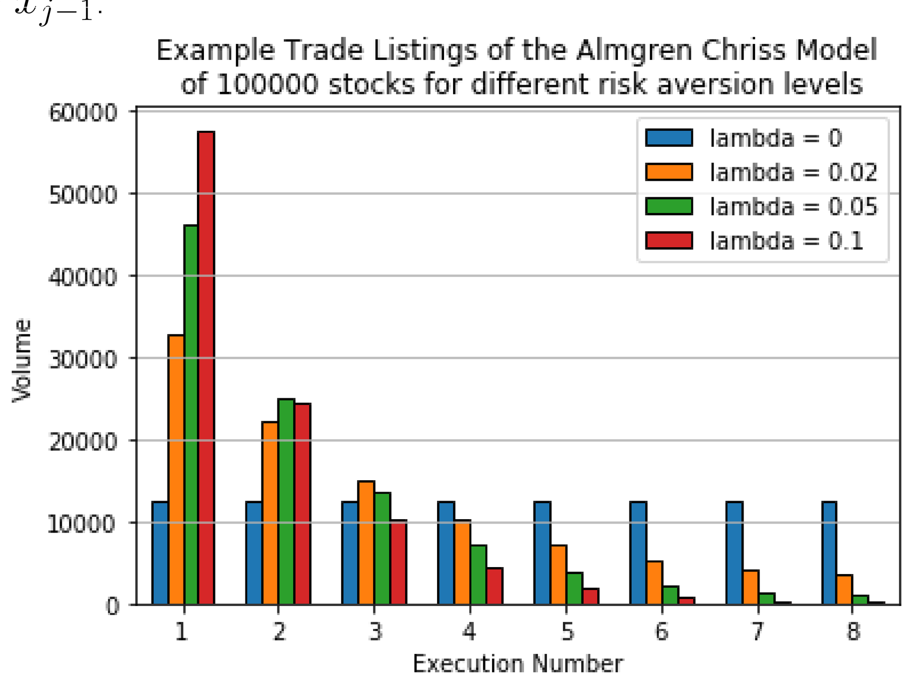
$$x_j = \frac{\sinh(\kappa(T - t_j))}{\sinh(\kappa T)} X,$$

where $t_j$ is the time of execution $j$, and $\kappa$ is given by:

$$\kappa = \frac{1}{\tau}\cosh^{-1}\left(\frac{\tau^2}{2}\tilde{\kappa}^2 + 1\right), \quad \tilde{\kappa}^2 = \frac{\lambda\sigma^2}{\eta(1 - \frac{\rho\tau}{2\eta})}, \quad t_j = \tau j,$$

where $\sigma$ is volatility, $\eta$ is the temporary price impact parameter, $\rho$ is the permanent price impact parameter, $\tau$ is the length of discrete time between executions, and $T$ is the time horizon of liquidation.

The recommended volume to sell at time $t_j$ is given by $n_j = x_j - x_{j-1}$.




## Testing the Almgren-Chriss Model

To test the Almgren-Chriss model, we simulated the buying of a number of stocks over a fixed time period based on recommendations by the model. We measure performance by:

$$IS = \frac{\text{ref} - \text{exec}}{\text{ref}},$$

where ref is reference cost and exec is execution cost. By this measure, a greater number is better in our tests, since we are buying stocks.

The following summarises the parameters and assumptions used for the results that follow:

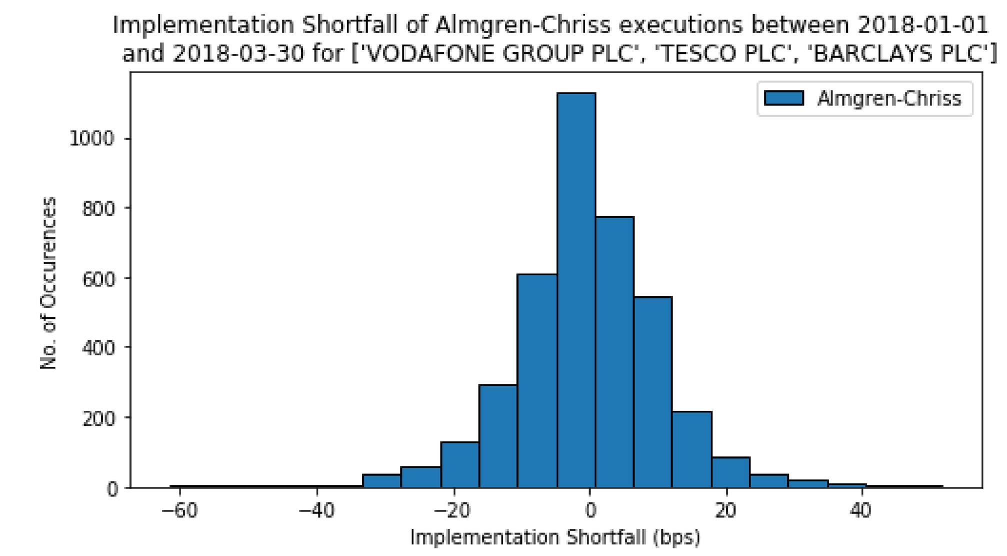
### Parameters

- $\lambda : 0.01, \tau : 5\text{min}, \eta : 0.8, \rho : 0.$
- $X$ : 20000.
- $T$ : 4(20min), 8(40min), 12(60min).
- Starting Hour: 9, 10, 11, 12, 13, 14, 15.
- Side: BUY.

### Assumptions

- Max 25% participation rate in order book.
- Our trading activity doesn't affect the state of the market.

## Almgren Chriss Results




## Reinforcement Learning Model

It is advantageous to increase trading activity when the bid-ask spread is tight because the market is more liquid. Likewise it is beneficial to increase trading activity when there is more volume on the market because price impact is lower. The Almgren-Chriss model does not take these factors into account, so Hendricks and Wilcox proposed an extension to the Almgren-Chriss model that does this using Q-learning [2].

The model takes the recommended volume from the Almgren-Chriss model, and executes a modified proportion of it based on the volume and spread state of the market, and the remaining number of executions. This proportion ranges from 0x to 2x the recommended volume. Subsequent execution volumes are adjusted accordingly. Any inventory left after the trading horizon is executed as a market order.

The model is trained on the first $1\frac{1}{2}$ months of data, and tested against the Almgren-Chriss model on the latter $1\frac{1}{2}$ months. The discount factor, $\gamma$, used is 1. The volume and spread states are based on the percentile of volume and spread. The immediate reward, $R(x, a)$, used for each action $a$ at state $x$ is the $IS$ of the simulated trade.

## Training Algorithm

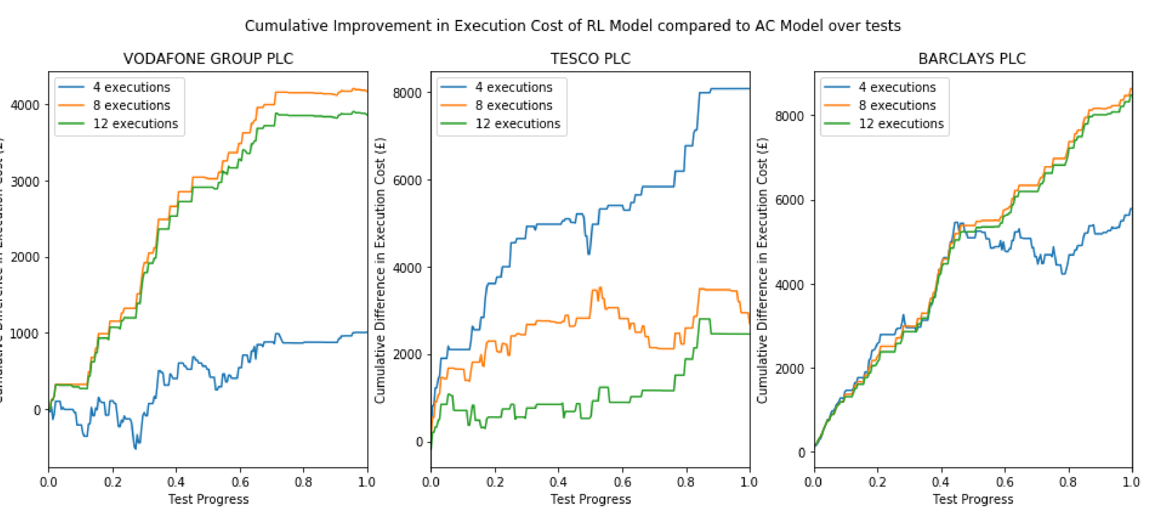
| Var. | Description |
|---|---|
| $A$ | No. of actions |
| $I$ | No. of inventory states |
| $x$ | State tuple: (exec. no., remaining inv, spread state, volume state) |
| $R(x, a)$ | Immediate reward for performing action $a$ at state $x$. |
| $Q(x, a)$ | Long term reward function. |
| $\gamma$ | Discount factor. |
| $\alpha$ | Learning rate. |

**Algorithm 1:** Training the Q-matrix

**Data:** Market Data
**Result:** Trained Q-matrix of long-term rewards
**for** *Epoch* $1$ *to* $N$ **do**
  **for** *Episode* $1$ *to* $E$ **do**
    Record reference price at $t = 0$.
    **for** $t = T$ *to* $1$ **do**
      **for** $i = 1$ *to* $I$ **do**
        Calculate spread state $s$ and volume state $v$ at current execution.
        **for** $a = 1$ *to* $A$ **do**
          $x \leftarrow$ (t, i, s, v)
          Determine the action volume of action $a$.
          Calculate $IS$ for trade, $R(x, a)$, and next state, $y$.
          $Q(x, a) \leftarrow Q(x, a) + \alpha(R(x, a) + \gamma \max_p Q(y, p) - Q(x, a))$
        **end**
      **end**
    **end**
  **end**
**end**

## Reinforcement Learning Results

The reinforcement learning model offers a statistically significant improvement over the Almgren Chriss model on 3900 training examples in a range of market conditions.

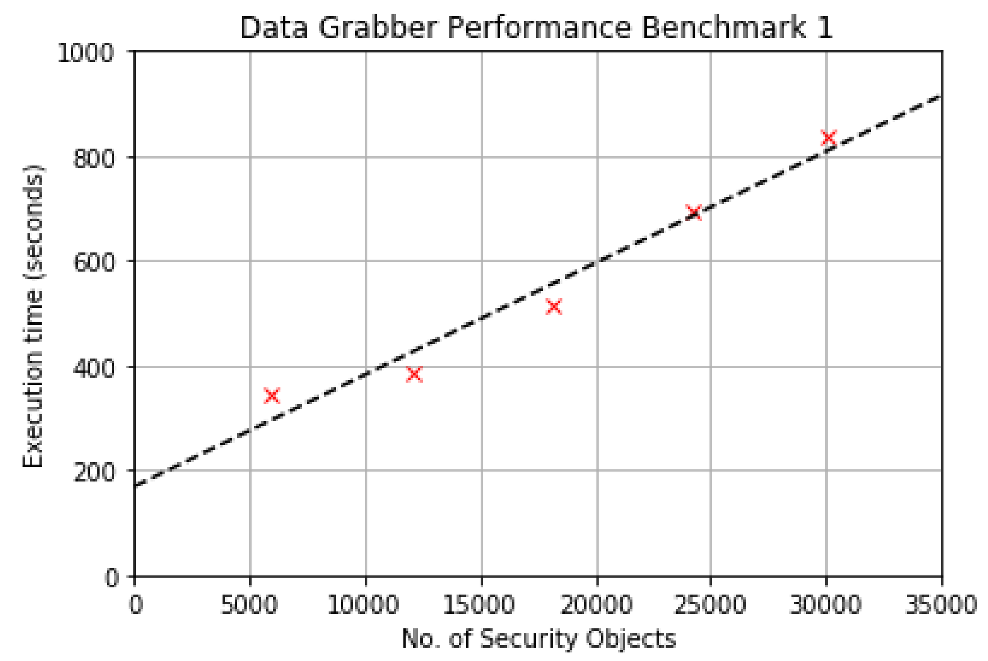



The charts above show that the reinforcement learning model reliably outperforms the Almgren-Chriss model and offers practical benefits, consistenly improving transaction costs when applied.

## Scaling Out Results

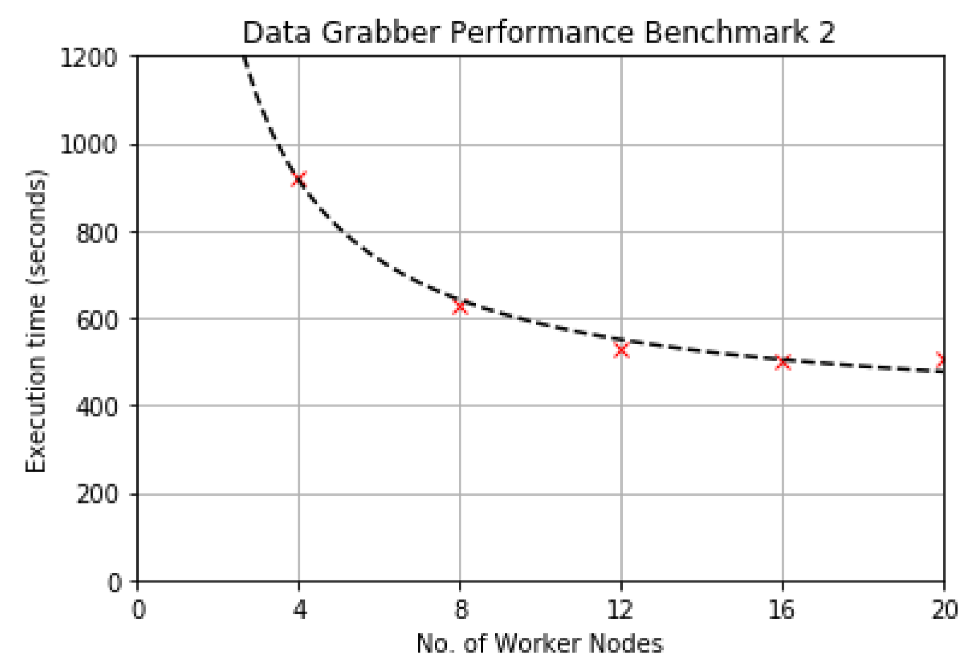
The reinforcement learning model outlined thus far has been extended to make use of make use of Amazon EMR clusters available via the BMLL platform. We will now demonstrate the benefits of this extension.

We first demonstrate how execution time scales with data volume given a constant cluster size. For this, we ran a script that grabs L1 and L2 dataframes on 3 to 15 months of L1 and L2 dataframes from the FTSE 100 constituents.




It is clear that in this test, execution time is proportional to the number of security objects, as expected.

We now demonstrate how execution time scales with the number of cluster worker nodes, given a constant volume of data to be processed. For this, we ran the data grabbing script on 12 months of historic data from the FTSE 100 constituents. We ran the script on Amazon EMR clusters of sizes between 4 and 20 worker nodes.




In this test, execution time is inversely proportional to the number of worker nodes, as expected, however plateaus as cluster size increases due to the computation required to distribute work across the clusters.

## Conclusion & Future Work

- Results from [2] are reproducable and give consistent results.
- Incorporate volatility into reward function, as it currently only optimises for $IS$.

## References

[1] R. Almgren, N. Chriss. *Optimal Execution of Portfolio Transactions*. Journal of Risk, 3, pp. 5-40, 2000.

[2] D. Hendricks, D. Wilcox. *A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution*. 2014 IEEE Computational Intelligence for Financial Engineering and Economics conference, 2014.

[3] A. Perold *The implementation shortfall: Paper versus reality*. The Journal of Portfolio Management Spring 1988.