# Simple Execution Algo Testing Framework

## *Background*

The problem of optimal execution of large orders over a finite interval of time is not new. Interested readers may review the risk-adjusted cost minimization approach introduced by Almgren and Chriss (2001).

Generally speaking, the trader faces the problem of optimal execution of large orders because of limited supply-demand liquidity in the market(s). If the order size is small enough there is no such a problem and the whole volume could be executed immediately.

One of the approaches used by traders is to split the original order into a sequence of orders of smaller sizes and execute them within a required time interval.

Assume the following:

(1) Midpoint between best bid and best ask is a good approximation of the book value of the asset:

$S_0 = \frac{BestBid+BestAsk}{2}$

(2) The trading size of large order is X

(3) The execution size of each partition is $x_i$ and fill price is $s_i$ such that:

$X = \sum_i^{\mathrm{N}} \mathrm{x_i}$ – The aggregate trade size

$\sum_i^{\mathrm{N}} \mathrm{x_i} \cdot \mathrm{s_i}$ - The aggregate trade cost

Then the total cost of trading is the difference $X \cdot S_0 - \sum_i^{\mathrm{N}} \mathrm{x_i} \cdot \mathrm{s_i}$ between the initial book value of the asset and aggregate trade cost. This is what is known as the **implementation shortfall (IS)**:

$IS = X \cdot S_0 - \sum_i^{\mathrm{N}} \mathrm{x_i} \cdot \mathrm{s_i}$ (*)

**IS** is a random variable which could be characterized by expected shortfall $E(X)$ and shortfall variance - $V(X)$.

So far we have two parameters we can use to compare execution algos. One is the expected cost associated with an execution algorithm and the other one is the risk of execution associated with this algorithm.

*For sell side orders

## *Market Order Fill Price Assessment*

If the user has access to historical market depth data on the given trading instrument – there is no such problem as expected order fill price could be directly derived from that data for the order of any size.

If historical market depth data are not available, then the researcher has to model the dependency of the market order fill price upon the order size in another manner. Generally speaking this is a very

complex problem as markets often provide hidden liquidity not directly represented even in the limit order book (LOB) data.

For our illustrative purposes we will use the following simple, but a little naïve approach:

(1) We will calculate a moving average trade size $\overline{TS}$ for each instrument using a rolling window over past 15 minutes. Using DCS as the research platform, DCS extended bars include the total volume and total number of trades for each bar. By aggregating both variables within a rolling time frame and dividing first one on the second one we get an estimate of the moving average trade size.

(2) We will assume the supply demand liquidity has the following form:

| Ask Depth | Ask Size | Bid Size | Bid Depth |
|---|---|---|---|
| 2 | Very Large | | |
| 1 | 4 $\overline{TS}$ | | |
| 0 | 2 $\overline{TS}$ | | |
| | | 2 $\overline{TS}$ | 0 |
| | | 4 $\overline{TS}$ | 1 |
| | | Very Large | 2 |

## *Execution Algorithms to Compare*

We selected the following algorithms to compare on DCS historical data:

(1) Minimal Variance;
(2) Minimum Impact; and
(3) Limit to Market hybrid algorithm.

### Minimum Variance

This is one type of extreme execution plan when the whole order size X is sent for execution by a market order immediately:

$$x_1 = X, x_2 = 0, \dots, x_N = 0$$

This plan has the smallest possible variance of trading cost but the trader may get a price hit when sending large volume to execute by market.

### Minimum Impact

This is also a commons execution plan when the order is executed at a constant rate over the given time interval. The original order size is divided on N equal parts and each part is sent for execution by market order at the beginning of particular sub-interval:

$$x_i = \frac{X}{N}, i = 1, \dots, N$$

This plan minimizes a total expected execution cost (shortfall) but this comes at the effect of a greater uncertainty in the variance of the cost. This algorithm is a basic kind of TWAP algorithmic order popular among traders.

### Limit-To-Market

This is another common execution plan which attempts to benefit by adding liquidity first.

The original order size is divided on N equal parts and each part is sent for execution by limit order at the beginning of particular sub-interval. The limit order offset (how deeply the order is placed in the book from the top of the book) is a parameter of the execution algo. At the end of the interval, there are two possible scenarios.

- The part size is filled completely. Nothing else to do.
- The part size is filled partially or not filled at all. The action is to cancel active limit order and to enforce execution of the remaining size by market order.

## *Experiment Setup*

The setup requires DCS data subscription with *QuantOfficeDCS Edition* installed on your machine.

The trading universe comprises of US Equities but it could be run on future contracts as well.

The data back-testing interval is 3+ years.

During the experiment for each trading day we decide on *trading direction* first (individually per instrument) which is set either to BUY or SELL.

During the day we generate a sequence of trading signals to buy or sell as per the day's *trading direction*. Each signal requires us to buy or sell a desired $ volume within a 15 minutes timeframe. Each trading signal is executed in parallel by all the execution algo(s) described above.

The implementation shortfall statistics are collected for all execution algo(s).

## *Experiment Results*

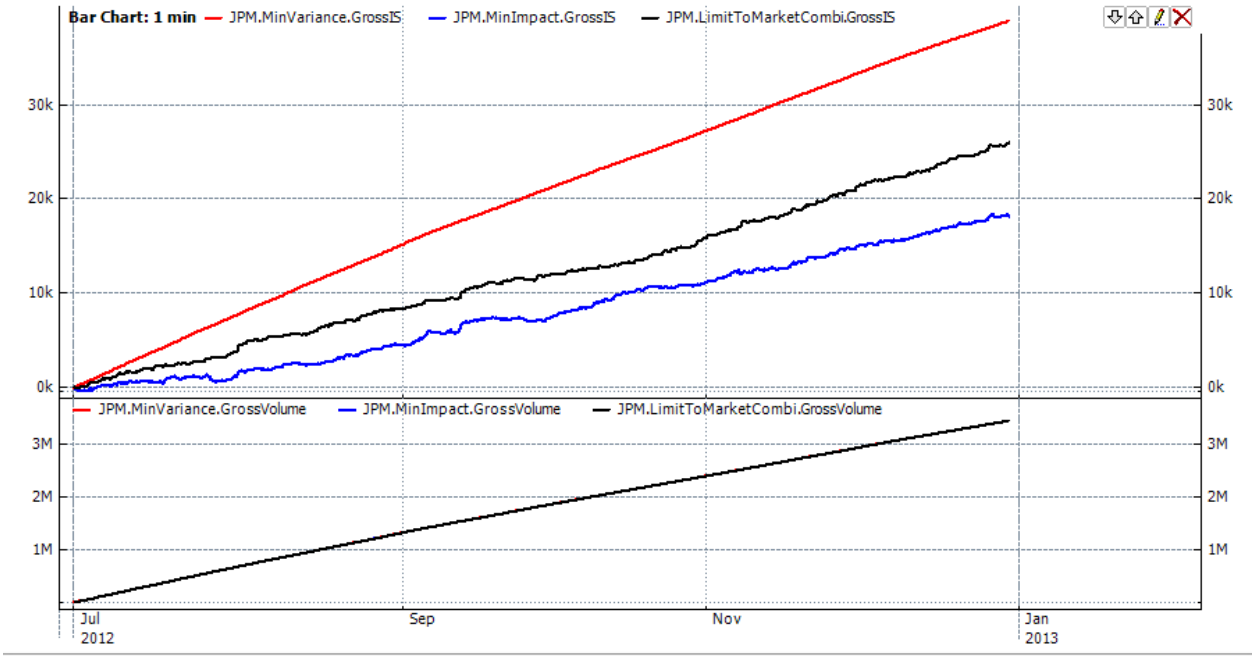
The results of experiments for JP Morgan Chase & Co (ticker JPM) and Well-Fargo (WFT) are represented in a chart format (cumulative Implementation Shortfall for each algo) and in report format.

The back-testing interval is set from July 1, 2012 to December 31, 2012.

### Cumulative Implementation Shortfall Chart for JP Morgan Chase & Co (JPM)

The results show that with our market liquidity assumptions the Minimum Impact algo provides the best execution cost and the Minimum Variance algo provides the worst execution cost.

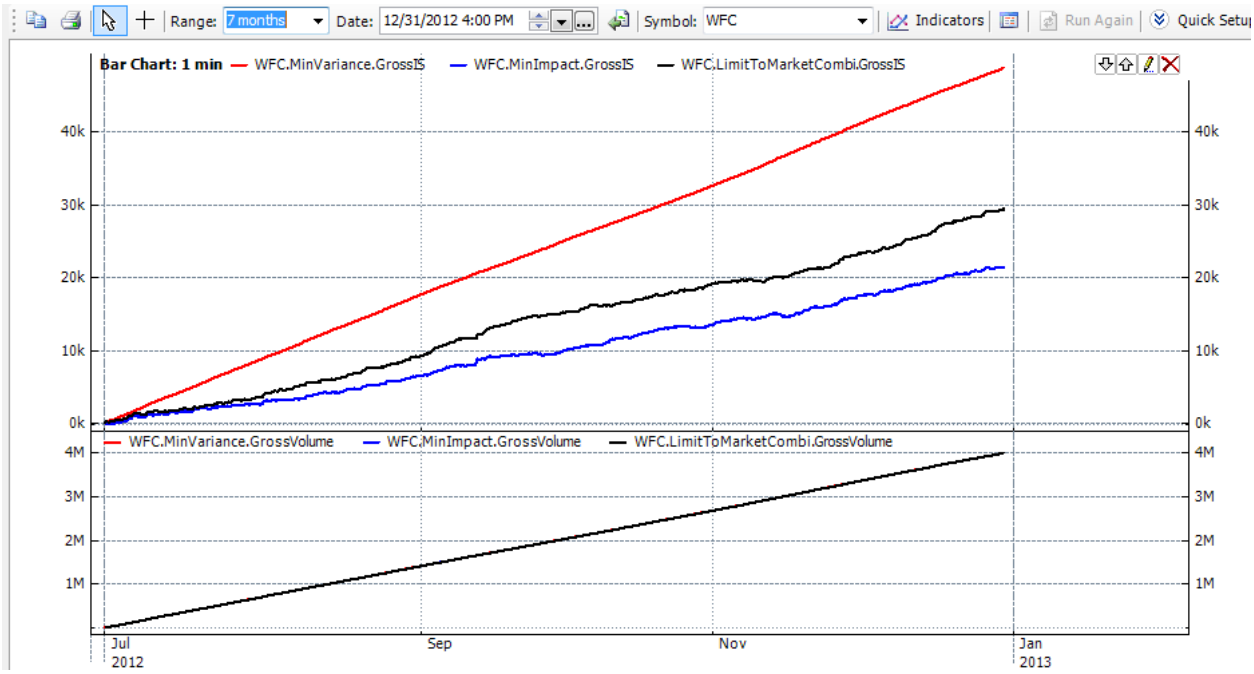
The execution cost by Limit-to-Market execution algo is in-between those two.




## Cumulative Implementation Shortfall Chart for Well Fargo (WFC)

The results for this stock show that with our market liquidity assumptions the Minimum Impact algo prvides the best execution cost and the Minimum Variance algo provides the worst execution cost.

The execution by Limit-to-Market ecxecution algo is in-between those two.




## Average Execution Cost and Variance report

The following table contains aggregate statistics on execution cost by all the execution algos mentioned above.

The table shows that during our back-testing, we had 8,151 trading signals for each ticker, each of which has been filled by all three algos.

The total shares turnover is presented in Qtty(Sum) column;

The cumulative Implementation Shortfall for each algo is presented in the column Shortfall(Sum);

The average shortfall per signal is presented in the column Shortfall (Average);

The standard deviation of the shortfalls is presented in the column Shortfall (STDEV).

| Group Name | Qtty (Sum) | Shortfall (Sum) | Shortfall (STDEV) | Shortfall (Average) |
|---|---|---|---|---|
| Symbol:JPM (8,151) | 10363872 | 83071.12 | 36.14 | 10.19 |
| executionAlgo:LimitToMarketCombi (2,717) | 3454624 | 25984.50 | 42.64 | 9.56 |
| executionAlgo:MinImpact (2,717) | 3454624 | 18160.90 | 45.48 | 6.68 |
| executionAlgo:MinVariance (2,717) | 3454624 | 38925.72 | 1.73 | 14.33 |
| Symbol:WFC (8,151) | 12011931 | 99728.04 | 32.83 | 12.24 |
| executionAlgo:LimitToMarketCombi (2,717) | 4003977 | 29486.91 | 41.23 | 10.85 |
| executionAlgo:MinImpact (2,717) | 4003977 | 21377.00 | 38.43 | 7.87 |
| executionAlgo:MinVariance (2,717) | 4003977 | 48864.13 | 1.98 | 17.98 |

## *On-Line Resources*

The following resource is related to the material presented in here:

(1) Tutorial and QuantOffice strategy source code:

http://dcs.developer.deltixlab.com/strategies/?category=45

## *References*

Robert Almgren and Neil Chriss. Optimal Execution of Porfolio Transactions. Journal of Risk 3, 5-40 (Winter 2000/2001)