Enhance Institutional Trading Performance:

Leveraging AlgoWheels and Advanced Cost Models

FLEXTRADE

Table of Contents

Vini Satish is VP of Trading Analytics and currently leads development within Flextrade's TCA & Analytics group, having launched the firm's trading research effort in 2010. He currently focuses on bringing advanced analytics and TCA to achieve intelligent automation. Prior to FlexTrade, Vini was a Principal and Co-founder of Quantanium LLC where he led an R&D team in applying machine learning and data mining to initiate and back test trading models, perform factor research, and improve portfolio construction.

He holds S.B. and M. Eng. degrees in Electrical Engineering and Computer Science from MIT, and an MBA in Quantitative Finance from NYU / Stern.

Optimizing the AlgoWheel using Pre-and Post-Trade Cost Models

In global equities trading, institutional participants invest substantial effort in selecting the best mix of brokers, algorithms, and strategies to reduce trading costs and deliver best execution. At times, this requires painstakingly stitching together disparate data sources and sifting through complex data to drive meaningful insights.

AlgoWheels gained momentum over six years ago as a mechanism to disrupt the A/B tests that dominated institutional attempts to select algorithms and brokers. These early efforts were criticized as over-engineered. Beyond Broker Scorecards[1] discusses the shift of AlgoWheels as a form of A/B testing, to the use of randomized controlled trials (RCTs). RCTs remain the gold standard within the financial industry and trading. FlexTrade's AlgoWheel was built to counter over-engineering concerns and ensures maximal flexibility in constructing automation rules and testing.

Clients have incrementally improved trading performance using FlexTrade's AlgoWheel, often relying on TCA and analytics to digest and act upon the results. However, the most common challenges clients face surround data: collecting, normalizing, and analyzing data, as well as achieving the sample size necessary to derive value from trading experiments.

But the challenges around data don't end there. A Coalition-Greenwich study on globalization of algorithmic equity trading revealed that institutions also struggle with transaction cost analysis. According to the study, "the buy-side's biggest obstacle in evaluating best execution today revolves around analyzing data - algo, venue and overall transaction cost performance."[2]

To address the challenge of algo comparisons across different brokers, FlexTrade's AlgoWheel and TCA & Analytics solutions allow for normalized comparisons, flexibility in routing logic and experiment design. Analytical tools include in-house pre-trade and post-trade cost models allowing institutions to quantifiably improve trading.

The most common challenges clients face surround data: collecting, normalizing, and analyzing data as well as achieving the sample size necessary to derive value from trading experiments.

1 To request access to this whitepaper please visit: https://flextrade.com/beyond-broker-scorecards-whats-next-for-algo-wheels
2 Forster, J. (2023, March 28). Globalization of Algorithmic Equity Trading: A Buy-Side View. Coalition Greenwich. https://www.greenwich.com/equities/globalization-algorithmic-equity-trading-buy-side-view

In this article, we discuss how institutions can use FlexTrade's AlgoWheel and TCA & Analytics tools to enhance institutional trading performance. First, we discuss the key obstacles in evaluating algo performance, then move to integrating pre-trade and post-trade cost models with peer data in the AlgoWheel, next applying pre-trade cost models to AlgoWheel, and lastly, cover reweighting the AlgoWheel using a reinforcement learning based feedback loop.

To draw meaningful conclusions from AlgoWheel or TCA results, there needs to be enough historical data.

I. Key Obstacles in Evaluation of Algo Performance

The problem of analyzing broker or algo performance is multifold, and incorporates algo design, comparison, and statistical questions. Below we will discuss the top 5 key obstacles in evaluating an algo's performance.

Market Noise and Sample Size. The first obstacle when analyzing AlgoWheel results, or even standard broker performance, is market noise. Market noise arises from many factors including volatility, market events, and trading activity, and it typically dwarfs the expected, difference in performance between brokers and strategies. The principal aim of both the pre-trade and post-trade models in the AlgoWheel is to reduce noise. This allows institutions to compare trades against a proper benchmark.

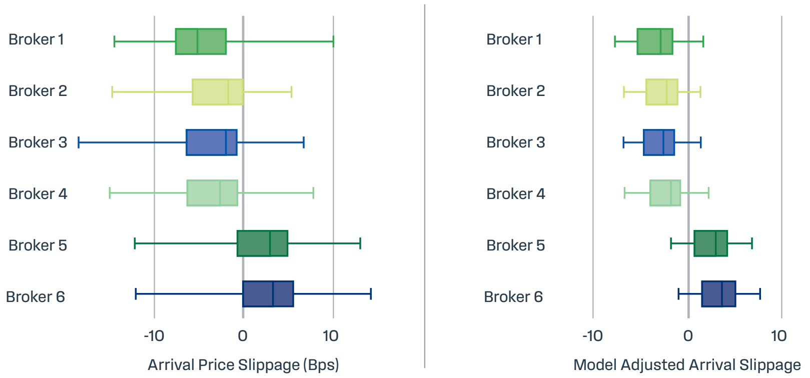
A second obstacle institutions encounter is **sample size.** To draw meaningful conclusions from AlgoWheel or TCA results, there needs to be enough historical data. The analytics derived from FlexTrade's AlgoWheels are optimized to gain actionable insights from a client's trading flow, within the parameters they allocate to testing. While reducing noise is the primary mechanism within the AlgoWheel inputs, there is flexibility in both the inputs and rule structures.[3]

To illustrate the value of noise reduction, consider the two broker comparisons in Figure 1. The left-hand series shows a box plot that visually displays the range and quartile cut points of the distribution of results for a particular AlgoWheel experiment using Arrival Slippage[4] as the benchmark. The right-hand shows the same series but displays only the excess beyond what is predicted by FlexTrade's Post-Trade model. The reduction in variance in the distributions allows the trader to make a clear comparison.

[3] The benefits of noise reduction address both standard TCA and pre-trade inputs into the wheel. In particular, it is common to assess the excess return over the model forecast. This is referred to as the Excess Cost Model (ECM) adjusted arrival price, and often "net cost" in the TCA front end.

[4] Arrival Slippage, Arrival Price Slippage is the change in price from the last price at the time an order was received. We assign slippage consistently as positive (+) representing out-performance (depending on side) and negative (-) representing under-performance. The positive or negative slippage measure accounts for whether the order was a buy or a sell. Example: if arrival price is lower than executed price for a buy order, then slippage is negative. For a sell order, executing at a higher price than arrival, would mean a positive slippage. "Buy low, sell high."

Figure 1: Broker Comparisons using Post-Trade Model



Parameter Normalization. Algorithms have many parameters that vary wildly per broker, style[5] and scale[6]. Trying to stitch and map this data together from different brokers is nearly impossible given the levels of variation and complexity involved. FlexTrade's TCA & Analytics tools map and normalize these parameters to assist in proper experiment design and comparison.

Order Difficulty Adjustment. Industry sectors can have large intraday moves that add an additional hurdle for comparison. Consider the problem that arises when comparing a high-volatility trade of a high-volume stock on a day when its sector moved away vs. a low-volatility trade of a low-volume stock where the sector stayed flat, but both trades had numerically equal slippage. While on paper the performance of these two trades might be the same, observers would agree the former represents exceptional performance and the latter is mediocre.

FlexTrade's post-trade model within the TCA & Analytics tool adjusts for order difficulty allowing institutions to run comparisons that are insightful and actionable.

Outlier Handling. Market return distributions are fat-tailed, which means that very large movements are more common than expected in the normal distribution that drives ordinary statistical comparisons.

While noise reduction can help, outliers require the use of unbiased estimators[7] and adjustments to the data such as winsorization[8]. The intent of winsorization is to limit extreme values in the statistical data to reduce the effect of possibly false outliers. Proper analysis requires one or both methods to address the case of one large outlier skewing the performance of a particular broker or algo.

5 Execution style, Urgency, Aggressiveness
6 3-point scales ranging to 10-point scales
7 e.g. trimmed means
8 i.e. the capping of extreme values at a particular percentile of the distribution, such as the 1st and 99th

II. Integrating Pre-Trade and Post-Trade Cost Models with Peer Data in the AlgoWheel

While brokers offer generic pre-trade and post-trade models which can be used with the AlgoWheel, FlexTrade's in-house pre- and post-trade cost models were built to address algo and strategy selection problems and avoid licensing policies that can restrict the use of model data in third party tools.

Moreover, FlexTrade's in-house models are particularly suited to evaluating algorithmic trading. Advances in machine learning mean that in-house models have better accuracy than older broker models often distributed as cost curves. Machine learning is applied to several different facets of the R&D process at FlexTrade, including factor discovery, feature engineering, model construction, and in the simulator underlying the feedback loop.

FlexTrade's peer-data effort anonymizes and tags data to train the in-house cost models and feedback loop simulations[9]. A substantial portion of the data draws directly from AlgoWheel experiments, with models paying particular attention to performance attribution within the Wheels. The result is appropriately oriented models that result from the pooled data.

III. Applying Pre-Trade Cost Models to AlgoWheel

Pre-trade cost models can be used to route orders based on their difficulty, allowing for better handling and granular comparison of algo strategies from different brokers and across different wheels. They can also be used in conjunction with AlgoWheel to experiment on which brokers and strategies perform better when the expected cost varies. For example, by using pre-trade cost models it is possible to create an aggressive liquidity seeking wheel with various brokers, rather than having separate wheels for different brokers using different algo styles to minimize cost based on order size.

FlexTrade's in-house global equity pre-trade models perform well in the aggregate and serve as a handicap measure

Pre-trade cost models only examine order and market characteristics available before the execution of an order. Typical characteristics include the target quantity series, historical volume, volatility[10], returns, and spread. They are commonly examined against the observed cost.

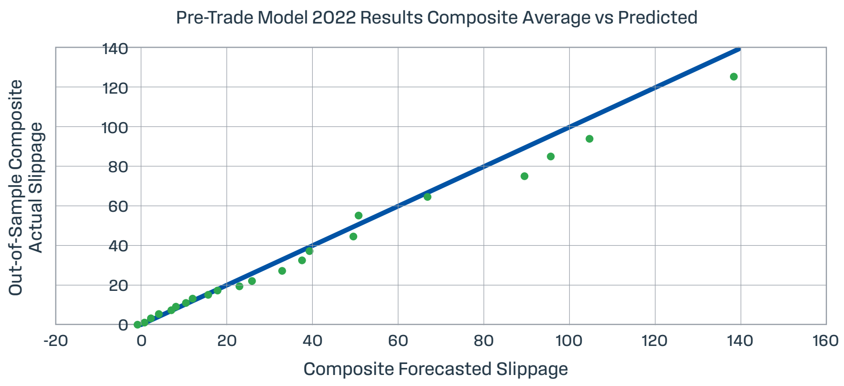
Pre-trade cost models predict the benchmark and the average cost of a basket of trades. While cost models work very well on aggregate, they generally do not work well on an individual order.

[9] Clients must opt-in to share this data
[10] Both historical and current

Figure 2, below, shows the aggregate predictive power of cost models through the accuracy of FlexTrade's global pre-trade model. The aggregated out-of-sample forecast is on the x-axis and the actual aggregated slippage is on the y-axis. To align with trading literature, we reverse the sign of slippage so higher cost is represented with a more positive number.

Performance-based AlgoWheel weighting is an evolution from standard weighting.

Figure 2: Pre-Trade Forecast Aggregate Performance



The key takeaway from Figure 2 is that FlexTrade's in-house global equity pre-trade models perform well in the aggregate and serve as a handicap measure. This can be seen in the tracking error between the plotted points which represent the accuracy of the pre-trade model and the unity line which represents a perfect forecast ability. While there is mild deterioration in accuracy at higher cost levels, around 95% of the trading data sample is within the region that is less than 20 basis points[11]. For those interested in the details, the underlying data at a particular point has been aggregated in a sliding interval across the sample and is represented by the midpoint of the interval. That midpoint is used to represent both the predicted and the actual composite average value for a particular window.

IV. Reinforcement Learning-based Feedback Loop

FlexTrade has developed a reinforcement learning feedback loop within the AlgoWheel solution. The feedback loop automatically increases the weight toward higher performing destinations.

Traditionally within the AlgoWheel, traders initially create wheels by routing equal dollar amounts of orders to a pre-determined number of brokers. During the standard quarterly performance analysis the wheels are re-weighted to send more order flow to the best performing brokers, while decreasing the allocation to brokers which have underperformed the benchmark.

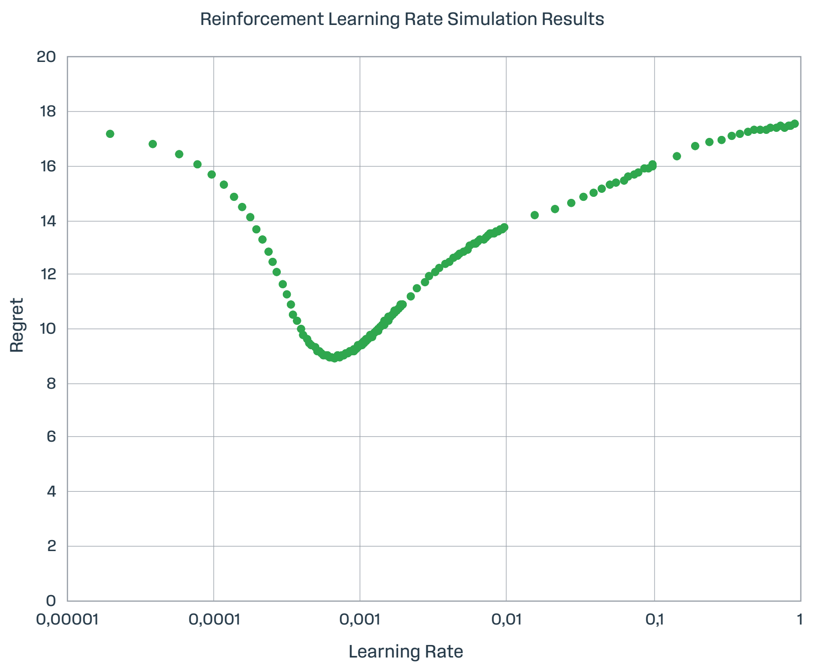
[11] Much of this deterioration is owed to statistical artifacts

Performance-based AlgoWheel weighting is an evolution from standard weighting. The key to understanding performance based AlgoWheel weighting is to understand the concepts of exploration and exploitation. Exploration is the time it takes to identify the best destination, and exploitation is the benefit from routing to the best destination. Building a performance based AlgoWheel weighting capability requires a trade-off between the two. The speed with which a system switches from exploration to exploitation is called the learning rate. The goal is to create a wheel that starts an equal random weight, and through machine learning with exploration and exploitation, gradually shifts the highest weight towards the top performing broker.

FlexTrade's Feedback loop frames the destination selection problem (i.e. the best broker) as a multi-armed bandit problem. In the classic multi-armed bandit problem, a system has no foreknowledge of the reward distribution and only learns the rewards by making individual selections. In the AlgoWheel destination case, that could be the different broker and algorithm combinations.

The model then creates rewards for the brokers and weights all the non-random weight pool toward the best observed broker and reweights the wheel. This model, known as Epsilon Greedy, can automate the algo wheel based on how the client's algos are performing over a certain number of days and then train the model on it.

Figure 3: Reinforcement Learning Rate Simulations Results



The learning rate is the speed at which the system switches from exploration to exploitation.
Regret represents the trade-off between different learning rates.

Monte Carlo style simulations are used to identify the optimal learning rate across millions of simulations. Plotting the delta between the optimal broker and the selected path as averaged over many simulations, also known as regret, represents the tradeoff between different learning rates. Figure 3 plots regret along the x-axis to represent different learning rates, with higher x values representing higher learning rates. The reward is the slippage against arrival from picking a particular broker.

Figure 3 illustrates two concepts:

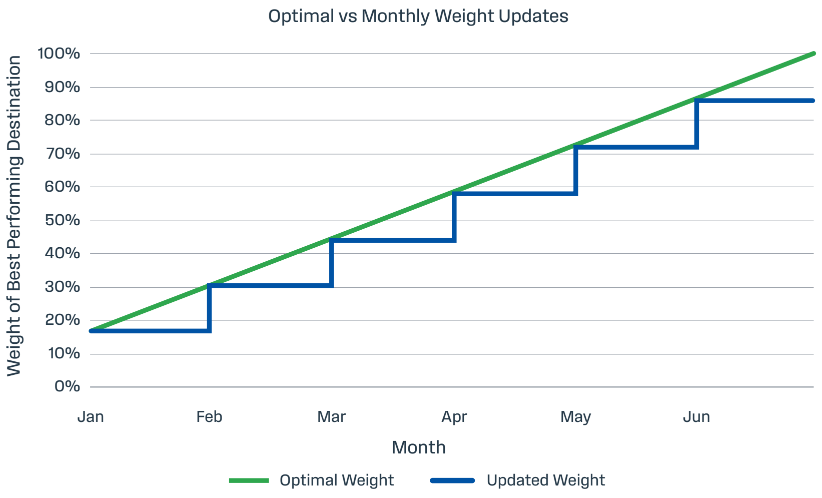
1. Learning rate, which is the speed at which the system switches from exploration to exploitation
2. Regret, which is the delta between the optimal broker and the selected path as averages over many simulations

Over the course of this particular trial, it was possible to save 10 bps per trade over the life of the experiment. The simulation used 3000 steps, typical for an AlgoWheel trial, and represents substantial savings for an institution.

Some clients prefer to use regular arrival price as the input as the results are readily interpretable as savings, and a sufficient sample size will cancel out the market movements over time. However, using a model-adjusted slippage instead of arrival within the reinforcement learning mechanism has a measurable effect on the trading performance a client can obtain using the AlgoWheel.[12] This is because the variance reduction properties from the model help the reinforcement learning algorithm reduce the time it takes to determine the optimal broker by as much as 60%.

The following chart illustrates the benefit of using automatic re-weighting vs the standard TCA approach to rebalancing weights.

Figure 4: Reinforcement Learning Rate Simulations Results



[12] Post-trade cost models examine factors and conditions that occurred throughout the entire life cycle and post-life cycle of the order. Typical factors and conditions include interval volume, fill size, market drift, the evolution of spread, and index beta.

V. Conclusion

Buy-side institutions are turning to Algowheels to minimize trading costs and achieve best execution. By removing market noise and normalizing disparate data sets along with statistical outliers, transaction cost models provide the tools to properly analyze broker performance. As an input to the AlgoWheel, the normalized data is used to draw comparisons to discover the optimal mix of brokers, algorithms, and strategies. This is further improved as desks adopt experiments as part of the process, which helps them gain a competitive edge.

FlexTrade built its AlgoWheel to ensure maximum flexibility in constructing automation for testing and rules providing incremental improvements in trading performance. In addition to addressing industry issues with collection and data normalization head on, the FlexTrade AlgoWheel and TCA & Analytics products deliver the sample size needed to extract value from trading experiments.

Through the application of pre- and post-trade models, and the use of peer data and techniques such as reinforcement learning to train the AlgoWheel, institutional traders can overcome the data challenges of setting up algo wheels and designing experiments to achieve actionable results.

Here are key takeaways from this report:

- The key obstacles for evaluating algo performance include market noise and sample size, parameter normalization across brokers, and order difficulty adjustments, along with outlier handling.

- Normalized parameters assist in conducting proper experiment design and comparison. FlexTrade's post-trade model within the TCA & Analytics tool adjusts for order difficulty allowing institutions to run comparisons that are insightful and actionable.

- Using anonymized peer data, in-house cost models and feedback loop simulations can be trained and used directly within AlgoWheel experiments.

- A reinforcement learning-based feedback loop automatically increases the weight toward higher performing broker algo destinations, while decreasing weights to lower performing algo destinations.

- The application of the post-trade model to the feedback loop resulted in a 60% reduction of the time it took to identify the optimal broker.

FLEXTRADE