Beyond TWAP: Smarter Rule-Based Execution Algorithms

Abhinav Chhabra, Sarvesh Sakhare, Ananya Bajaj

17 May 2025

1 Introduction

In recent years, quantitative execution strategies have become central to algorithmic trading, offering data-driven methods to reduce trading costs and manage market impact. This project develops and evaluates two systematic execution algorithms - an event-driven strategy and a microprice-and-spread-tier-based strategy, designed to improve upon the baseline Time-Weighted Average Price (TWAP) benchmark. TWAP, which executes trades evenly over a fixed interval to minimize market impact, is widely used for its simplicity but often fails to exploit real-time market conditions.

Using historical limit order book data across multiple stocks and market regimes, we incorporate microstructural features such as bid-ask spread, weighted mid-price, order flow imbalance, momentum, and volume imbalance to generate trading signals. Each strategy executes trades only under favorable market conditions, with built-in fallbacks to ensure timely execution.

Through simulation, grid search-based threshold tuning, and backtesting, we demonstrate that our strategies consistently outperform TWAP in terms of reduced average spread across diverse tickers and market conditions. The results underscore the value of feature-driven, adaptive execution logic over naive time-based methods and highlight opportunities for further improvements, such as incorporating learning-based models and richer order book signals.

2 Data Dictionary

2.1 Datasets

The dataset comprises historical high-frequency trading data for five major U.S. equities: Amazon (AMZN), Apple (AAPL), Alphabet (GOOG), Microsoft (MSFT), and Intel (INTC). Each stock's data is stored in a separate CSV file and includes the following columns: timestamp, ask price, ask volume, bid price, bid volume, order type, order id, size, order price,

order direction, mid price, and spread. These variables capture granular details of the order book and trade events necessary for execution strategy analysis.

2.2 Duplicate Handling and Timestamp Normalization

As a pre-processing step, we eliminated duplicate rows from the dataset to prevent redundant data from skewing strategy performance. Furthermore, we addressed cases where multiple bid and ask records appeared with identical timestamps, suggesting updates faster than the dataset's minimum time resolution. To handle these collisions, we aggregated such entries by computing the average price at each timestamp. This consolidation ensures a more representative and smoother view of the prevailing best bid and ask prices, preserving the integrity of market microstructure signals used in our algorithms.

2.3 Data Symmetrization for Robustness Testing

The raw dataset primarily reflected bearish market conditions, which could bias the evaluation of our trading strategies. We created synthetic bullish market data to test robustness across different market regimes by symmetrizing the existing dataset. This was achieved by subtracting a large constant offset from both bid and ask prices, effectively inverting the price trend and swapping bid and ask volumes to mirror the underlying demand-supply dynamics. This transformation preserved the micro-structural characteristics of the original data while simulating upward market movement, allowing us to evaluate strategy performance in both bearish and bullish environments.

2.4 Additional Features Added

To enhance the dataset and enable more informed decision-making in strategy design, several engineered features were introduced:

- Spread Ticks: This represents the bid-ask spread expressed in terms of discrete tick units. It reflects market liquidity and trading cost, i.e., smaller values indicate tighter markets and lower execution slippage. This was calculated as the spread divided by tick size so that the spread value is an integer. Tick size for our case = 0.01.

$$\text{spread\_ticks} = \frac{\text{Ask Price} - \text{Bid Price}}{\text{Tick Size}}$$

- Order Flow Imbalance: A measure of the relative buying and selling pressure in the market. It captures short-term supply and demand dynamics and is useful for predicting price direction. It is calculated as the rolling sum of order flow imbalance for the past 50 points.

$$\text{OFI} = \sum_{j=t-k+1}^{t} \left(V_b(j) - V_a(j)\right)$$

  where $V_b(t)$ is the bid volume at time t and $V_a(t)$ is the ask volume at time t, and k is the window length which is equal to 50 for our case.

- Momentum: This feature captures the recent trend in the mid price, helping to identify directional moves in the market. Positive momentum suggests upward movement, while negative momentum indicates downward movement. It is calculated as the logarithmic difference of the mid price over a rolling window of the past 50 points in the data.

$$\text{Momentum} = \ln(\text{Mid}(t)) - \ln(\text{Mid}(t-k))$$

  where k is the window length which is equal to 50 in this case.

- Microprice: A forward-looking price estimate that considers both bid and ask prices and their corresponding volumes. It accounts for order book imbalance and often provides a better representation of where the next transaction might occur. It is calculated as the weighted average of the bid and ask price, adjusted for volume, i.e., the normalized imbalance between bid and ask volumes, highlighting market inefficiencies.

$$\text{Micro Price} = \frac{\text{Bid Price} \times \text{Ask Volume} + \text{Ask Price} \times \text{Bid Volume}}{\text{Bid Volume} + \text{Ask Volume}}$$

- Price Velocity: A short-term rate of price change that captures how quickly the mid-price is moving. This feature helps detect bursts of activity or volatility that may signal an opportunity for execution. It is calculated as the 5-period (past 5 data points) difference of the mid price.

$$\text{Price Velocity} = \text{Mid}(t) - \text{Mid}(t-5)$$

These features were normalized to ensure comparability across stocks and time periods, aiding model training and strategy development.

3 Strategies

To improve execution quality relative to the TWAP benchmark, we implemented and tested a variety of execution strategies. Each strategy leverages different market signals, decision rules, and adaptive behaviors to optimize fill timing and minimize trading costs. After testing multiple strategies, we finalized the following two strategies since they performed the best on the given stocks.

3.1 Event-Driven Dual Trade Execution Strategy

Strategy used for: GOOGL, INTC, MSFT

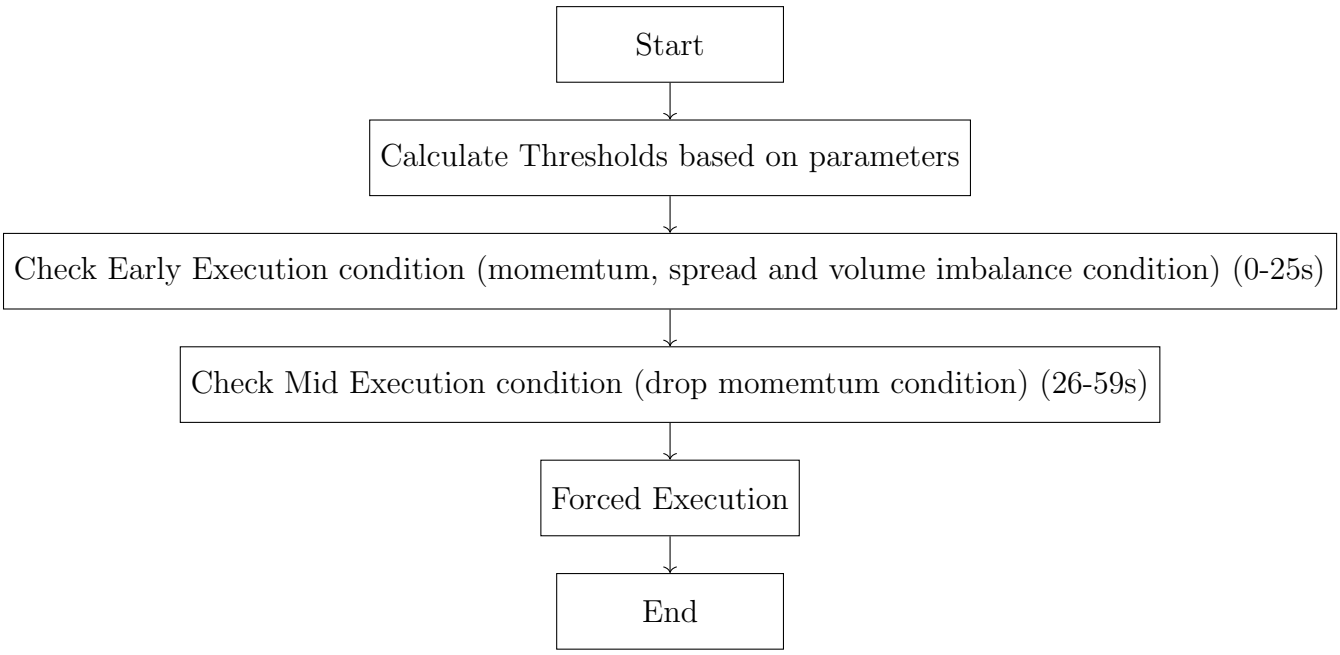
Description:

The `exec_event_driven_dual` strategy is designed to execute both buy and sell trades based on market events in a single pass over the data. The strategy leverages market features such as spread ticks, volume imbalance, and price velocity to make trading decisions. The

execution logic is divided into three stages: Early Execution, Mid Execution, and Fallback Execution.
The strategy tries to manage overall risk by first trying to execute in favorable market conditions. It then slowly relaxes the thresholds as execution pressure builds up.

- **Data Preprocessing:** The dataset is grouped by minute intervals, and relevant features such as elapsed time are computed.
- **Threshold Calculation:** Spread ticks, volume imbalance, and price velocity thresholds are precomputed.

Flowchart:



Parameters:

- spread threshold: this is a quantile threshold. We check whether the spread is greater than the value at the spread threshold quantile in the data so far. It is used to check early and mid execution condition (0s-59s)
- volume imbalance threshold: used to check early and mid execution condition (0s-59s)
- momemtum (price velocity) threshold: used to check early execution condition (0s-25s)

Execution Logic:

For Each Minute Group:

1. Compute fallback prices as the last observed bid and ask prices.

2. Evaluate **Early Execution** conditions:
    - If elapsed time is less than 25s and the following conditions are met:
    - If `spread` $\leq$ `spread_thresh` and
    - `vol_imb` $>$ or $<$ `vol_imb_threshold` and
    - `mom` $\geq$ `mom_threshold`,
    - Then execute the trade.
3. If not executed, evaluate **Mid Execution** conditions (for $25 \leq \tau < 59$ seconds):
    - If `spread` $\leq$ `spread_thresh` and
    - `vol_imb` $<$ `vol_imb_threshold`,
    - Then execute the trade.
4. If still not executed, forced execution using the last observed prices.

3.2 Combined Micro Price and Spread Tiers

Strategy used for: AAPL, AMZN

Description:

The `exec_microprice_tiers` strategy is designed to execute trades based on a combination of microprice behavior and tiered spread conditions. This strategy prioritizes early fills when favorable conditions exist and guarantees execution by the end of the minute. It dynamically evaluates market conditions over predefined time intervals, using progressively relaxed thresholds to maximize trade quality while minimizing slippage.

- **Data Preprocessing:** The strategy uses pre-processed data and grouped by minute intervals, and computed features such as micro-price, spread ticks, and time are used for each quote.
- **Threshold Configuration:** Spread thresholds are predefined for different elapsed time intervals (e.g., 5s, 15s, 25s, 45s, 59s). The thresholds are relaxed as we go deeper in the minute since we have execution pressure.

Parameters:

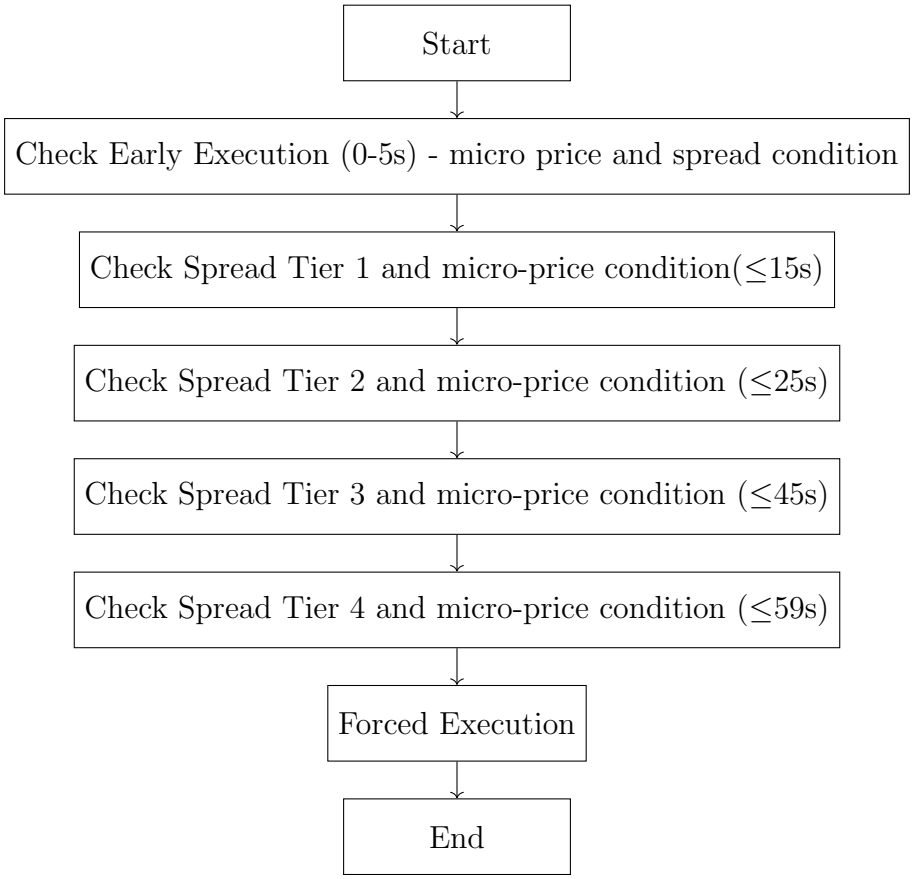
- param1: the spread threshold from 15s-25s
- param2: the spread threshold from 25s-45s
- param3: the spread threshold from 45s-59s

Execution Logic:

For Each Minute:

1. Evaluate **Early Execution**:
    - If elapsed time $\leq$ 5s and spread $\leq$ 1 tick and the weighted mid-price is close to the bid or ask ($\leq$spread/2): execute immediately.
    - If elapsed time $\leq$ 15s, spread $\leq$ 2 ticks, and weighted mid-price is close to bid price or ask price ($\leq$spread/2): execute.
2. If no early execution, evaluate spread tiers based on time:
    - 0–25s: spread $\leq$ param1 ticks and weighted mid-price is close to bid price or ask price ($\leq$spread/2)
    - 25–45s: spread $\leq$ param2 ticks and weighted mid-price is close to bid price or ask price ($\leq$spread/2)
    - 45–59s: spread $\leq$ param3 ticks and weighted mid-price is close to bid price or ask price ($\leq$spread/2)
3. If still not executed, forced execution using latest available prices at t=59s.

Flowchart:



The early execution conditions ensure faster fills in alpha-favorable conditions (i.e., tight spread and favorable microprice), while still guaranteeing an execution by the end of the minute if no better opportunity arises.

4 Adjusments and Optimization

To improve the performance of our execution strategies, we implemented a grid search-based optimization procedure to fine-tune key parameters. For the event-driven strategy, we optimized thresholds related to spread ticks, volume imbalance, and momentum using both bear and synthetic bull market data. Similarly, the microprice and spread tiers strategy was calibrated by adjusting spread thresholds at various time intervals throughout the minute.

The optimization objective across both strategies was to minimize the average spread during execution on both bull and bear data, thereby enhancing execution quality. Optimal parameter sets were computed individually for each stock (e.g., GOOG, MSFT, INTC for the event-driven strategy and AAPL, AMZN for the microprice-based strategy), reflecting their distinct liquidity profiles and market behaviors. Post-tuning results showed noticeable improvements in execution performance across most stocks, validating the effectiveness of data-driven parameter tuning in enhancing algorithmic execution efficiency.

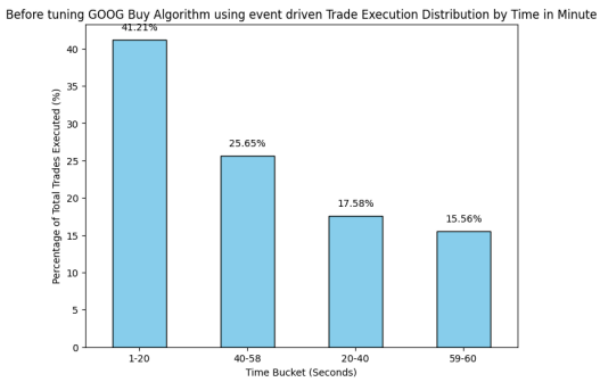


Figure 1: Minute-wise Execution Breakdown for GOOG pre-Event-Driven-optmization

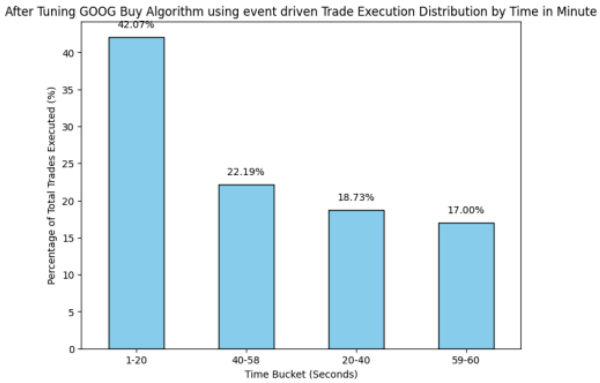


Figure 2: Minute-wise Execution Breakdown for GOOG post-Event-Driven-optmization

There was a marginal change in the percentage of executions across the different time tiers, post optimization as can be seen in the plots above.
Best parameters for each of the stocks were as follows:

Ticker	Strategy	Param 1	Param 2	Param 3
AAPL	Micro Price Spread Tiers	3	4	8
AMZN	Micro Price Spread Tiers	2	3	4
GOOG	Event Driven	0.3 (Vol Imb)	0.175 (Spread)	0 (Mom)
INTC	Event Driven	0.7 (Vol Imb)	0.475 (Spread)	0 (Mom)
MSFT	Event Driven	0.2 (Vol Imb)	0.325 (Spread)	0 (Mom)

Table 1: Optimal strategy parameters

5 Results

5.1 Performance on Training Data

The following table summarizes the details of the execution results for the two strategies when run on the training data.

Symbol	Strategy	TWAP Spread	Avg Buy Price	Avg Sell Price	Avg Spread	TWAP Improv.
AAPL	Combined Micro Price and Spread Tiers	0.1572	584.0443	583.9934	0.0509	0.1063
AMZN	Combined Micro Price and Spread Tiers	0.1356	223.0581	222.9780	0.0801	0.0555
GOOG	Event Driven	0.2812	571.4671	571.3125	0.1546	0.1266
INTC	Event Driven	0.0101	27.0924	27.0885	0.0039	0.0062
MSFT	Event Driven	0.0101	30.6010	30.5978	0.0031	0.0070

Table 2: Strategy Performance by Symbol

5.2 10-Minute Strategy Execution Analysis

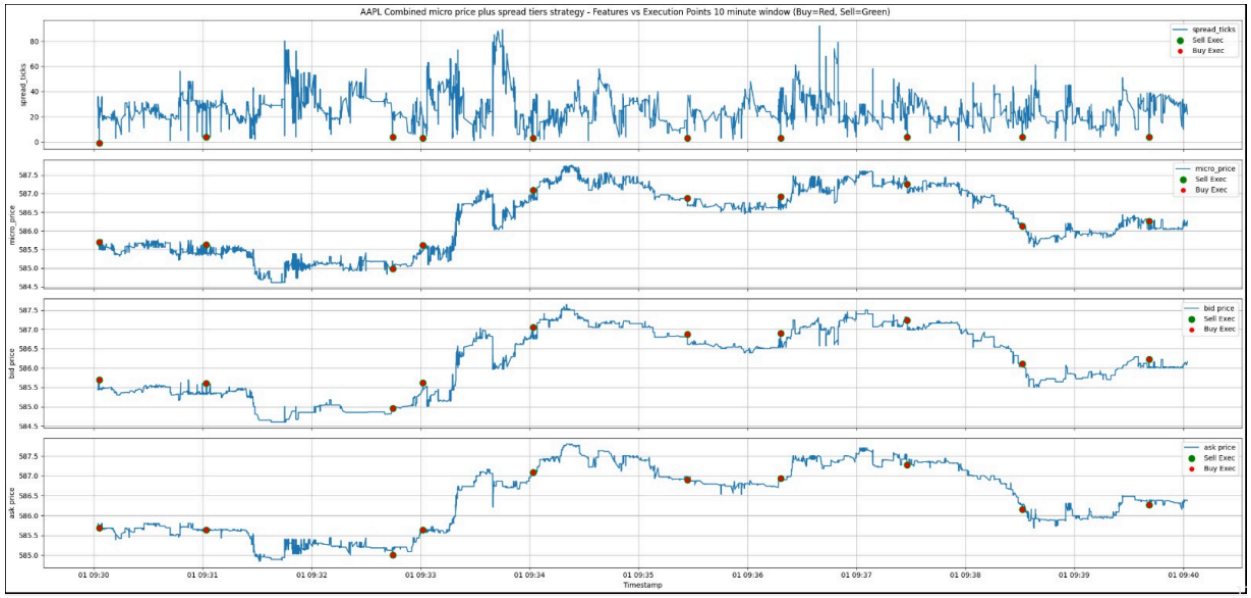


Figure 3: Buy / Sell Triggers by the Combined Microprice and Spread Tiers Strategy on a 10-minute window of data for AAPL

Figure 3 reveals overlapping buy (red) and sell (green) triggers throughout the entire 10-minute window. This occurs because the Combined Microprice and Spread Tiers Strategy prioritizes minimal spread conditions, with execution triggers clustering around points where the spread was narrowest during the observed period.

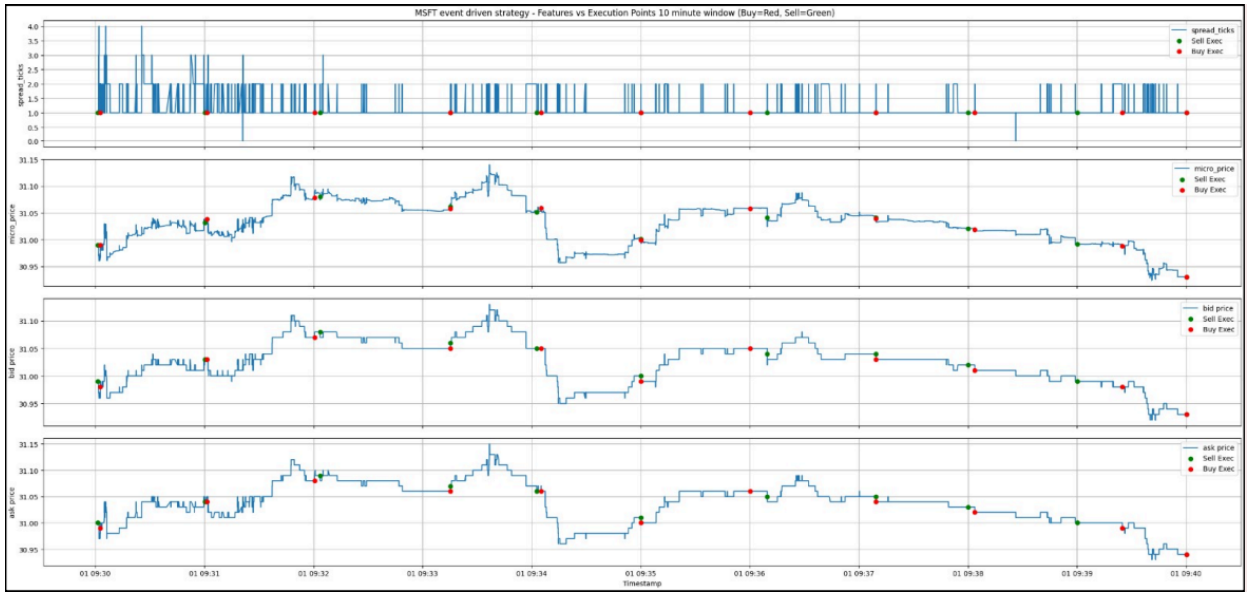


Figure 4: Buy / Sell Triggers by the Event-Driven Strategy on a 10-minute window of data for MSFT

Figure 4 demonstrates that the buy and sell triggers occur in close proximity but, unlike the previous strategy, do not overlap. This behavior stems from the event-driven strategy's reliance on multiple dynamic factors, which determine execution timing based on the most favorable combination of these variables.

5.3 Strategy Behavior Over Time

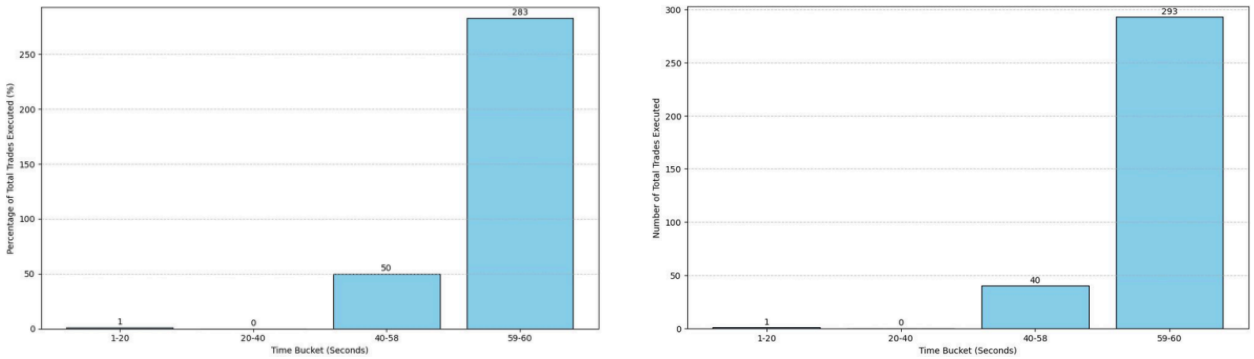


Figure 5: Second-wise Breakdown of TWAP Functioning on AAPL and INTL

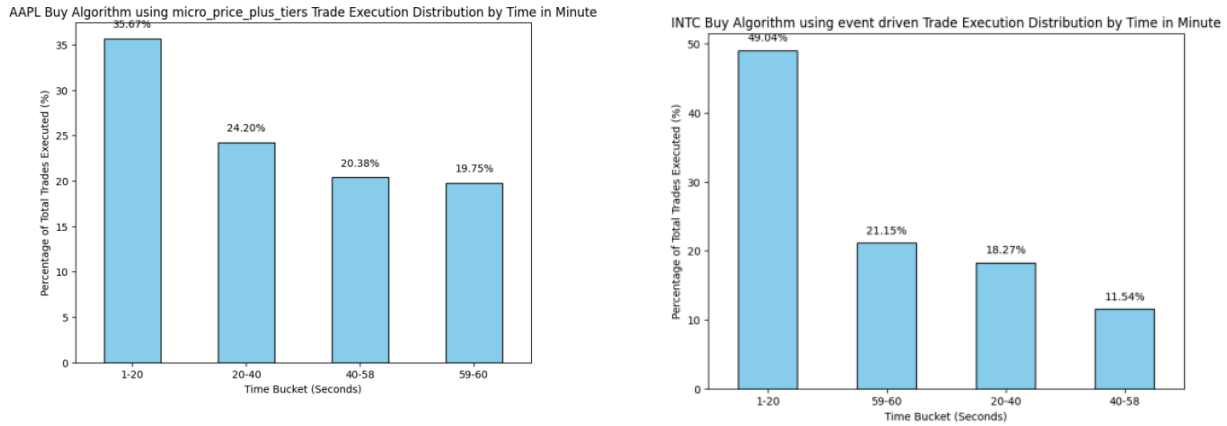


Figure 6: Second-wise Breakdown of the Strategies' Functioning on AAPL and INTL

As illustrated in Figure 5, the simple TWAP strategy adheres to a naive execution approach, resulting in a concentration of trade calls in the final second of each minute across all equities.

On the other hand, the rule-based trading algorithms distribute executions evenly across the second-tier intervals within each minute. This balanced distribution indicates that our trigger constraints, designed to identify optimal trade timings, are well-calibrated: neither overly restrictive nor excessively permissive. As a result, the strategy consistently captures favorable execution opportunities throughout the trading minute.

5.4 Results on Backtesting Data

Symbol	Strategy	TWAP Spread	Avg Buy Price	Avg Sell Price	Avg Spread	% Improvement
AAPL	Combined Micro Price and Spread Tiers	0.1421	578.6813	578.6500	0.0313	77.97%
AMZN	Combined Micro Price and Spread Tiers	0.1031	221.0234	220.9916	0.0318	69.15%
GOOG	Event Driven	0.1992	564.9772	564.9326	0.0447	77.56%
INTC	Event Driven	0.0146	26.7246	26.7212	0.0033	76.71%
MSFT	Event Driven	0.0139	30.1672	30.1649	0.0023	83.45%

Table 3: Strategy Performance by Symbol

5.5 Observations and Key Takeaways

Based on the backtesting results, the following observations were made:

- Spread emerged as the dominant feature across all strategies. Even when other features were removed, strategies incorporating spread metrics consistently outperformed TWAP, albeit with a marginal performance decline. Weighted mid-price was the second most impactful feature, contributing to execution quality but with lesser influence than spread.
- Combined Micro Price and Spread Tiers worked best on AAPL and AMZN. These stocks showed moderate TWAP spreads and noticeable improvement using microstructure-aware strategies. A possible reason could be their high liquidity and tight spreads which in turn allowed microprice and spread thresholds to capture better entry/exit points with minimal market impact. These strategies benefit from stable order book dynamics, which AAPL and AMZN typically exhibit.
- Event-Driven strategy worked best on GOOG, followed by INTC and MSFT. GOOG had the largest TWAP spread and highest improvement, suggesting larger price moves or opportunities around market events. Event-driven strategies likely captured price momentum or reversals better in more volatile conditions. GOOG may be less predictable in short-term microstructure, favoring event-based triggers over microprice/spread logic. Although INTL and MSFT exhibited very narrow TWAP spreads, the strategy still achieved meaningful improvements.

5.6 Reasoning for Potential Efficacy of Strategies

Microprice and Spread-based Strategies are more effective in:

1. High-liquidity environments.
2. Stocks with tight but slightly variable spreads.
3. Conditions where order flow imbalance and spread behavior are predictive.

Event-driven strategies perform well in:

1. Stocks that react to macro or firm-specific news.

2. Assets with larger price dislocations or volatility spikes.

3. Situations where timing based on external signals matters more than microstructure.

6 Conclusion

This report compared the performance of two rule-based trading strategies: a simple method that combined micro price and spread tiers, and a reactive event-driven approach. These strategies were tested across a group of well-known stocks. The results show that the success of each strategy depends on the specific behavior and structure of the stock being traded.

Overall, the findings support the idea that straightforward, rule-based methods can perform well across different types of stocks and market conditions, leading to strong improvements in execution. For example, the microstructure-based strategy worked well for liquid and stable stocks like AAPL and AMZN, proving that you do not always need complex models to get good results. Well-designed rules based on market behavior can be both practical and effective.

The event-driven strategy performed best for stocks like GOOG, which react more strongly to news and events. Interestingly, both INTC and MSFT showed good improvement, even though they had very small spreads and seemed highly efficient. This suggests that there are still small inefficiencies in the market that simple strategies can take advantage of.

Ultimately, the results support a modular and adaptive approach to execution design. While sophisticated methods may offer incremental gains in certain contexts, simple rule-based strategies remain a powerful and interpretable tool, particularly when tailored to the microstructural features of the underlying asset. Future work could focus on integrating these strategies into a hybrid framework that dynamically selects execution logic based on real-time market regimes and asset profiles