

Order Book Mechanics and the Cost of Execution

A limit order book with price-time priority, and what four ways of working the same parent order actually cost.

Sophia Ize-Iyamu · Independent research · September 2026

WHY THIS EXISTS

Most execution commentary compares average fill prices. That measurement is wrong in a specific and expensive way: it prices the shares you bought and ignores the shares you failed to buy. A passive order that fills two-thirds of a parent looks cheap on every filled share and leaves the remaining third exposed to whatever the market did while it waited. This study implements the book itself, runs the same parent order four ways, and prices the whole order.

THE BOOK

A limit order book in Python with strict price-time priority. It matches an aggressive order across levels at resting prices, so the taker receives price improvement where it exists and pays the worse level once the touch is exhausted. It supports resting limits, market orders, IOC and FOK, partial fills, and cancels. Each resting order records the volume queued ahead of it at the moment it rests, and that figure drains as the orders in front of it trade, which is how queue position is actually earned and lost.

The mechanics are covered by self-tests: that the first order at a price fills before the second, that a market order walks to the next level for the balance, that IOC leaves no residue, that FOK trades all or nothing, and that queue position decrements only when the queue ahead trades.

THE MEASUREMENT

Perold implementation shortfall against the arrival mid. Executed shares are charged the difference between their fill price and the arrival mid. Unfilled shares are charged the difference between the terminal mid and the arrival mid. The two are summed over the full parent notional, so an order that does not finish cannot report a low cost.

A 50,000-share buy, worked over a 240-step window in 2,000-share clips, against a background of refreshing quotes and random taking flow. Two hundred seeded paths per strategy, so the ranking is a distribution rather than one tape.

RESULT

STRATEGY	IS MEAN (BP)	IS MEDIAN	IS 90TH PCT	FILL RATE	PASSIVE FILLS
Aggressive (take)	6.6	6.7	13.5	96.0%	0%
TWAP (take)	5.7	6.3	32.2	100%	0%
Adaptive (post, cross when behind)	5.3	4.6	33.2	100%	38.9%
Passive (post only)	4.7	3.3	39.4	65.4%	100%

The ranking flips depending on where you look. Passive is the cheapest strategy at the median and the most expensive at the 90th percentile. It posts inside the spread and earns it when the market comes to it, and it carries the cost of the 35% of the parent it never completes. Aggression is the most expensive on average and has by far the tightest distribution, at 13.5bp of 90th-percentile cost against 39.4bp for passive. Crossing the spread is not a mistake, it is the price of a narrow outcome.

The adaptive rule, which posts while it is on schedule and crosses when it falls behind, takes the lowest mean of the strategies that actually complete. It earns the spread on roughly two fifths of its volume and buys certainty on the rest. That is the shape of most production schedulers, and this is the arithmetic that justifies them.

WHAT THIS IS NOT

The book is simulated, not a replay of a real venue tape, so the numbers describe the mechanics rather than any particular name. Background liquidity refreshes on a fixed rule, so there is no queue-jumping, no hidden or iceberg size, no fee and rebate schedule, no latency, and no multi-venue routing. Permanent impact is not modelled separately from the spread paid. Each of those pushes the passive tail wider rather than narrower, so the central finding is conservative.

Implementation: Python, no external dependencies. The matching engine, its self-tests and the execution simulator are reproducible from a fixed seed. Implementation shortfall follows Perold (1988), *The Implementation Shortfall: Paper Versus Reality*, Journal of Portfolio Management, which is the standard the industry measures against.