

MTH 9879 Market Microstructure Models, Spring 2019

Lecture 11: Latency and order routing

Jim Gatheral

Department of Mathematics



Outline of Lecture 11

- A stylized model of latency cost due to Moallemi and Sağlam
- A realistic study of latency cost due to Stoikov and Waeber
 - Backtesting using Level-1 data
- Smart order routing algorithms
 - Almgren and Harts
 - Laruelle, Lehalle, and Pagès
 - Kearns et al.
- Optimal limit/market order placement (Cont and Kukanov)

Overview of execution algorithm design

Typically, an execution algorithm has three layers:

- The macrotrader
 - This highest level layer decides how to slice the order: when the algorithm should trade, in what size and for roughly how long.
- The microtrader
 - Given a slice of the order to trade (a child order), this level decides whether to place market or limit orders and at what price level(s).
- The smart order router
 - Given a limit or market order, which venue should this order be sent to?

In this lecture, we are concerned with the lowest level of the algorithm: Where to send orders and the effect of latency.

Latency

We can identify a number of reasons why low latency should be important to traders:

- The greater the latency, the more stale information is.
 - What use is an old order book signal?
- In a market with price and time priority, being ahead in the limit order queue gives a greater chance of execution.
- The ability to cancel orders ahead of others can be very valuable.

Relative latency rather than absolute latency is what really counts.

Moallemi and Sağlam

- The idea of the Moallemi-Sağlam (MS) model is to compare the expected profits on a limit order strategy with and without latency.
- They end up with a simple formula that estimates the cost of latency CL as a fraction of the cost of immediacy δ (basically the bid-ask spread) as follows:

$$CL \approx \frac{\sigma\sqrt{\Delta t}}{\delta} \sqrt{\log \frac{\delta^2}{2\pi\sigma^2\Delta t}}$$

- We see that the formula depends only on the ratio

$$\frac{\sigma\sqrt{\Delta t}}{\delta},$$

the characteristic fraction of the spread that the stock price can move between the time the order is sent and the time the order is received by the exchange.

Comparative statics

The cost of latency CL :

- Increases as latency (Δt) increases.
- Increases as volatility (σ) increases.
- Decreases as the bid-offer spread (δ) increases.

For a small-tick stock, it is found empirically that

$$\delta \approx 2\sigma\sqrt{\tau}$$

where τ is the characteristic time between trades.

Then, if $\epsilon := \Delta t/(4\tau)$, we have, as $\epsilon \rightarrow 0$,

$$CL \approx \sqrt{\epsilon \log \frac{1}{2\pi\epsilon}}$$

- Thus, the more active the stock, the more important it is to have low latency.

The MS trading strategy

In the MS model

- The objective is to sell one share of stock.
- The bid price evolves as arithmetic Brownian motion (ABM), $S_t = S_0 + \sigma W_t$.
- Impatient traders arrive at the rate μ . These traders execute at the bid price plus δ .
- If the limit order is not filled by time T , a market sell order is sent.
- The trader is assumed risk neutral, his goal is to maximize the expected profit of selling one share. Let P denote the transacted price and L the collection of all admissible (continuous) strategies of posting limit orders within the time horizon $[0, T]$. The problem can be recast as the optimization problem

$$h_0 := \max_{\ell \in L[0, T]} \mathbb{E}[P] - S_0.$$

Limit order execution in the MS model

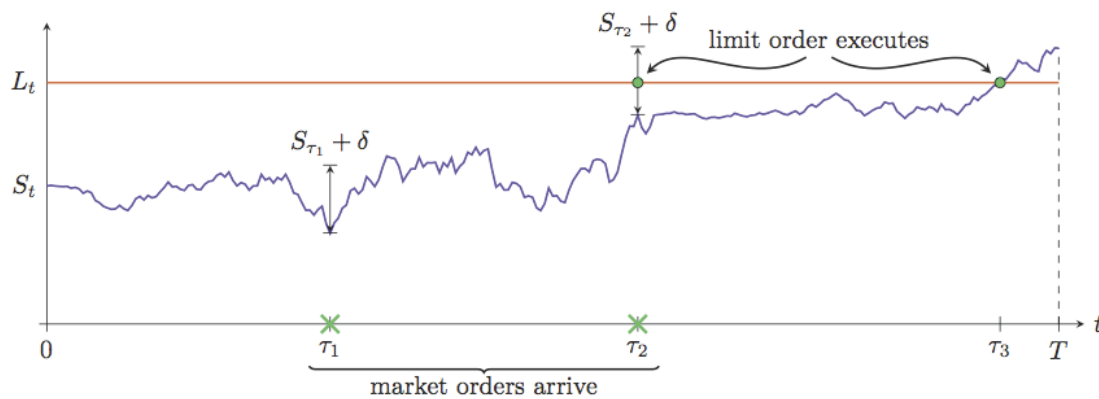


Figure 1: An illustration of the limit order execution in the stylized model over the time horizon $[0, T]$. Here, we assume the trader leaves a limit order with the (constant) price L_t and S_t is the bid price process. If market orders arrive at times τ_1 and τ_2 , the limit order would execute at time τ_2 but not time τ_1 , since the limit order price is in excess of δ to the best bid price. The limit order would also execute at time τ_3 in the absence of a market order arrival, since the bid price crosses the limit order price at this time.

Optimal strategy without latency

- When there is no latency, the optimal strategy ℓ_t^* is to peg a limit order at the bid price S_t plus δ , i.e., $\ell_t^* = S_t + \delta$.
 - Orders are continuously canceled and replaced.
- The optimal value is given by

$$h_0 = \delta (1 - e^{-\mu T}).$$

The optimal strategy with no latency

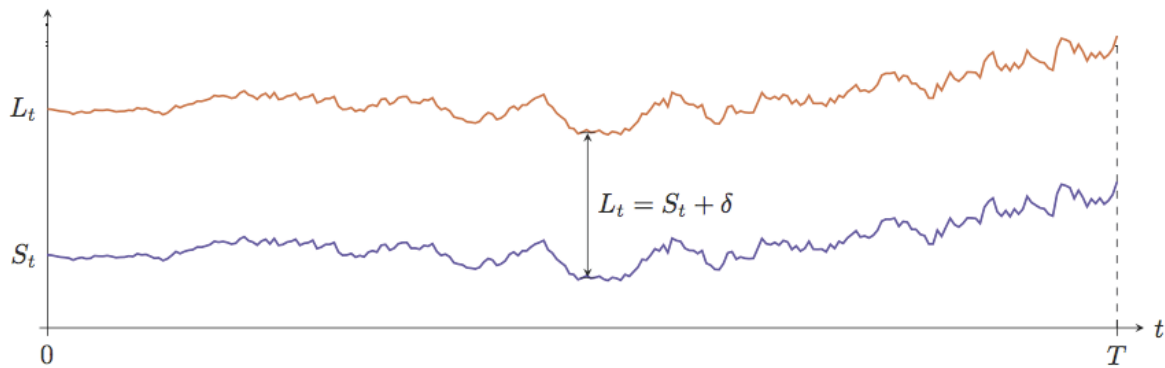


Figure 2: An illustration of an optimal strategy with no latency, over the time horizon $[0, T]$. The trader uses only limit orders prior to end of the time T . The limit order price L_t is pegged to the bid price S_t , with an additional premium corresponding to the bid-offer spread δ .

When there is no latency, the expected cost saving relative to a market order is given by

$$h_0 = (1 - e^{-\mu T}) \delta.$$

Introducing latency

- The interval $[0, T]$ is split into n subintervals of length $\Delta t = T/n$.
 - Δt is the system latency.
- The trader sets the price ℓ_i of the limit order at time $t_i = i \Delta t$.
- This order does not reach the market until time $t_{i+1} = t_i + \Delta t$. It is active from time t_{i+1} to time t_{i+2} .

Scenarios

Between time t_i when the limit price ℓ_i is set and time t_{i+1} when the limit order with this price reaches the market, three scenarios can unfold:

1. \mathcal{A}_1 : An impatient buyer can arrive with probability $\mu \Delta t$. The limit order ℓ_{i-1} in force at that time will be executed if $\ell_{i-1} \leq S_i + \delta$. The one share is sold and there is no more trading.
2. \mathcal{A}_2 : If $S_{i+1} \geq \ell_i$, the limit order will be marketable when it reaches the market and will be filled. Again, the one share is sold and there is no more trading.
3. \mathcal{A}_3 : Otherwise, the limit order at ℓ_i stays in the book for the next period $[t_{i+1}, t_{i+2})$. The process continues.

The optimization problem

The trader is again assumed to be risk-neutral. He seeks to solve the optimization problem

$$h_0(\Delta t) = \max_{\{\ell_0, \dots, \ell_{n-1}\}} \mathbb{E}[P] - S_0$$

where P is the sale price of the stock.

This problem may be solved exactly using dynamic programming with the boundary condition that if there is only one period left, the continuation value is zero because at that point, a market order must be submitted.

Intuition behind the solution

- Consider the situation at time $t = 0$ when the bid price is S_0 .
- If there were no latency, the optimal strategy would be to peg our sell order to the offer at $S_0 + \delta$; if an impatient buyer arrives, our probability of selling to him is one.
- If there is latency, this strategy is no longer optimal because the probability of a fill after time Δt is only $1/2$.
- By lowering the limit price a little however, we can significantly increase the probability of trading with an impatient buyer thus potentially significantly increasing our potential cost saving relative to a market order.

The effect of latency on the optimal choice of limit price

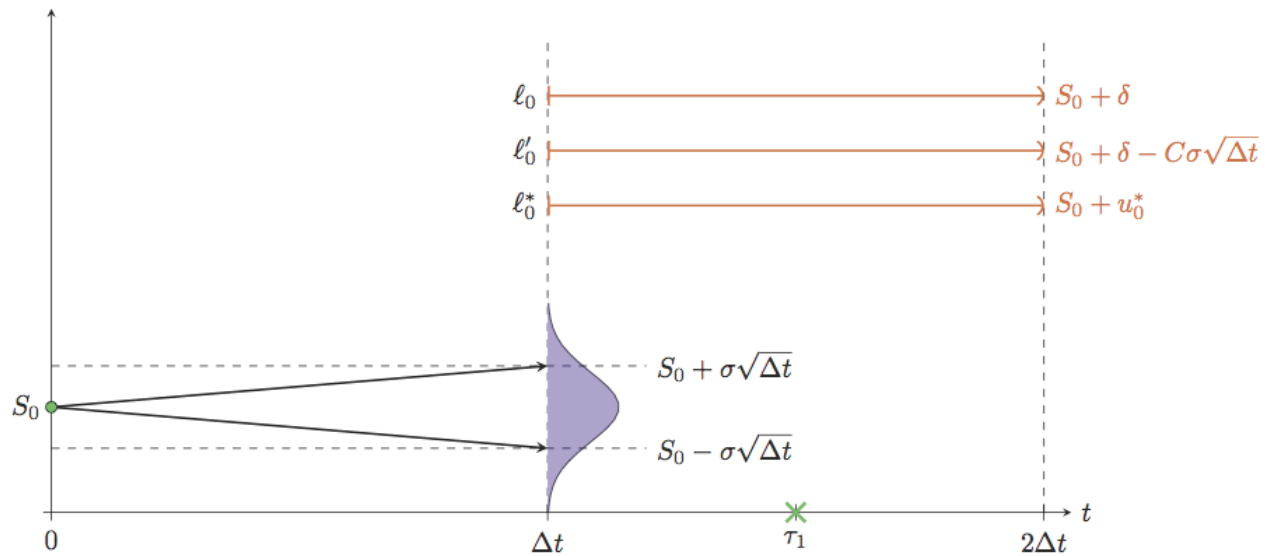


Figure 4: An illustration of the optimal policy of Theorem 2. In the absence of latency, at time $t = 0$, the trader would set the limit price at a premium of δ , i.e., $\ell_0 = S_0 + \delta$. In an environment with latency, the trader might set the limit price to be ℓ'_0 , which lowers ℓ_0 by an additional safety margin of C standard deviations. This serves to increase the likelihood of trade execution in the interval $(\Delta t, 2\Delta t)$. The optimal limit price ℓ_0^* utilizes a safety margin that is slightly larger.

The probability of a fill

- If the limit price $\ell_0 = S_0 + \delta$, the probability of execution (conditional on the arrival of an impatient trader) is

$$\begin{aligned}
 \Pr(\ell_0 \leq S_1 + \delta) &= \Pr(S_0 + \delta \leq S_0 + \sigma \sqrt{\Delta t} Z + \delta) \\
 &= \Pr(Z \geq 0) \\
 &= \frac{1}{2}
 \end{aligned}$$

where $Z \sim N(0, 1)$.

- If the limit price $\ell_0 = S_0 + \delta - C \sigma \sqrt{\Delta t}$, the probability of execution (conditional on the arrival of an impatient trader) is

$$\begin{aligned}\Pr(\ell_0 \leq S_1 + \delta) &= \Pr(S_0 + \delta - C \sigma \sqrt{\Delta t} \leq S_0 + \sigma \Delta t Z + \delta) \\ &= \Pr(Z \geq -C) > \frac{1}{2}\end{aligned}$$

Value function and dynamic programming

Define the value function J_{t_i} at time t_i as

$$J_{t_i}(S_{t_i}, \ell_{i-1}) := \max_{\ell_i, \dots, \ell_{n-1}} \mathbb{E} [P | S_{t_i}, \mathcal{E}_{t_i}, \ell_{i-1}] - S_{t_i}$$

where \mathcal{E}_{t_i} denotes the event that the trader's limit order hasn't been executed till time t_i .

J satisfies the Bellman equation

$$J_{t_i}(S_{t_i}, \ell_{i-1}) = \max_{\ell_i} \mathbb{E} [\mathbf{1}_{\mathcal{A}_1}(\ell_{i-1} - S_{t_i}) + \mathbf{1}_{\mathcal{A}_2}(S_{t_{i+1}} - S_{t_i}) + \mathbf{1}_{\mathcal{A}_3} J_{t_{i+1}}(S_{t_{i+1}}, \ell_i) | S_{t_i}, \mathcal{E}_{t_i}, \ell_{i-1}],$$

where recall that \mathcal{A}_1 denotes the event that the limit order at price ℓ_{i-1} is lifted by an impatient buyer at time t_i , \mathcal{A}_2 the event that the limit order is lifted by the market bid, and \mathcal{A}_3 the event that the limit order remains unfulfilled till t_{i+1} .

The terminal condition for value function is $J_{t_n}(S_{t_n}, \ell_{n-1}) = 0$. In principle, one can solve the Bellman equation for the value function J and the optimal strategy ℓ^* by backward induction.

Heuristic solution for small Δt

- We can get very close to the optimal solution by greedily maximizing the probability of interacting with an impatient buyer at each step.
- With $\ell_0 = S_0 + u$, The optimal value u^* of u is then given by

$$u^* = \arg \max_u \mathbb{E} [\mathbf{1}_{\mathcal{E}} (S_0 + u - S_1) | S_0, \ell_0 = S_0 + u]$$

where \mathcal{E} is the event that $S_1 \leq \ell_0 = S_0 + u \leq S_1 + \delta$.

- If $S_1 > S_0 + u$, the limit order becomes marketable at time t_1 and is executed at the bid.
- If $S_0 + u > S_1 + \delta$, the limit order cannot be filled by an impatient buyer.

Note

This is equivalent to maximize the conditional expected profit of selling one limit order at time t_1 , conditioned on the arrival of an impatient buy order at t_1 .

Formulation of solution

By assumption $S_1 - S_0 = \sigma \sqrt{\Delta t} z$ with $z \sim N(0, 1)$. Let

$$x = \frac{u}{\sigma \sqrt{\Delta t}}; \quad \Delta = \frac{\delta}{\sigma \sqrt{\Delta t}}.$$

Then

$$\mathbb{E} [(S_0 + u - S_1) \mathbf{1}_{\mathcal{E}}] = \sigma \sqrt{\Delta t} \int_{x-\Delta}^x (x - z) N'(z) dz.$$

Define the dimensionless ratio $\theta^* = u^*/\delta$. Then

$$\theta^* = \arg \max_x \frac{1}{\Delta} \int_{x-\Delta}^x (x-z)N'(z) dz.$$

- Note that θ^* depends only on Δ which in turn depends only on latency expressed in the natural timescale $(\delta/\sigma)^2$.

Code to compute $\theta^*(\Delta)$

```
In [2]: h <- function(x, del){
  dnorm(x) - dnorm(x - del) + x*(pnorm(x) - pnorm(x - del))
}

theta.raw <- function(del){
  hh <- function(x){-h(x, del)}
  res <- optimize(hh, interval=c(0, del))
  res$minimum/del
}

theta <- function(del){sapply(del, theta.raw)}
```

```
In [3]: library(repr)
options(repr.plot.height=5)

curve(theta(x), from=0.01, to=100, col="red", ylab=expression(theta(Delta)), xlab=expression(Delta))
```

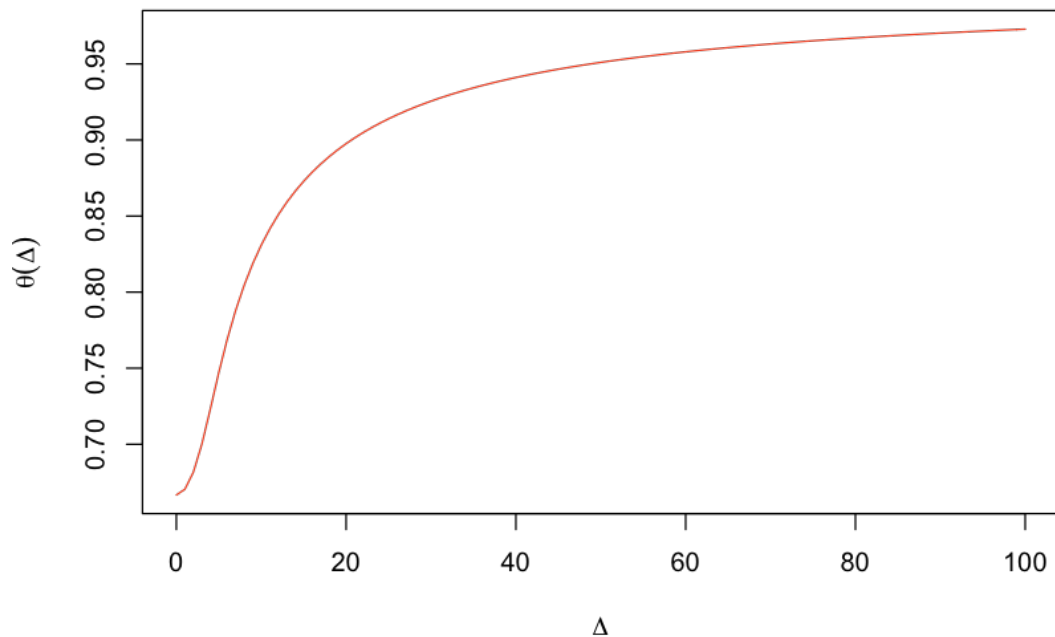


Figure 1: Graph of optimal limit order level $\theta^*(\Delta)$ vs $\Delta = \delta/\sigma \sqrt{\Delta t}$

Cost of latency

The proportion of profitability lost due to latency is given by $CL := 1 - \theta^*(\Delta)$. We can redo the plot in terms of latency τ in normalized units (which is just $1/\Delta^2$).

```
In [4]: del <- seq(1, 100, .1)
```

```
x <- 1/del^2  
y <- 1 - theta(del)
```

```
In [5]: plot(x,y,type="l",col="red",ylab="Cost of latency",xlab="Latency")
```

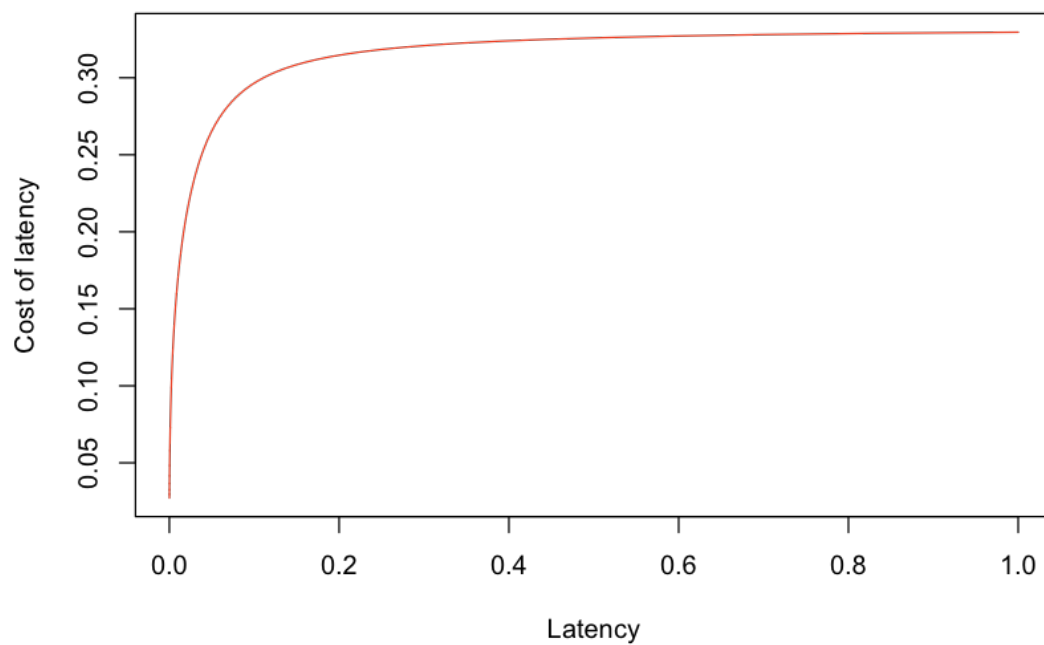


Figure 2: Cost of latency vs normalized latency τ

Solution

Define

$$\begin{aligned} f(x) &= \int_{x-\Delta}^x (x-z)N'(z) dz \\ &= x [N(x) - N(x-\Delta)] + N'(x) - N'(x-\Delta). \end{aligned}$$

The maximum is found when $f'(x) = 0$.

$$\begin{aligned} f'(x) &= N(x) - N(x-\Delta) \\ &\quad + x [N'(x) - N'(x-\Delta)] + N''(x) - N''(x-\Delta) \\ &= N(x) - N(x-\Delta) + x [N'(x) - N'(x-\Delta)] \\ &\quad - x N'(x) + (x-\Delta) N'(x-\Delta) \\ &= N(x) - N(x-\Delta) - \Delta N'(x-\Delta). \end{aligned}$$

Formally

$$\theta^* = \{x : N(x) - N(x-\Delta) - \Delta N'(x-\Delta) = 0\}.$$

We are interested in the limit $\Delta \rightarrow \infty$. Define $y = \Delta - x$. Then

$$f'(x) \sim 1 - N(-y) - \Delta N'(-y) = 0.$$

For this equation to have a solution, we must have

$$\Delta N'(-y) \sim \Delta e^{-y^2/2} \sim 1$$

from which it follows that $y \sim \sqrt{\log \Delta}$.

Then $N(-y) \rightarrow 0$ as $\Delta \rightarrow \infty$ and

$$\Delta N'(-y) = \Delta \frac{e^{-y^2/2}}{\sqrt{2\pi}} \approx 1$$

which has the solution

$$y \approx \sqrt{2 \log \frac{\Delta}{\sqrt{2\pi}}} = \sqrt{\log \frac{\Delta^2}{2\pi}}.$$

Define the normalized latency $\tau = \sigma^2 \Delta t / \delta^2 = 1/\Delta^2$. By definition of the cost of latency,

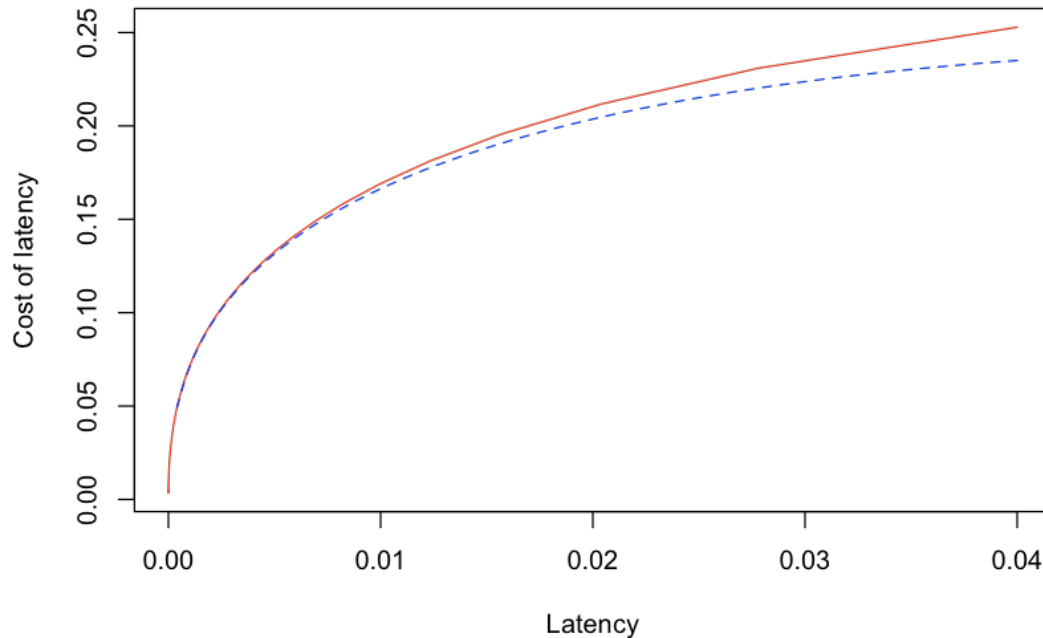
$$CL = \frac{y}{\Delta} \approx \sqrt{\tau} \sqrt{-\log 2\pi\tau} = \sqrt{-\tau \log 2\pi\tau}$$

which coincides with the result of [Moallemi and Sağlam]^[6].

Approximation quality

```
In [9]: latencyCostApprox <- function(tau){sqrt(-tau*log(2*pi*tau))}
```

```
In [10]: del <- 5:1000
tau <- 1/del^2; lc <- 1-theta(del)
plot(tau,lc,type="l",col="red",ylab="Cost of latency",xlab="Latency")
curve(latencyCostApprox(x),from=0,to=0.04,col="blue",lty=2,add=T)
```



Code to compute approximation given latency, spread and volatility

```
In [12]: tau <- function(latency.ms, spread, price, vol){ # latency in milliseconds
  latency.annualized <- latency.ms/1e3/60/60/24/252
  vol^2*latency.annualized*price^2/spread^2
}
```

The following code fragment takes latency in milliseconds and other stock statistics to estimate latency cost.

```
In [13]: tau.norm <- tau(latency.ms=200, spread=0.15, price=943, vol=.1544)
latencyCostApprox(tau.norm)
0.158746693757444
```

Numerical estimates of cost of latency

| | Microseconds | Years | | | | | | |
|---------|--------------|-------------|-----------|------------|-------------|-------|------------|-----------|
| Latency | 10 | 1.69584E-12 | | | | | | |
| | | | | | | | | |
| Ticker | Spread | Price | Vol (50D) | Sigma | Tau | LC | LC (100ms) | LC (10ms) |
| MSFT | \$0.01 | \$95.36 | 30.9% | \$0.000038 | 1.47242E-05 | 0.77% | 7.05% | 12.34% |
| C | \$0.01 | \$69.96 | 24.7% | \$0.000023 | 5.06381E-06 | 0.48% | 15.87% | 8.71% |
| IBM | \$0.03 | \$145.60 | 29.3% | \$0.000056 | 3.42925E-06 | 0.40% | 15.12% | 7.56% |
| BAC | \$0.01 | \$30.39 | 28.2% | \$0.000011 | 1.2455E-06 | 0.25% | 11.74% | 5.12% |
| PFE | \$0.01 | \$36.86 | 18.3% | \$0.000009 | 7.71608E-07 | 0.20% | 10.07% | 4.23% |
| GOOG | \$0.96 | \$1,076.76 | 29.5% | \$0.000414 | 1.85662E-07 | 0.10% | 5.99% | 2.33% |
| NM | \$0.0065 | \$0.8327 | 81.8% | \$0.000001 | 1.86226E-08 | 0.04% | 2.34% | 0.86% |

Stoikov and Waeber

- In a more recent paper, [Stoikov and Waeber]^[7] explore a similar idea but in a much more realistic setting.

- It is widely known that the order book (or imbalance) signal

$$I_t = \frac{B_t}{B_t + A_t}$$

is a good predictor of future price movement over the short term (see Lecture 1).

- In particular, market participants tend to send market orders only when I_t is close to 0 or 1.

- The question is to what extent latency reduces the value of this signal.
- In passing, Stoikov and Waeber show how to backtest an algorithm using only Level-1 data.

Order imbalance just before a trade

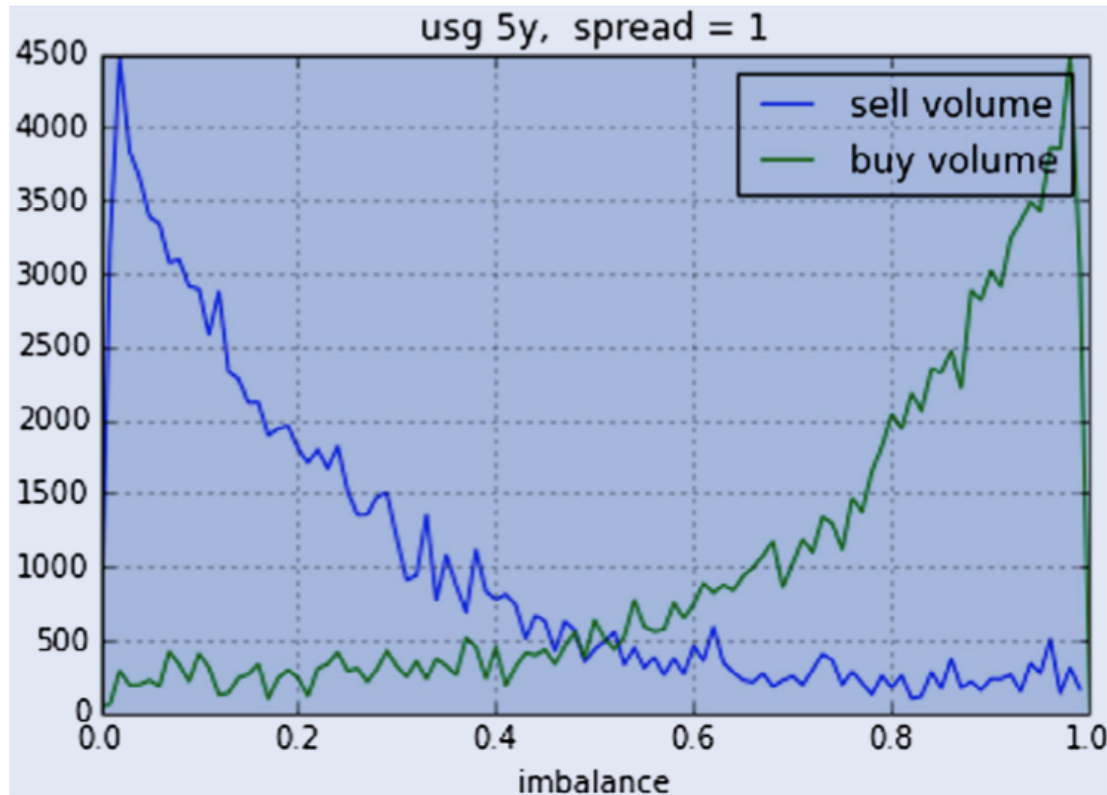


Figure 1. Histograms of imbalances I_t , the millisecond before a (sell or buy) trade arrives at the exchange.

Model setup

- The current bid price P_t and order imbalance I_t are jointly Markov. Price increments are independent of the current price.
- Given our system latency L , our objective is to compute a stopping time τ that maximizes our expected sale price for one share in the interval $(t, T]$ conditional on the current order imbalance I_t . That is

$$\sup_{t \leq \tau \leq T-L} \mathbb{E}[P_{\tau+L} - P_t | I_t]$$

- Define the payoff at the latest possible time $t = T - L$:

$$g^L(x) = \mathbb{E}[P_T - P_{T-L} | I_{T-L} = x] = \mathbb{E}[P_L - P_0 | I_0 = x]$$

by the Markov property.

- The function g^L thus quantifies the loss of benefit of the signal I_t as a function of latency L .

- The following figure shows that empirically (on data from the 5-year Treasury market in July 2010), as expected, g^L grows dramatically with L .

The effect of latency as a function of I_t

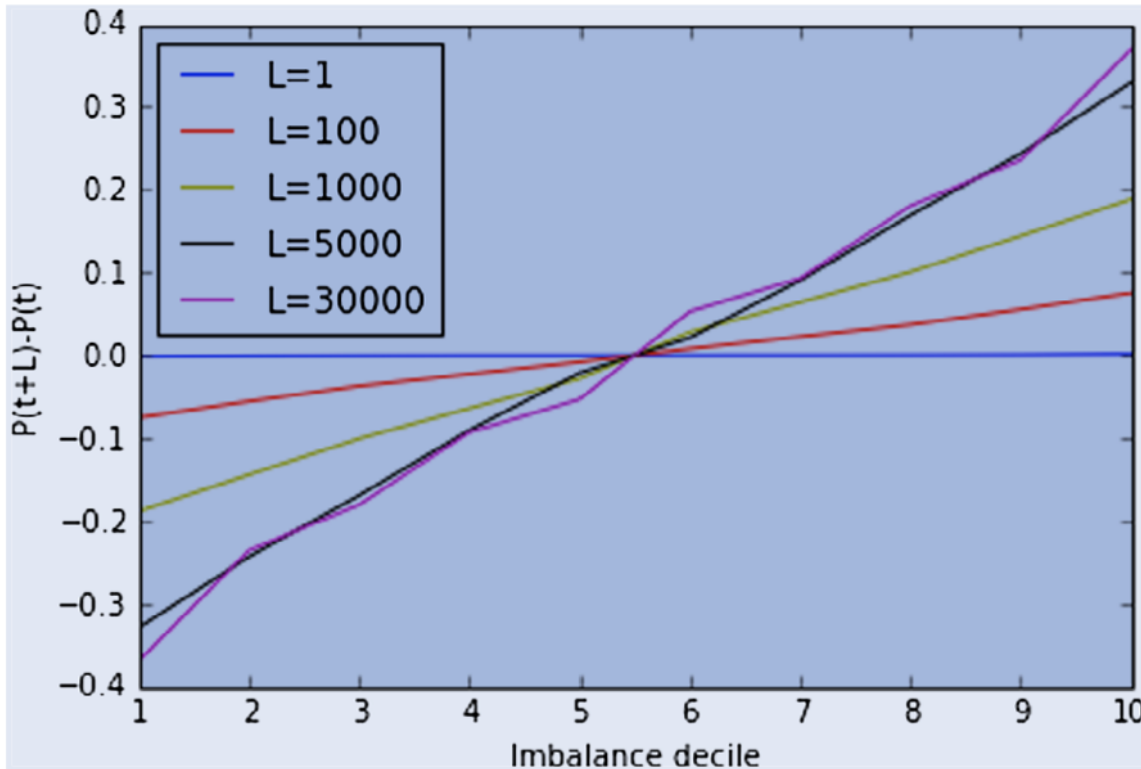


Figure 2. The function $g^L(x)$ expressed in fractions of the bid-ask spread. The latency L is in milliseconds.

Solution of the optimization problem

- To solve the problem, prices P_t are assumed to be on a tick grid and imbalances I_t are quantized (by computing quantiles) so that $I_t \in \{1, \dots, M\}$ for some M .

- The value function for this discretized problem

$$V^L(0, i) = \sup_{0 \leq \tau \leq T-L} \mathbb{E}[P_{\tau+L} - P_0 | I_0 = i].$$

is found by solving the Bellman equation

$$V^L(n, i) = \max \{ G^L(i), \mathbb{E}[V^L(n+1, I_{n+1}) | I_n = i] \}$$

where

$$G^L(i) = \mathbb{E}[P_L - P_0 | I_0 = i]$$

is the quantized immediate payoff function.

- To compute the conditional expectation, we need the transition probabilities. These can be estimated empirically.
- Just as in the case of optimal exercise of American options, there are trade and no-trade regions.
 - Don't trade if $G^L(i) \leq V^L(i)$.
 - Trade if $G^L(i) = V^L(i)$.

Backtesting the algorithm

Stoikov and Waeber address market impact and latency of real orders as follows.

- The backtest uses market orders only.
- The imbalance-based algorithm IMB is compared with TWAP (both of which presumably create similar market impact).
- $T = 1$ minute; both IMB and TWAP are required to liquidate one share per minute.
- An order submitted at time τ is executed at the best price available at time $\tau + L$.

Backtesting results

Table 1. The performance of the optimal imbalance-based algorithm over TWAP for various values of T and L . The values are expressed as percentages of the minimum bid ask spread.

| T | $L = 1$ ms | $L = 100$ ms | $L = 1000$ ms | $L = 5000$ ms |
|-------|------------|--------------|---------------|---------------|
| 10 s | 0.20 | 0.17 | 0.11 | 0.05 |
| 1 min | 0.32 | 0.26 | 0.17 | 0.07 |
| 5 min | 0.33 | 0.35 | 0.18 | 0.07 |

- We see that as expected, the value of incorporating the imbalance signal declines as latency increases.

Order routing

- There are currently approximately 13 lit venues and over 40 dark venues in the US.
 - On the one hand, it would be prohibitively complicated and expensive to route to all of them.
 - On the other hand, by not routing to a particular venue, the trader misses out on potential liquidity, and all things being equal, will incur greater market impact.
- Most traders have to use a smart order routing (SOR) algorithm provided by a dealer.

Turnover by lit venue (from Fidessa)**Lit Venue Turnover**

| Venue | Value(USD) | Volume | Trades | Share | |
|------------------------|-----------------|---------------|------------|---------------|--|
| NASDAQ | 241,776,690,749 | 4,091,919,679 | 32,632,306 | 21.16% | |
| NYSE | 169,559,622,848 | 3,908,600,661 | 12,132,519 | 20.21% | |
| NYSE Arca | 145,902,630,554 | 2,732,157,581 | 15,528,583 | 14.13% | |
| BZX | 115,073,773,451 | 2,159,236,783 | 16,778,960 | 11.17% | |
| EDGX | 104,994,030,730 | 2,152,442,201 | 13,206,086 | 11.13% | |
| BYX | 55,817,525,493 | 1,569,580,944 | 12,332,809 | 8.12% | |
| NASDAQ BX | 46,075,857,524 | 1,011,761,239 | 9,546,556 | 5.23% | |
| IEX | 39,211,702,811 | 735,126,180 | 4,882,967 | 3.80% | |
| EDGA | 21,302,022,165 | 434,835,920 | 3,311,723 | 2.25% | |
| NASDAQ PHLX | 15,069,496,897 | 240,687,033 | 1,686,897 | 1.24% | |
| Chicago Stock Exchange | 16,701,849,120 | 198,267,446 | 42,511 | 1.03% | |
| NYSE MKT | 2,871,628,016 | 101,031,423 | 586,008 | 0.52% | |
| Unlisted | 10,131 | 25,000 | 15 | 0.00% | |

List of dark pools (from Wikipedia)

****Independent dark pools****

- Chi-X Global
- Instinet
- Liquidnet
- NYFIX Millennium
- Posit/MatchNow from Investment Technology Group (ITG)
- State Street's BlockCross
- RiverCross Securities
- SmartPool
- TORA Crosspoint
- ETF One
- Codestreet Dealer Pool for Corporate Bonds

****Broker-dealer-owned dark pools****

- JPMorgan Chase Bank - JPMX
 - Barclays Capital - LX Liquidity Cross
 - BNP Paribas - BNP Paribas Internal eXchange (BIX)
 - BNY ConvergeX Group (an affiliate of Bank of New York Mellon)
 - Cantor Fitzgerald - Aqua Securities
 - Citi - Citi Match, Citi Cross
 - Credit Agricole Cheuvreux - BLINK
 - Credit Suisse - CrossFinder
 - Deutsche Bank Global Markets - DBA (Europe), SuperX ATS (U.S.)
-
- Fidelity Capital Markets
 - GETCO - GETMatched
 - Goldman Sachs SIGMA X
 - Knight Capital Group - Knight Link, Knight Match
 - Deutsche Bank Global Markets - DBA (Europe), SuperX ATS (U.S.)
 - Merrill Lynch - Instinct-X
 - Morgan Stanley - MSPOOL
 - Nomura - Nomura NX
 - UBS Investment Bank - UBS ATS, UBS MTF, UBS PIN
 - Société Générale - ALPHA Y
 - Daiwa - DIRECT

List of dark pools (from Wikipedia)

You can see the whole list here:

https://en.wikipedia.org/wiki/Dark_liquidity (https://en.wikipedia.org/wiki/Dark_liquidity)

Smart order routing (SOR)

- The goal of an SOR algorithm is to buy (or sell) as many shares as possible in the shortest time by optimally allocating orders across both lit and dark venues.
 - In the case of lit venues, there are hidden orders so there is typically more liquidity available than is displayed.
 - Where to route an order involves not just visible liquidity but also expected costs of execution.
 - In dark venues, by definition, all liquidity is hidden.

- We will describe
 - A heuristic algorithm due to Almgren and Harts
 - An algorithm due to Charles-Albert Lehalle and his collaborators
 - An algorithm based on machine learning techniques due to Michael Kearns and his collaborators.

The Almgren and Harts (AH) algorithm

- The idea of this algorithm is that the more hidden quantity is detected in a given venue, the more hidden quantity there is likely to be.
 - Recall that this is a characteristic of distributions with fatter tails than exponential.
 - Also, we believe in power-law distribution of quantity in which case this assumption would definitely be justified.
- For simplicity, let's focus on the sale of stock.
- If hidden quantity w is detected (by selling more than the visible quantity) on a particular venue, the current estimate of hidden liquidity is increased by w .
- If no hidden quantity is detected on a venue, the existing estimate is decremented by a factor ρ .

Conditional distribution of quantity: Exponential case

Suppose that the distribution of order sizes Q is exponential so that

$$\Pr(Q > n) = \frac{1}{\lambda} e^{-\lambda n}$$

Assuming the conditional probability that hidden quantity is greater than n given that n slices have already been observed is given by

$$\Pr(Q \geq (n+1) | Q \geq n) = \frac{\Pr(Q \geq (n+1))}{\Pr(Q \geq n)} = e^{-\lambda}$$

If the distribution of Q is exponential, the conditional probability of more quantity is independent of the quantity already observed.

Conditional distribution of quantity: Power-law case

Suppose that the distribution of order sizes Q is power-law so that

$$\Pr(Q > n) = \frac{C}{n^\alpha}$$

Assuming the conditional probability that hidden quantity is greater than n given that n slices have already been observed is given by

$$\begin{aligned} \Pr(Q \geq (n+1) | Q \geq n) &= \frac{\Pr(Q \geq (n+1))}{\Pr(Q \geq n)} \\ &= \left(\frac{n}{n+1} \right)^\alpha \\ &\rightarrow 1 \text{ as } n \rightarrow \infty \end{aligned}$$

If the distribution of Q is power-law, the conditional probability of more quantity tends to 1 as the quantity observed increases.

The AH algorithm

Our goal is to execute a sell order as quickly as possible by optimally allocating quantity to all N venues. To avoid eating into the book, we send IOC ("immediate or cancel") orders only. We will simplify the Almgren-Harts algorithm by assuming no liquidity above the displayed best bid.

- We allocate quantity quasi-greedily sequentially to the venue with the highest estimated quantity, visible and hidden.
- If we see a fill of size n_j when the displayed quantity is q_j on the j th venue, the pre-existing liquidity estimate R_j is decayed by a factor ρ and incremented by the detected hidden liquidity:

$$R_j \mapsto \rho R_j + (n_j - q_j)^+$$

- Repeat until our quantity is exhausted and the order is completed.

A stochastic Lagrangian algorithm

[Laruelle, Lehalle and Pagès]^[4] propose the following way of looking at optimal allocation to dark pools.

- Let be V be the total volume to be executed
- For the i th dark pool, ρ_i is the relative cost saving, D_i is the quantity available, and r_i is the proportion of the order sent there.
- Then the cost of a buy order is given by

$$C = S \left\{ V - \sum_{i=1}^N \rho_i \min(r_i V, D_i) \right\}$$

where the $\rho_i \in (0, 1)$.

- The optimization problem is then

$$\max_{r_i} \left\{ \sum_{i=1}^N \rho_i \mathbb{E} [\min(r_i V, D_i)] : \sum_{i=1}^N r_i = 1, r_i \geq 0 \forall i \right\}.$$

The mean execution function

Given a dark pool i , define its *mean execution function*

$$\varphi_i(r) := \rho_i \mathbb{E} [\min(r V, D_i)].$$

Then φ_i is finite, nonzero, and concave in r . Also

$$\varphi'_i(r) = \rho \mathbb{E} [\mathbb{1}_{\{r V < D_i\}} V].$$

The optimization problem becomes

$$\max_{r_i} \left\{ \sum_{i=1}^N \varphi_i(r_i) : \sum_{i=1}^N r_i = 1 \right\}$$

Solution

Introducing a Lagrange multiplier λ , this problem is equivalent to

$$\max_{r_i} \left\{ \sum_{i=1}^N \varphi_i(r_i) - \lambda \left(\sum_{i=1}^N r_i - 1 \right) \right\}.$$

At the optimal point r_i^* , we must have

$$\varphi'_i(r_i^*) - \lambda = 0 \quad \forall i \quad \implies \quad \varphi'_i(r_i^*) = \frac{1}{N} \sum_{i=1}^N \varphi'_i(r_i^*) =: \bar{\varphi}'.$$

The stochastic Lagrangian algorithm

This leads naturally to the suggested algorithm

Stochastic Lagrangian (SL) algorithm

$$\Delta r_i^{n+1} = r_i^{n+1} - r_i^n = \gamma_n \{ \varphi'_i(r_i^n) - \bar{\varphi}'_n \}$$

- More quantity is sent to dark pools with higher liquidity D_i , as observed in the history of executions.
- More quantity is sent to the dark pools offering lower-cost execution (*i.e.* greater ρ_i).
- With various technical assumptions, [Laruelle, Lehalle and Pagès]^[4] prove the convergence of this algorithm.

Estimation of the ρ_i

- How would one go about estimating the cost saving (if any) associated with execution in a dark pool (or in a lit venue)?
- The total cost comprises commissions, fees and rebates and most importantly, adverse selection.
 - Commissions, fees and rebates vary widely between exchanges.
 - Adverse selection can be proxied by measuring the realized spread: Where is the mid-price (for example) one minute after an execution?

Discussion of [Laruelle, Lehalle and Pagès]^[4]

- The fact that the algorithms of [Laruelle, Lehalle and Pagès]^[4] explicitly take into account the perceived relative advantages of executions in the various venues is a major strength of their algorithms.
- However, they make no assumption on the distribution of quantity in the dark pool – we know empirically that this is fat-tailed.
 - If you submit r_i V and your order is filled in its entirety, there is very likely to be more quantity available.

The Kearns et al. (GKNW) algorithm

- The idea behind the GKNW algorithm is not dissimilar to the ideas behind the previous algorithms although, as written, it is applied only to dark pools.
- In the *allocation* phase, orders are allocated greedily to the venue with the greatest estimated liquidity.
 - The greedy algorithm is explained in detail in the GKNW paper and is proved to be optimal in some precise sense.
- In the *re-estimation* phase, the estimated tail quantities of the order-size distributions are updated according to a specified rule.
- Allocation and re-estimation are performed in a continuous loop.

The greedy allocation algorithm

- Let $P_j(q)$ be the probability of there being q shares at the j th venue. Define

$$T_j(q) = \sum_{n \geq q} P_j(n)$$

to be the j th tail probability. The empirical estimate of T_j is denoted by \hat{T}_j .

- If \mathbf{v} is the allocation vector of the total quantity V to the various venues, allocate as follows:
 - Set $\mathbf{v} = \mathbf{0}$;
 - While $\sum v_i < V$, $i = \arg \max_i \hat{T}_i(v_i + 1)$, $v_i = v_i + 1$;
 - Return \mathbf{v} .

The Kaplan-Meier estimator

- If an order is filled in its entirety, we know only that there was at least that much quantity available.
 - In statistics, this is called a *censored* observation.
- Let $z(s) = \Pr(q = s | q \geq s)$ be the conditional probability that there are exactly s orders given that the quantity available is at least s . Then

$$\begin{array}{l} T_j(q) = \Pr(n \geq q) = \prod_{s=0}^{q-1} \frac{\Pr(n \geq (s+1))}{\Pr(n \geq s)} = \prod_{s=0}^{q-1} (1 - z(s)) \\ \Pr(n > s | n \geq s) = \prod_{s=0}^{q-1} (1 - z(s)) \end{array}$$

The Kaplan-Meier estimator

The Kaplan-Meier estimator is then

$$\hat{T}_j(q) = \prod_{s=0}^{q-1} (1 - \hat{z}(s))$$

where

$$\hat{z}(s) = \frac{D_s}{N_s}$$

is an empirical estimate of the probability that the available quantity is exactly s .

D_s is the number of direct (uncensored) observations and N_s is the total number of observations (censored and uncensored) that could potentially have been s . (In medicine, D_s typically represents deaths at time s and N_s represents the number of patients still in the study at time s).

Modifying Kaplan-Meier

- [Kearns et al.]^[3] derive a cut-off point c_j for each venue such that for $s < c_j$, the tail probability estimate $\hat{T}_j(s)$ is guaranteed to be close to the true probability $T_j(s)$.
- In an *Exploitation Lemma* they show that if the allocation to the j th venue is less than c_j , there is sufficient data to estimate the tail probability and the allocation is provably optimal.
 - In this case, the Kaplan-Meier estimate is used as $\hat{T}_j(s)$ in the allocation algorithm, without modification.

- If $s > c_j$, the tail probability is optimistically estimated as $\hat{T}_j(s) = \hat{T}_j(c_j)$.
 - By making this “optimistic” modification to the K-M estimate, the authors ensure that no venue remains unexplored simply because the algorithm has only a few observations of small quantity from that venue.

The main result

Theorem

For any $\epsilon > 0$ and $\delta > 0$, with probability $1 - \delta$, after running for a time polynomial in N , V , $1/\epsilon$ and $\log 1/\delta$, the algorithm makes an ϵ -optimal allocation on each subsequent time step with probability at least $1 - \epsilon$.

- Recall that N is the number of venues and V is the total quantity to be traded.

Parametric quantity distributions

- In practice, the Kaplan-Meier estimate of tail probability does very well in-sample but not surprisingly not very well out-of-sample due to serious over-fitting.
- By splitting a large sample of dark-pool trades into a training set and a test set, [Kearns et al.]^[3] were able to show that the parametric form ZB+power-law works best.
 - That is, one estimates the probability of zero quantity separately (most of the time there is zero quantity), and fits a power-law of the form $1/s^\beta$ to the rest of the data.
 - β is found to lie mostly between 0.25 and 1.3. Compare with our previous stylized fact that quantity is distributed according to a 3/2 law.

A simulation study

- As is usual with these problems, an algorithm can only be tested by experiment or simulation.
 - The data comes from particular choices of algorithm and we can't predict what would have been if the algorithm had chosen to act differently.
- It being impossible to experiment, the authors chose simulation.
- Four algorithms are compared:
 - Ideal allocation from knowing the true distributions used for the simulation.
 - Equal allocation across venues
 - The GKNW learning algorithm with ZB+Power Law parameterization
 - A “bandit” algorithm that begins with equal weights. If there is any execution at a particular venue, that venue's weight is increased by a factor $\alpha = 1.05$.

Simulation results

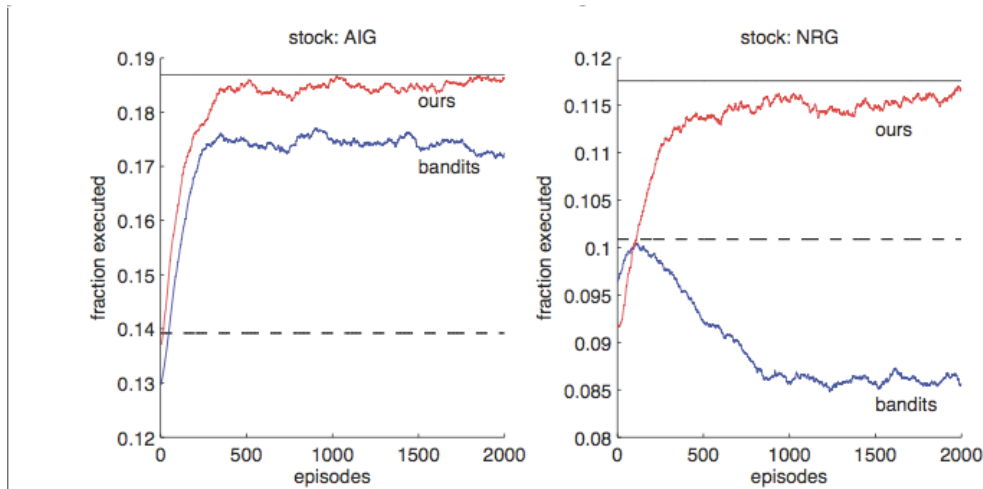


Figure 1: Sample learning curves. For the stock AIG (left panel), the bandits algorithm (labeled blue curve) beats uniform allocation (dashed horizontal line) but appears to asymptote short of ideal allocation (solid horizontal line). For the stock NRG (right panel), the bandits algorithm actually deteriorates with more episodes, underperforming both uniform and ideal allocation. For both these stocks (and for the other 10 in our data set), our algorithm (labeled red curve) performs nearly optimally.

Why do new venues attract any order flow?

- Once a venue has volume, it's clear to see that it is optimal to route some part of your order flow to that venue.
- Why does the new venue attract any flow to start with?
- Part of the answer might lie in the use of algorithms like GNKW.
 - A machine-learning algorithm will always allocate a minimum amount of flow to a given venue; otherwise that venue will remain unexplored.

Cont and Kukanov: optimal limit order placement

- Assume a buy order
- Venues $k = 1, \dots, K$ are characterized by
 - Bid queue lengths Q_k
 - Make rebates r_k , take fees f_k , half-spread $h = s/2$.
- Simplify by assuming that any market order is sent to the venue with the smallest take fee f .

- Order placement strategy is $X = (M, L_1, \dots, L_K)$ where the components represent volumes of the single market order and the k th limit orders respectively.
- The tradeoff is between cost and execution probability.
 - On a given exchange, the lower the price of the limit order, the lower the probability of execution
 - On multiple exchanges, the higher the rebate, the longer the queue length, the lower the probability of execution.

Solution for single venue (see Lecture 2)

- If λ_u is in a suitable range,

$$L^* = F^{-1} \left(\frac{2h + f + r}{\lambda_u + h + r} \right) - Q, \quad M^* = \tilde{X} - L^*.$$

where $F(\cdot)$ denotes the distribution of ξ (which is increasing in ξ).

- Comparative statics. L^* is:
 - decreasing in Q ,
 - increasing in h ,
 - increasing in f ,
 - decreasing in λ_u ,
 - increasing in r .
- As the target size \tilde{X} increases, L^* is fixed and M^* increases.
- $F(\cdot)$ may be estimated in real time using recent outflow data (and potentially other signals).
 - $F(\cdot)$ need not be parametric.

Solution for multiple venues

- Let $\mathcal{A}(X^*, \xi)$ denoted executed quantity. When the allocation is optimal,

$$\begin{array}{l} \mathbb{P}(\left(X^{\wedge}, \xi \right) < \tilde{X} \mid \xi_j > Q_j + L_j^*) = \frac{h + f + \lambda_o}{\lambda_u + \lambda_o} \\ \mathbb{P}(\left(X^{\wedge}, \xi \right) < \tilde{X} \mid \xi_j < Q_j + L_j^*) = \frac{\lambda_o}{h + f + \lambda_o} \end{array}$$

- Note that the bigger the rebate r_j on a given exchange, the lower the conditional shortfall probability.
 - This can be used to define λ_u and λ_o in terms of maximal under-fill probabilities.
- One interesting insight from the solution is that execution cost is lower with multiple venues relative to a single venue if the outflows ξ_k are sufficiently uncorrelated.
 - This amounts to a condition for optimality of order-flow fragmentation.

Summary

- The cost of latency can be estimated using a simple formula that uses only volatility and spread.
 - The benefit of extra technology investment can be estimated on the back of an envelope.
- Stoikov and Waeber apply this idea to a realistic strategy based on order imbalance and further show how to backtest their algorithm using only Level-1 data.

- We presented various smart order routing algorithms, some of which are provably optimal in some sense.
 - A better algorithm would incorporate the best features of the best of these algorithms (SL and AH?).
- [Cont and Kukanov]^[2] show how to optimally distribute limit orders and market orders between venues based on estimation of execution probabilities, rebates and fees.

A philosophical question

Question

Why should it be so complicated to trade stock?

One answer goes something like this:

- Changes in market structure together with technological innovation have massively reduced trading costs.
- Nevertheless, some market participants achieve significantly lower costs.
 - This requires either substantial investment in technology and trading expertise or
 - careful selection of broker algorithms.

Potential cost savings from optimal scheduling

With reasonable assumptions, assuming the square-root model, savings relative to a simple VWAP are roughly of the following order:

- Around 15% for the Alfonsi-Fruth-Schied bucket-like strategy
- Around 25% for a suitable bursty strategy (intermittent interval VWAPs) with a capped participation rate.

Potential cost savings from microtrader improvements

- Were we just to blindly send market orders, we would estimate that the cost of each child order would be around a half-spread.
- Practical experience shows that it is not possible to reduce this cost much below one third of a spread.
- We conclude that we could potentially save up to one sixth of the spread.

Potential cost savings from smart order routing

- A naïve estimate would be to use the square-root market impact formula, changing the denominator V to reflect the potential increase in liquidity from routing to extra venues.
 - If the potential liquidity accessed is doubled, costs should be decreased by $1 - 1/\sqrt{2} \approx 30\%$ according to this simple computation!
 - This could be one explanation for the multiplicity of trading venues in the US.
- We expect actual savings to be much less than this because the different venues are all connected and information leaks from one venue to the other.

Final conclusion

- Recent empirical and theoretical work on market microstructure has led to a much improved understanding of how to trade optimally.
- Potential savings from careful order execution relative to VWAP using a simple first-generation algorithm are substantial.
 - Relative to a naïve strategy, cost savings of 25% for reasonably sized orders are not unreasonable.

References

1. ¹Robert Almgren and Bill Harts, A dynamic algorithm for smart order routing, *White paper StreamBase* (2008).
2. ²Rama Cont and Arseniy Kukanov, Optimal order placement in limit order markets, *Quantitative Finance* **17**(1) 21-39 (2017).
3. ³Kuzman Ganchev, Michael Kearns, Yuriy Nevmyvaka and Jennifer Wortman Vaughan, Censored exploration and the dark pool problem, *Communications of the ACM* **53**(5) (2010).
4. ⁴Sophie Laruelle, Charles-Albert Lehalle, and Gilles Pagès, Optimal split of orders across liquidity pools: a stochastic algorithm approach, *SIAM Journal on Financial Mathematics* **2**(1) 1042–1076 (2011).
5. ⁵Costis Maglaras, Ciamac C Moallemi, and Hua Zheng, Queueing Dynamics and State Space Collapse in Fragmented Limit Order Book Markets,
6. ⁶Ciamac Moallemi and Mehmet Sağlam, The cost of latency, *Operations Research* **61**(5) 1070–1086 (2013).
7. ⁷Sasha Stoikov and Rolf Waeber, Reducing transaction costs with low-latency trading algorithms, *Quantitative Finance* **16**(9) 1445-1451 (2016).