streambase

StreamBase White Paper
Smart Order Routing

■ A Dynamic Algorithm for Smart Order Routing

By Robert Almgren and Bill Harts

A Dynamic Algorithm for Smart Order Routing

Robert Almgren and Bill Harts

1 The Smart Order Routing Problem

The US equity markets have become increasingly complex in recent years. In part due to regulatory changes such as decimalization and Reg NMS, and in part due to technology improvements such as fast data communication, the trader has a wide variety of choice of where to send each piece of a big order (see [Hasbrouck, 2007, Appendix] for a review). Furthermore, the human trader is increasingly replaced by an algorithmic trading engine, which makes routing decisions based on continuous real-time updating of its model of market liquidity. Doing this well is "smart order routing."

The problem is made vastly more complicated by the presence of "hidden liquidity:" limit orders resting at a market center whose existence is not disclosed to outside traders. These orders have the advantage (from the poster's point of view) of not disclosing the existence of a large order, but they usually receive lower priority for execution against an incoming market order than do displayed limit orders. A special case of hidden liquidity is "reserve" or "iceberg" orders, of which only a prespecified part is displayed, that refills immediately when consumed. In the extreme case, "dark pools" contain only hidden liquidity, potentially of very large size. If all liquidity were displayed, then smart order routing would be easy: the difficulty comes from trying to discern the presence of hidden liquidity and to use that information in making routing decisions.

The optimal use and detection of hidden liquidity has received substantial recent attention in the literature. Esser and Mönch [2007] estimate that iceberg orders represent 8.24% of total liquidity on Xetra in 2002. Tuttle

[2003] estimates that non-displayed size represents 25% of depth at the NBBO in the Nasdaq 100; De Winne and D'Hondt [2007] estimate that for CAC40 stocks in 2002, 45% of depth at the first five quotes is hidden; Pardo and Pascual [2006] estimate that 18% of all trades on the Spanish Stock Exchange in 2000 involved some execution against hidden liquidity.

Hidden liquidity is discovered only when orders execute against it. If a trade at a particular venue is reported, whose size is larger than the displayed size at that market just before the execution, then some of the execution must have been against hidden liquidity in addition to displayed; Pardo and Pascual [2006] and De Winne and D'Hondt [2007] give specific examples. The interest of these authors is in identifying backwards-looking signals to identify the presence of hidden liquidity, and to assess whether traders place more aggressive orders when they guess that hidden liquidity may be present. Our goal is more ambitious: we want to maintain a *dynamic* estimate of the hidden liquidity present on each of several different exchanges, and use this information to make routing decisions now and in the future.

Our position is similar to trying to estimate how many people are in a room that is hidden from view, and into which people may enter through a door that we do not see. All we see is a door which occasionally opens and through which one or more people occasionally emerge from the room. Each time we see some people emerge, we know that at least that many people were present in the room before, but we do not have any information about how many people are still there.

Our approach to estimating this problem is based on the simple principle that the more people we see coming out of the room, and the more often they come, the larger is our estimate of the number of people inside. This is somewhat counter-intuitive, since each person who comes out represents a decrease in the number inside, but since we do not try to model the process by which people enter, this is our only source of information. The other principle is that as time passes with no people coming out of the room, we continually decrease our estimate of the number inside. In fact, we measure time by the number of door-opening events: each time the door opens (whether a person comes out or not), we decrease our estimate of the number of people inside, and in addition we increase our estimate each time some people come out.

The parallel to estimation of hidden liquidity should be apparent. The number of people inside the room represents the size of hidden liquidity available on a particular trading center. Executions against hidden liquidity when market orders arrive corresponds to people coming out of the room. We

have no model for the submission of orders of hidden liquidity, corresponding to people entering the room. We measure time using "trading time:" the number of trade events which correspond to door opening.

Of course, like all analogies, this one is far from perfect. Real markets are far more complicated, and we have much more detailed information available, such as the price and timing of executions in addition to the mere fact of their existence. In particular, we should point out one misleading feature of this analogy: for people in a room we may imagine a sort of "pressure," which causes the rate at which people exit to be positively related to the number of people inside. In markets, unless some participants have special information, there is no particular reason to expect any relationship between the amount of hidden liquidity present at a market center and the arrival rate of market orders there. Nonetheless, this model captures the essential features of what we want to model and is a very useful starting point.

2 The Algorithm

We take the point of view of a trader, either human or algorithmic, given the task of executing a given number of shares at the best possible price, as quickly as possible. As the trader looks out over the landscape of possible destinations for these shares, he, she or it sees a wide variety of information. Some venues display liquidity in the form of limit orders resting in the book and available for immediate execution. Some of these may have additional liquidity that is in the book, but is revealed only when an incoming market order executes against it. Other venues may be completely "dark" with no displayed liquidity, only hidden. The trader's problem is to guess where this hidden liquidity lies, in order to send the new order where it has the greatest chance of being filled quickly and well.

2.1 Simple update formula

To begin with, let us focus on a single venue and a single side, let us say buy orders posted on the bid. Suppose that we see a series of trades executing at the bid prices, of sizes $S_1, S_2, \ldots$. For each execution S_n, let us denote

$$\begin{aligned} s_n &= \text{visible liquidity just before execution } n, \text{ and} \\ r_n &= \text{hidden liquidity just before execution } n. \end{aligned}$$

We know s_n, but r_n we can only estimate from the relationship

$$r_n \;\geq\; w_n \;\equiv\; \max\{\, S_n - s_n,\, 0 \,\}$$

Thus w_n is the quantity executed against hidden liquidity, which will be zero if the executed quantity is less than the visible quantity. Our task is to form some estimate $\hat{r}_{n+1}$ of the hidden liquidity remaining after the execution.

There are two problems with this construction:

1. It gives only a *lower bound* for r_n: we have no way of knowing whether extra liquidity was present that was not executed against. Any estimate $\hat{r}_n$ must only satisfy $\hat{r}_n \geq w_n$.

2. It only gives us information about the liquidity that was present in the market before the nth trade; it does not directly help us estimate the new hidden liquidity r_{n+1}. We might plausibly argue that $\hat{r}_{n+1} = \hat{r}_n - w_n$, but unless we construct a model for how liquidity replenishes, we will only decrease our estimate.

We propose the following two principles for maintaining and updating our estimate $\hat{r}_n$ of the size of hidden liquidity:

- Each time we observe an execution with $w_n > 0$, we *increase* $\hat{r}_n$ by w_n, the amount of the execution that can be attributed to hidden liquidity.

- As time passes, we decrease $\hat{r}_n$ toward zero using an exponential decay. It will be more appropriate to use trade time than clock time, so an equivalent statement would be that on each execution, whether $w_n = 0$ or $w_n > 0$, we multiply $\hat{r}_n$ by a fixed factor ρ with $0 < \rho < 1$.

Thus, our update formula in this simple case of one market is

$$\hat{r}_{n+1} \;=\; \rho\, \hat{r}_n \;+\; w_n.$$

Increasing the estimate by the trade size may seem counterintuitive, but it is the only simple way to take account of the available information. This algorithm will increase the estimate of reserve liquidity when we see large or frequent executions against hidden liquidity. In the example of people exiting a room, we decrease our estimate every time we see the door open, whether or not any person comes out; if the door opens many times with no exiting people, our estimate will trend downwards.

2.2 Application to real markets

Now let us address some of the complexities that we must add to make this simple model at all plausible for real smart order routing.

We suppose that there are N market centers or "venues," trading a single stock. Each one maintains an independent limit order book, consisting of limit bids and limit offers at a range of prices. At each center, the "best bid" is the highest bid that has nonzero displayed volume, and the "best offer" is the lowest offer that has nonzero displayed volume. There may be nondisplayed volume inside the best bid and the best offer, but nondisplayed volume cannot be crossed within a single venue. The "national best bid and offer" (NBBO) are the best displayed bid and displayed offer across all the venues; undisplayed volume may be crossed between different market centers.

These market centers may also be "dark pools," at which no liquidity is displayed; the same construction applies. Also, the hidden liquidity may be a reserve order, which refreshes the visible liquidity immediately after the visible part is consumed.

The problem of smart order routing is this: We are given the task of executing a single order as rapidly as possible. We must choose to which venue to send a single order for immediate execution. We could use a market order, expecting to get filled at the visible top of book based on visible limit orders. But this runs the risk of the limit order disappearing before our order arrives, leaving us to execute deeper in the book. In practice, we would use a marketable limit order, setting the limit order at the price at which we expect to execute. In addition, we will use the "immediate-or-cancel" (IOC) or "fill-or-kill" flag, so that unexecuted shares are returned to us rather than defining a new inside market.

An example is shown in Table 1. Our task is to route a sell order for 600 shares. We will use an IOC limit order at the national best bid of 20.01, to protect ourselves from executing below this value, and our only question is to which market we should send the order. Based on displayed liquidity, market A looks better since we will certainly get filled on 500 shares at 20.00; we may expect to find another 100 hidden shares and if not the order will be returned to us. But if we were able to detect the hidden liquidity, we could get much better execution at market B: 200 shares as 20.03 and the remaining 300 at 20.02. The whole question is how to guess the presence of this hidden liquidity.

	Market Center A				Market Center B			
	Bid		Offer		Bid		Offer	
Price	Displ.	*Hid.*	Displ.	*Hid.*	Displ.	*Hid.*	Displ.	*Hid.*
20.05			100	*200*			400	*400*
20.04			200	*100*				*200*
20.03				*200*		*200*		
20.02		*300*				*500*		
20.01	500	*200*			200	*100*		
20.00	200	*700*			100	*200*		

Table 1: Example composite order book. The NBBO is 20.01/20.04, based on displayed liquidity. The composite market is "crossed" based on hidden liquidity, which would not be possible on a single market. A sell market order of 600 shares would receive much better execution at venue B than A, despite the larger size of the displayed liquidity at A.

2.3 Liquidity discovery

Now we describe a simple algorithm for maintaining an estimate of the hidden liquidity available at each market venue, based on our observations of the public stream of trades and published quotes, as well as our own history of what happened when we have submitted orders. For concreteness, let us suppose that we are to execute a sell order, so that we are interested in the bid side of the market.

We consider only the market venues whose top of book is equal to the national best bid: we do not consider routing to venues who are not currently bidding at the national best level. Limit orders above that price level are necessarily entirely hidden. For each venue, we ignore levels below the top of the book, and we lump together all hidden liquidity above the best bid. Thus the state of the jth market center is described by four share quantities:

	Displayed	Hidden
Above best bid	0	r_j^+
At best bid	s_j	r_j

For example, Market Center B in Table 1 would be summarized as

0	$r_{\mathrm{B}}^+ = 700$
$s_{\mathrm{B}} = 200$	$r_{\mathrm{B}} = 100$

Our task is to estimate the unobservable quantities r_j and r_j^+. Specifically, we will produce a series of estimates $\hat{r}_{j,n}$ and $\hat{r}_{j,n}^+$, representing the hidden liquidity, at and above the best bid, respectively, on exchange j following the nth trade S_n observed anywhere in the market.

We do this using the framework outlined above.

1. First, let us ignore the quantity above the best bid. Suppose we see a print of S shares on market j, at the best bid price. We know that this must have happened because an incoming market sell order or marketable limit sell order crossed preexisting liquidity resting on that market. Using the construction above, we estimate the preexisting liquidity as
$$\hat{r}_j \;=\; \max\bigl\{\; S - s_j,\; 0 \;\bigr\}. \tag{1}$$
Furthermore, if a trade prints at the published bid, then we know there is no hidden liquidity above that level at that market center; we incorporate this below.

2. Now, suppose we see a print of S shares on market j, at a price above the best bid (but necessarily below the best published offer). We cannot be certain whether this was a new sell market order (or marketable limit) crossing a preexisting buy limit order, or the opposite. We allocate an estimate of the hidden liquidity proportional to the location of the trade price within the spread defined on that market center. Specifically, if the published bid and offer prices on market j are B_j and A_j, and the trade price is P, then we define
$$\mu \;=\; \frac{P - B_j}{A_j - B_j}$$
and we form the estimate of pre-trade hidden liquidity above the published bid as
$$\hat{r}_j^+ \;=\; \mu \max\bigl\{\; S - s_j,\; 0 \;\bigr\}. \tag{2}$$

As noted above, these estimates suffer from inevitable difficulties arising from the fact that we are trying to estimate something that is deliberately hidden. But we propose a very simple way to evolve these trades forward in time. We maintain an estimate of the current value of the hidden liquidity both at and above the best bid, at each exchange. Every time we observe a trade against hidden liquidity on that exchange, we bump this estimate

upwards, despite the fact that the trade itself will have consumed some of the liquidity. And every time we observe a trade *on any venue* that does not execute against hidden liquidity on exchange j, we reduce our estimate of the hidden liquidity on exchange j.

Specifically, let i count number of trades across all exchanges; this is equivalent to using "tick time" which is standard in high-frequency econometrics. We choose a decay factor ρ with $0 < \rho < 1$, by which estimated liquidity decays on each trade. Let $R_{i,j}$ be our estimate of hidden liquidity on exchange j immediately before trade i, and let $R^+_{i,j}$ be our corresponding estimate of hidden liquidity at all levels above the best bid.

- If trade i does not execute on exchange j, then

$$R_{i+1,j} \;=\; \rho\, R_{i,j}, \qquad R^+_{i+1,j} \;=\; \rho\, R^+_{i,j}.$$

- If trade i executes on exchange j at the best displayed bid, then

$$R_{i+1,j} \;=\; \rho\, R_{i,j} \;+\; \hat{r}_j, \qquad R^+_{i+1,j} \;=\; 0,$$

  where $\tilde{r}_j$ is the estimate of hidden liquidity from above (1).

- If trade i executes on exchange j above the best displayed bid, then

$$R_{i+1,j} \;=\; \rho\, R_{i,j}, \qquad R^+_{i+1,j} \;=\; \rho\, R^+_{i,j} \;+\; \hat{r}^+_j,$$

  where $\tilde{r}^+_j$ is the estimate of hidden liquidity from above (2).

Once we have this estimate, our routing decision is very simple. We send each block to the venue that has the largest total esimated liquidity: displayed quotes at the bid, plus our estimate of hidden liquidity at or above the bid.

References

R. De Winne and C. D'Hondt. Hide-and-seek in the market: Placing and detecting hidden orders. *Rev. Finance*, 11:663–692, 2007.

A. Esser and B. Mönch. The navigation of an iceberg: The optimal use of hidden orders. *Finance Research Letters*, 4:68–81, 2007.

J. Hasbrouck. *Empirical Market Microstructure: The Institutions, Economics, and Econometrics of Securities Trading*. Oxford University Press, 2007.

Á. Pardo and R. Pascual. On the hidden side of liquidity. Working paper, June 2006.

L. Tuttle. Hidden orders, trading costs, and information. Preprint, Nov. 2003.

streambase

About Robert Almgren

Robert Almgren has a long research and publication record in applied mathematics and finance; his papers on optimal execution of large transactions and on equity cost modeling have set the standards in the industry. Following a long academic career, he was a Managing Director and Head of Quantitative Strategies at a bulge-bracket bank. He is now pursuing independent consulting, as well as algorithmic trading and transaction cost modeling for non-equity markets.

About Bill Harts

Bill Harts is a management consultant known in the financial services industry as a pioneer of automated market making and algorithmic trading, as well as an authority on financial market structure andapplied technology. Most recently, Harts was Managing Director and Head of Equity Strategy for Banc of America Securities. He has also held executive positions including Executive Vice President of Corporate Strategy for The NASDAQ Stock Market, Inc., Managing Director of Strategic Business Development and head of Automated Market Making for the Global Equity Division of Salomon Smith Barney. Early in his career Harts was a consultant for Fischer Black's Quantitative Strategies Group at Goldman Sachs, one of Wall Street's first major algorithmic trading businesses.

About StreamBase

StreamBase Systems, Inc, a leader in high-performance Complex Event Processing (CEP), provides software for rapidly building systems that analyze and act on real-time streaming data for instantaneous decision-making. StreamBase's Event Processing Platform(tm) combines a rapid application development environment, an ultra low-latency high-throughput event server, and the broadest connectivity to real-time and historical data and leading EMS/ OMS software platforms. Six of the top ten Wall Street investment banks and three of the top five hedge funds use StreamBase to power mission-critical applications to increase revenue, lower costs, and reduce risk. It is also used by government agencies for highly specialized intelligence work. The company is headquartered in Lexington, Massachusetts with European offices in London. For more information, visit www.streambase.com.

Corporate Headquarters
181 Spring Street
Lexington, MA 02421
+1 (866) 787-6227

New York Office
845 Third Avenue, 6th Floor
New York, NY 10022
+1 (866) 787-6227

Virginia Office
11921 Freedom Drive, Suite 550
Reston, VA 20190
+1 (703) 608-6958

European Headquarters
34-36 High Holborn
London, WC1V 6AE
+44 (0) 20 7190 1713