Navigating dark and lit venues using algorithms and smart order routing

*David Morgan**

It is common to think of trading algorithms in terms of decision layers, but there is often a blurred line between these algorithmic layers and the smart order router (SOR). A new layer needs to be inserted to ensure the broker's SOR remains competitive (see Figure 1).

What, when, where, how?

In the not too distant past, the equity markets were relatively simple environments in which a human trader could know where an instrument was likely to trade best. The effects of regulation, fragmentation and new technology have made this effectively impossible for many stocks. There are too many markets, with prices changing too quickly for a human to keep up. A tool like the SOR brings back control, but for best effect needs to be combined with execution strategies and analytic indicators.

In order to delegate to a SOR successfully we need to know the distinctive functions of each layer: one to choose the strategy, one to manage trading against a benchmark and one to optimise the execution of individual slices.

At first sight some of the distinctions between these algorithmic layers and the SOR seem clear – the benchmark algorithms make the 'what?' and 'when?' decisions and the SOR makes the 'where?' decisions. In practice, however, this decision-making cannot avoid the question of 'how?' to trade.

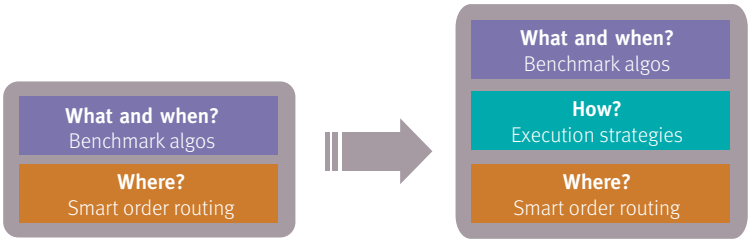
Algorithmic strategies

Common to the many types of algorithmic trading strategy is the same design objective – to achieve a superior performance when measured across a large number of orders. Algorithmic strategies exist in the world of probabilities.

Algorithmic trading is often built around a core set of 'benchmark strategies'. These

* **David Morgan,** solution architect, Fidessa

FIGURE 1: A NEW DECISION LAYER IN THE EXECUTION PROCESS



Source: Fidessa

algorithms aim to achieve or improve on a chosen benchmark, such as Arrival Price or VWAP, releasing slices to the market at intervals over the life of the order. The benchmark algorithm dictates what is to be traded – the size and the price limit of each slice – and also when it is to be released for trading.

Smart order routing

Smart order routing, on the other hand, is concerned with achieving best execution for each individual order slice, given the information available at that time. SOR strategies exist in the world of known facts.

Such strategies often have two phases: an aggressive first phase in which markets are swept to pick up all the liquidity that is already available; followed by a passive phase in which unfilled volume is posted to one or more venues. The sweep is driven by market data showing the displayed prices and sizes for the instrument across lit markets. This aggressive phase will often need to make multiple sweeps in order to exhaust all the available liquidity. Optionally the sweep may also ping dark markets, to look for any additional liquidity that is available inside the lit market spread.

The SOR dictates where the slice is to be traded, with the aim of trading as much of the slice as possible, at the best average price available, according to the published market data at that time.

Combining algorithms and SOR

When we combine a benchmark strategy and a SOR, the two together take us from the original order to a series of marketable slices over time and then to as many child orders as needed to

execute these slices. Together they answer the three questions of what, when and where.

With this model there is a clear separation between the role of the benchmark strategy, which creates slices using a statistical approach, and the role of the SOR that fans out these slices across venues using a deterministic approach. This separation of responsibilities occurs naturally when the two layers exist in different organisations; for example, where the buy-side runs the slicing algorithm but uses a broker to execute the slices.

But this simple separation leaves out an important part of the picture, which is the question of how the order is to be traded. This is the responsibility of an execution strategy.

Execution strategies

A SOR applies its rules to achieve a predictable outcome given the inputs available. Obviously the aim is to achieve the best result, but we need to define what 'best' means here. Achieving a best average price for the slice is fundamental, all things being equal, but there are other considerations.

While a SOR aims to optimise the execution of a slice, in general it is not possible to optimise everything; one has to optimise for specific targets, and accept that this will lead to compromises in other areas. The SOR must be told which set of compromises to make, as its default set may not be right for a particular case. It is likely that the right balance will differ depending on the size of the order, the market cap of the stock being traded and other order-specific context.

Execution strategies range from those that simply select a SOR strategy – for example, whether and when to use dark venues versus lit ones – to dynamic, adaptive functions that respond to market conditions. What is common to all is that the execution strategy uses the characteristics of the order to determine what is likely to be the best way to execute it, before handing over to a SOR to perform the actual execution. The execution strategy answers the question of how an order slice is to be traded. In other words, it selects the optimisation target.

A simple SOR dilemma

Consider a simple example. A large-cap European stock can be traded on its primary market, on several lit MTFs and also in several mid-point referencing dark MTF order books. Is it best to sweep the lit venues aggressively before the MTF dark order books, or vice versa?

The advantage of sweeping the lit venues first is immediacy. If a suitable price is displayed in the markets at the time the order slice is received by a SOR, then the SOR can and should take that price. That is the essence of best execution and the mechanism of public price determination. If the order is not completely filled then the SOR can subsequently sweep in the mid-point referencing dark books to see if any more volume is available.

This all sounds right and proper but there is a huge catch, which is that the execution on the lit markets may well affect the mid-price available in the dark markets. This obviously happens if all the volume displayed at the touch price on the reference market is taken, but can also happen even if only a portion is taken, or if only secondary lit markets are swept.

A better approach then seems to be to aggress the dark venues before the lit ones. Conduct a quick sweep around the mid-point books to pick up anything that is available, then execute in the lit markets. Assuming that the price-referencing venues are doing their mid-point calculation reliably, this should not result in a fill at a worse price than a lit market execution.

There is a catch to this too, though: the speculative sweep of dark venues delays the residual order volume getting to the lit markets. In the worst case nothing gets filled in the dark sweep and, by the time it has completed, a lit price originally in the market is no longer available either.

This is a simple example that creates a clear dilemma for the SOR. Which approach is best? The answer is neither one nor the other necessarily. It depends on what the trader is trying to achieve. Only when the all-important 'how?' question has been answered can the SOR do the right thing. If the aim is to optimise the price achieved, then sweep the dark venues first. If the aim is to optimise the chance of hitting currently displayed liquidity, then sweep the lit venues first. Both are right in different circumstances. When choosing SOR strategies, the question is not 'which is best?' but 'how is this order to be traded?'.

Determining 'how?'

The decision process in determining how to trade a particular market order may be simple or complex, according to need. Attributes such as the urgency and aggressiveness of the order, its sensitivity to market impact and the sensitivity to execution fees will all affect

the decision. The result of the decision process is a SOR strategy: a specific set of instructions for trading that order.

In the simplest cases the behaviour of the SOR is directed according to the origin and type of the market order. For example, if the creation of a slice has been triggered by a market event – by a sniper style of algorithm perhaps – then the job of the SOR is to hit all the relevant markets as quickly and as simultaneously as possible. In contrast, if the creation of a market slice has been triggered at a time interval, by a POV or VWAP algorithm, then the job of the SOR could be to dig out the best average price available, using a combination of dark venue sweeps together with lit sweeps at various price points.

At one step up in sophistication, the behaviour of the SOR can be directed in different ways at different times for the same parent order. For example, as it nears completion, with less time to catch up if liquidity is missed, a benchmark algorithm may be more aggressive and favour lit venues more.

Execution strategies come into their own when they add colour to the grey facts about what and when, by bringing in external market indicators. Many indicators are calculated in analytics systems that monitor current market data and match today's cumulative activity against historical trends and patterns, to determine measures such as volatility, liquidity, momentum and fair value. This is another dimension to optimising execution. An execution strategy may take direct input about the degree of aggression or stealth desired and weigh it together with facts about current market conditions and trend indications in order to choose the right SOR strategy to use to trade that slice at that time.

Beyond analytic indicators

As well as tailoring execution of the individual order slice and providing a link to analytics, an execution strategy has more flexibility than a SOR to implement specific approaches to market access. For example, an execution strategy for a passive slice might implement a minimum dwell time for some dark markets. For lit markets an execution strategy might dynamically peg a passive slice a few ticks outside the touch. Both of these depend on some analytic data – the optimum dwell time, or the likelihood of getting executed – but also add some specific new behaviours to the market interaction.

These are sometimes proposed as desirable features for a SOR, but that fails to understand the true function of smart routing, which is to produce a sensible result given the known facts at the time, without speculation about what might or might not be the case. It is perfectly valid behaviour for an execution strategy, though, as part of a quantitative approach to market access.

Is a SOR just an execution strategy?

It is tempting to think that with so much intelligence being built into the execution strategy, it may as well go the whole way and directly create the child orders for the various markets, dispensing with the SOR. This is a two-fold mistake.

First, it confuses the distinct functions of an execution strategy and a SOR strategy. As previously stated, the execution strategy is an algorithm that aims to produce a superior outcome on average, but it is dealing with inferences and probabilities rather than facts alone. The SOR, in contrast, is deterministic and predictable.

Secondly, a SOR needs a mass of detailed logic in order to operate, logic that is best kept hidden from view most of the time. Depending on the markets involved, it may have to deal with differences in order types, expiry types, dealing capacities, tick sizes, lot sizes, minimum display quantities, market phases, end-of-day behaviour, price thresholds, response semantics and more. As well as encompassing all this variation, the SOR exists in a fast-paced world of asynchronous events from markets with differing latencies, where keeping tight control of the child order state is paramount. These processes need to be fast and reliable and to work closely with the low-level market access components. The SOR does all this heavy lifting, allowing the execution algorithm layer to pursue its strategy unencumbered by the low-level technical details.

From this perspective, the job of the SOR is to manage such complexity and provide a 'virtual market' as a higher-level abstraction to the SOR user, whether that is an algorithm, a system or an individual trader.

Conclusions

The job of the smart order router is to take a marketable order at a given moment in time and fan it out across many markets in a predictable and optimal way. The job of trading algorithms is to trade many orders over an extended period of time to

produce a superior outcome. Several layers of algorithm may work together to choose and execute an algorithmic trading strategy. Sitting at the interface between the algorithmic trading stack and the virtual market created by the SOR is an execution strategy layer. This layer directs the SOR, allowing it to optimise the execution in the right way for each market order. It improves overall performance by defining not what, when or where to trade, but how to trade.

An implication of this is that if the broker is providing the SOR, the broker should also provide the execution strategies that drive the SOR engine. The execution strategy and SOR have to be closely integrated to provide the right sort of execution at low latency. The next layer up can assist by providing guidance to the execution layer about order urgency and sensitivity.

When the execution strategy is also informed by analytic indicators and the SOR has access to a full set of internal and external pools, both lit and dark, then truly smart execution can occur.