A Cinnober white paper on:

Algorithmic trading and its implications for marketplaces

Lars-Ivar Sellberg, 27 April 2010

Accommodating algorithmic trading

Algorithmic trading and the related term high frequency trading are generally defined as using computers to generate orders that are entered on marketplaces. Lately the term, rightly or wrongly, has been widely associated with its negative impact on marketplaces from a technical as well as a business and fairness perspective. Others argue that the positive effects of algorithmic trading – such as increased liquidity and the elimination of market inefficiencies – outweigh its potential negative effects.

Regardless of this, it is a fact that algorithmic trading has become increasingly popular in various forms on most of the major marketplaces around the world, demanding well thought-out strategies for how to accommodate this trend.

The purpose of this paper is to:

- Categorize different types of algorithmic trading
- Describe their implications for marketplaces
- Describe how to accommodate algorithmic trading in marketplaces while minimizing potential adverse effects
- Describe the use of special-purpose orders in the context of algorithmic trading

The author would like to thank Anders Mårtensson for invaluable support and advice during the writing of this paper.

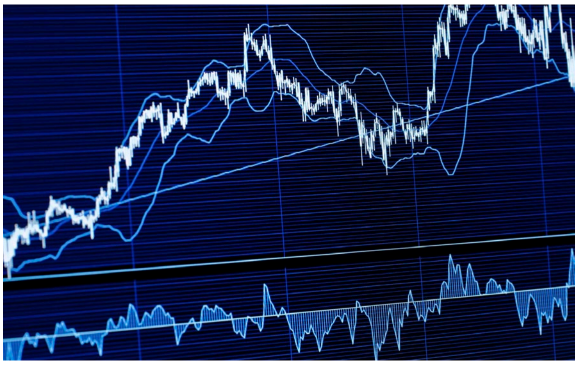
Categorization of algorithmic trading

The term algorithmic trading incorporates a wide range of trading strategies. Some are well known, for example arbitrage between index futures and the underlying instruments making up the index, while other are carefully guarded secrets at hedge funds.

This paper will not present an exhaustive list of trading strategies used in algorithmic trading. First of all, it is for practical purposes almost impossible to do so; secondly there is no need to do so from the perspective of analyzing the implications for the marketplace.

Instead two main categories of algorithmic trading are defined:

- Alpha-preserving
- Alpha-creating



Alpha-preserving algorithms are used when an investor has – through fundamental analysis or in some other manner – reached a trading decision and is about to execute it. Typically alpha-preserving algorithms deal with reducing market impact, thus minimizing slippage when executing the chosen trading strategy. A trivial example is breaking up a large order into smaller chunks executed over time.

Alpha-preserving algorithms are often perceived as being more benign than alpha-creating ones. For a marketplace that is out to maximize liquidity, they may nevertheless constitute a serious challenge.

Alpha-creating algorithms, on the other hand, include a diverse set of algorithms that by themselves try to create alpha. While alpha-creating algorithms do provide additional liquidity they may have a negative effect on a marketplace to varying degrees, both from a technical and business perspective. The degree of negative effect, if any, is dependent on the type of market model that is used by the marketplace, as well as the deployed technical solution.

Implications of alpha-preserving algorithms for marketplaces

Most alpha-preserving algorithms involve spreading out an execution of a large order both over time and execution venues. The goal is to minimize slippage when executing the order.

Slippage in itself occurs for two main reasons:

- Not enough resident liquidity at the marketplace. Note that there may be more latent liquidity outside of the order book, which might become available given enough time.
- Market impact. The market is shifting as a result of the new information created by the order being executed.

From a technical perspective, spreading out the execution of an order over time and over execution venues involves chopping it up in to small pieces which are then fed to the various execution venues gradually over a period of time.

The main negative effects of alpha-preserving algorithms can be summarized as follows:

1. Increased transaction flow as a result of chopping up orders into small pieces. This effect is typically negligible.
2. Decreased liquidity as a result of withholding orders instead of entering the orders directly into the market. The decrease in liquidity can potentially be quite significant.

Accommodating alpha-preserving algorithmic trading

Dealing with increased transaction flow

The increased order flow resulting from the chopping up of orders is usually not large enough to pose a significant performance problem for a modern marketplace system. The increased order flow will, however, reflect itself in an increased amount of trades. It is unlikely that these trades will affect performance, but they do have the effect of making traditional simplistic billings methods based on charging per trade somewhat questionable.

An important, and often neglected, aspect of a marketplace system is therefore to provide comprehensive information that is fine-grained enough to support a flexible billing model.

It is important that the organization clearing the trades for the marketplace can deal with the increased trade flow in an efficient manner, keeping the cost per trade down. One way of doing this is to perform trade-netting. This drastically reduces the number of trades that market participants need to deal with in their back-office applications.

Dealing with decreased liquidity

One of the major causes of slippage due to market impact is a non-optimized market transparency. In order to encourage order entry the following techniques can be used:

- Iceberg orders, a classical marketplace implementation of an alpha-preserving algorithm. Implementation is typically quite standardized although slightly more sophisticated variations exist, an example being where the quantity publically shown is randomized every time it is exhausted and refilled from the hidden quantity.
- Dark orders – orders that are active and match against the visible book but are not shown themselves. Note that a prerequisite in this area is that the dark orders are integrated with the visible book; a separation leads to an unnecessary loss of liquidity. It is also normally necessary to set some sort of threshold on the volume of the dark orders in order to avoid ending up with a totally non-transparent order book.
- Delayed publication of larger trades.

All the techniques above have in common that they give up some of the theoretical ideal of a transparent order book in return for increased resident liquidity.

From a marketplace system perspective it is important to provide enough functionality to facilitate such a tuning of market transparency.

Implications of alpha-creating algorithms for marketplaces

Although alpha-creating algorithms undeniably provide liquidity if looked upon in isolation, they may, as previously discussed, have a negative effect on the marketplace as a whole.

The main negative effects can be summarized as follows:

1. Increased transaction flow (by design or by mistake)

2. May discourage market makers

3. May discourage institutional investors

Some alpha-creating algorithms produce large transaction flows, often with a high order-to-trade ratio. This implies that the marketplace needs to scale up its system to be able to handle the increased transaction flow.

As a special case one also needs to consider the case of runaway algorithms, not as uncommon as one might think, that produce an irrational amount of transactions i.e. far beyond anything motivated by the underlying algorithm.

Market makers may be discouraged by alpha-creating algorithms that take advantage of weaknesses in the market maker applications. An example might be an algorithm that attempts to outwit a market maker application in a warrant market by simply being faster. By reacting faster to changes in underlying prices than the market maker application, the algorithm can trade at stale market maker prices, thus making a profit.

In other words, what is occurring is an “arms race” between passive alpha-creating algorithms (the market maker) and an active alpha-creating algorithm (the sniper).

Note that it is possible to reason in two ways about the above warrant market maker example. One is that the alpha-creating algorithm deployed by the sniper is removing inefficiencies in the market; the other is that it will force the market maker to widen his spread or indeed stop making a market, thus reducing the liquidity of the marketplace.

Theoretically the correct way of reasoning is probably that the sniper is helping to make sure that the market is as efficient as possible. In a “real” market the sniper might reduce liquidity by scaring away market makers.

Some institutional investors feel that some alpha-creating algorithms take unfair advantage of the information leakage inherent when executing large orders.

A marketplace needs to weigh the benefits of the extra liquidity created by algorithmic trading against the above-listed negative effects – and do it on a continuous basis.

A prerequisite for doing this is not only to understand which part of the total transaction flow originates from algorithmic trading but also its characteristics in terms of burstiness, order/trade sizes, average order life spans, etc. since the cost of handling a transaction flow is significantly affected by these types of characteristics. A marketplace system therefore needs to include the proper tools facilitating such an analysis.

Accommodating alpha-creating algorithmic trading

Dealing with increased transaction load

One direct way of controlling the transaction flow is to throttle the inbound transaction flow. A market participant is assigned a maximum transaction rate. If the rate is exceeded then transactions from that market participant are either delayed or rejected. A general recommendation is to leave the possibility to rapidly withdraw from the market un-throttled. Otherwise market participants might be inclined to add a risk premium on their orders.

An important choice when assigning throttling rates to market participants is if the goal is only to protect against runaway algorithms or also to protect against sustained usage.

If only protecting against runaway algorithms, then a certain amount of over-commitment can be acceptable. This means that the sum of the assigned throttling rates can exceed the total available marketplace system capacity, i.e. the marketplace operator effectively bets on all algorithms not going haywire at the same time. This might be perfectly acceptable if the effect of the increased transaction flow is limited to an increase in general latency experienced by all market participants.

It can be considered best practice to at least protect the marketplace system from runaway algorithms by using throttling.

Prevent market makers from being discouraged

The most drastic way to protect market makers from alpha-creating algorithms is simply to deny market participants who deploy these types of algorithms access to relevant market segments.

The obvious disadvantages of such a method are that it might lead to market maker complacency as well as a loss of the liquidity provided by the alpha-creating algorithms. Another disadvantage is the marketing problem of having a market segment labeled as "Only for uninformed retail order flow".

Another line of approach is to give incentives that encourage passive orders. That is, orders that reside in the order book as opposed to active orders that never reside in the order book but only match against the passive orders. This will not protect the market makers from competition but will make it more likely for the alpha-creating algorithms to contribute liquidity to the order book as opposed to just "sniping". In other words, hopefully converting them from liquidity takers to liquidity providers.

Incentive-creating techniques for passive orders include the following:

- Increase tick size. Increasing tick size makes it more expensive to buy priority, thus encouraging market participants to enter orders into the book. This is a classical way of increasing liquidity which has partly gone out of fashion with the increasing competition between venues to provide the narrowest spread however thin in volume. Thinner volumes even more enforce the usage of alpha-preserving algorithms.
- Use a pricing scheme where the passive party of a trade pays less or even gets paid for participating in a trade.

Yet another line of defense is disbanding continuous trading for one of the following alternatives:

Call auctions

Run call auctions with even intervals although the exact uncross times are randomized. This will effectively prevent any type of sniping. If the interval is small enough, potentially sub-second, many of the advantages of continuous trading can be maintained while still enjoying the sniping protection of the call auction.

The interesting problem is to determine the length of the call auction period. If it is too long then one loses the attractive property of the nearly instant execution that continuous trading provides. If it is too short then the sniping problem will start to become significant again.

Adaptive micro auctions

A more dynamic solution to this problem might be to run a call auction as soon as X ms have passed since the last order came in that would match another order. That would ensure that an owner of a resident order has time to absorb and react to the information provided by incoming orders, while still providing near immediate execution. An outer randomized boundary is needed in this case as well, in order to prevent auctions from running too long. Such a matching mechanism can be labeled as an "adaptive micro auction".

Indicative quotes

The market maker quotes on a continuous basis, but if a trade is to be executed a request must be sent to the market maker, who then confirms his quote and executes the trade. Although it might be considered a bit crude, it is used successfully in several marketplaces, among others the warrant markets in some regions.

Note that depending on the length of time the market maker has at his disposal when choosing if to execute the order or not, this method might imply an overprotection of the market maker, since the time the market maker has at his disposal in effect constitutes a free option.

Delayed executions against quotes

Instead of executing an order immediately against a market maker quote, the execution can be delayed for X ms. If the market maker has not changed his quote during this period, the execution proceeds; otherwise the order is cancelled and no trade is performed. Note that the market maker does not have any knowledge of the pending order during this period.

This procedure gives the market maker protection against sniping but removes the somewhat unfair advantage that arises from allowing the market maker to examine the incoming order prior to confirming his quote, as is the case with the indicative quotes model described above.

Prevent institutional investors from being discouraged

There are several ways to prevent information leakage when large orders originating from an institutional investor are executed. One of these is the use of dark pools.

Dark pools limit the transparency of large orders by not disseminating any pre-trade information, thereby lowering the market impact of these orders. It is also desirable to prevent alpha-creating algorithms from polling these dark books at too low a cost. That is, attempting to match at different prices using a small volume, thereby discovering the current prices in the dark pool. Such polling is made more expensive by making it possible to attach a minimum volume condition to orders in the dark pool – a minimum allowed execution size.

Dark pools have gained in popularity, but it should be noted that there is a trade-off between legitimate protection of institutional investors executing large orders and simply putting other investor categories at a disadvantage by reducing market transparency too much. Such a discussion is, however, outside the scope of this paper.

An important conclusion that can be drawn from this section is that a marketplace system needs to be flexible in order to provide the marketplace operator with a toolbox diverse and powerful enough to design a market model that is optimal for that particular market.

The use of "funnies" – special-purpose order types

Most marketplace systems include support for at least some complex special-purpose order types. These order types can be used to implement both alpha-creating as well alpha-preserving algorithms. From a marketplace operator perspective they are interesting for several reasons:

- Some algorithmic trading strategies can be implemented more efficiently from a marketplace system perspective using special-purpose order types. Thus by offering the use of special-purpose order types the total transaction flow is lowered, reducing the cost of handling it.
- Some algorithmic trading strategies can be implemented in a functionally superior way compared to implementing the same algorithms using standard order types, thereby generating additional liquidity. This is possible since special-purpose order types offer atomic complex operations that remove execution risk.

Some of the more common special-purpose order types are discussed below.

Pegged orders

A pegged order is an order whose price is linked to some sort of external reference. An example might be "I am buying at whatever the primary market is bidding up to a maximum price of X". Whenever the external reference price is changed the pegged order is automatically updated unless the cap price X has been reached.

Pegged orders can to a certain extent be used to mitigate some of the problems created by transaction flow generated by algorithmic trading. A common algorithm like "make a market at least as a good as on the primary market" is easily implemented with the use of pegged orders without the associated cost of processing inbound transactions that would have been the case if pegged orders were not available.

From a marketplace system perspective, however, it is important to realize that although the total system load is reduced, the pegged order feature might turn out so attractive by end users that its use rises dramatically, causing the resulting load to increase sharply, possibly beyond what would have been expected in the scenario with no peg orders available.

It is therefore important to be able to control the load resulting from pegged orders in the same manner as normal inbound transactions can be throttled. Typically this is done by scrubbing in-bound reference data, i.e. reducing the number of updates to an acceptable level, and limiting the number of active pegged orders.

Another aspect is that designers of algorithms tend to find very creative uses for pegged and other complex order types that sometimes are way outside what the marketplace system designer had expected and optimized for. It is therefore very important that the use of such features is tracked and that parameter changes and marketplace-specific optimizations are applied when needed.

Combinations (strategy orders)

A combination order makes it possible for an end user to enter a combined order like "Buy instrument A, Sell Instrument C at the net price of X".

From an algorithmic trading perspective such an order makes it easier to implement certain categories of algorithms. For example, many arbitrage strategies can be expressed directly using combination orders. This effectively removes any arbitrage possibilities; as previously discussed, this might be considered an advantage or a disadvantage depending on the viewpoint.

In order to provide a good implementation of combination orders the marketplace system must support the following combination order features:

- Implied orders (baits) i.e. that virtual orders based on the combination order are generated in the underlying order books. If not, the liquidity provided by the combination order is not visualized.
- Atomic execution of combination orders without any execution risk. The implied orders (baits) should also be generated atomically. That is, there should never be any risk of losing execution opportunities due to timing issues.

Linked orders (one-cancels-other)

A linked order is the conceptual opposite of a combination order. Each leg has an individual price and volume. But an execution in one of the legs automatically decreases the volume in the other legs proportionally.

Linked orders can be used to partially mitigate the problems previously discussed, where a market maker's alpha-creating algorithm is pitched against a sniper's.

Assume a marketplace where the market maker is obliged to provide liquidity in a large number of similar instruments. An example might be a fixed income instrument with approximately the same duration and credit risk.

If the market maker gets a trade in one of the instruments then he is likely to change his quote on the others. This means that the possibility of being hit multiple times in different instruments before having time to change his quote is something that will negatively influence the quality of his initial quote.

If the market maker on the other hand uses linked orders to quote, this risk is effectively removed. A trade in one of the instruments automatically reduces the size of the quote in the others. Thus the market maker is more likely to provide a better spread, given that one instrument at a time is considered.

A prerequisite for a marketplace system implementation of linked orders is that atomicity is guaranteed; otherwise none of the positive effects is likely to occur.

Conclusions

Algorithmic trading is steadily increasing in popularity and provides a significant part of marketplace liquidity today. There are two main objectives behind algorithmic trading: to preserve alpha and create alpha. Both of these have implications for the marketplace that need to be taken into consideration.

Algorithmic trading is a phenomenon that all marketplaces need to continuously monitor and analyze in order to be able to provide a well functioning market that attracts liquidity. This paper has outlined some of the implications of algorithmic trading for marketplaces and how its potentially adverse effects can be minimized.

Accommodating algorithmic trading in an effective way puts high demands on the marketplace trading system. It is necessary to provide the market operator with a high degree of flexibility and the means to design a market model that is optimal for that particular market.

Passion for change | Cinnober is the world's leading independent provider of innovative marketplace and clearing technology. Our solutions are tailored to handle high transaction volumes with assured functionality and low latency.

We are passionate about one thing: applying advanced financial technology to help trading and clearing venues seize new opportunities in times of change.

Among our customers are leading exchanges such as the Chicago Board Options Exchange, the London Metal Ex- change and NYSE Liffe. We also power new initiatives and alternative trading systems such as Alpha Trading Systems, Burgundy, Markit BOAT and Turquoise. Our clients rely on our platform-independent, Java-based technology for leveraging change quickly and cost-effectively.

Cinnober is headquartered in Stockholm and employs 170 people together having more than 1500 man years of experience from developing exchange and post-trade systems.

We are an independent technology provider and do not operate a market of our own, avoiding any conflicts of interest. We are not owned by — nor have any ownership interests in — any market operator.

Our track record says it all. We help our customers turn change into a competitive advantage.

Cinnober®

Cinnober Financial Technology AB
Kungsgatan 36
SE-111 35 Stockholm
Sweden
+46 8 503 047 00
cinnober.com