ITG INVESTMENT TECHNOLOGY GROUP

380 Madison Avenue, New York, NY 10017
Tel. 212.588.4000 www.itg.com

Maintaining Trade List Structure Across Dark Pools:
The Right Aggregating Algorithm Makes all the Difference

Hitesh Mittal
Managing Director, Head of Product Management, Algorithmic Trading
ITG Inc. in New York, NY
Hitesh.Mittal@itg.com

Ronald Taur
Vice President, Algorithmic Trading
ITG Inc. in New York, NY
Ronald.Taur@itg.com

Because they provide anonymity, dark pools are growing increasingly popular. Institutional investors and hedge funds can covertly move large blocks of stock without revealing themselves on the open market. According to recent NYSE figures, approximately 20% of all trades in New York Stock Exchange-listed stocks have gone through these dark pools, which is a notable increase from just 3% to 5% two years ago. And with the advent of dark pool aggregators which make choosing the "right" dark pools at the right time easier, that percentage is likely to increase even more.

It's important to note that not all dark pools are created equal. As such, they pose major challenges to traders faced with the all too common task of maintaining the structural integrity of a trade list. To get a large block done, traders need to simultaneously chase hidden liquidity across multiple pools. Dark pool aggregating algorithms can provide efficiencies, but due to the disparate characteristics of each pool, traders can find it extremely challenging to maintain the cash neutrality of a list executed across multiple venues. This article focuses on the need to maintain list structure and the particular challenges of doing so when accessing dark pools and then presents solutions to the problem.

THE NEED TO MAINTAIN CASH AND SECTOR NEUTRALITY

During large portfolio rebalances or transitions, it is often important for portfolio managers and traders to maintain the cash and/or sector neutrality of the list of orders generated. Essentially, cash neutrality means that the total executed sell and buy amounts are identical, or that the total sell-to-buy ratio is maintained, either for the entire list or at the sector level.

There are many reasons to maintain cash neutrality, including the needs to:

1. **Maintain cash flow** Traders want to preserve the total cash in hand, particularly if they are trading a list over multiple days, and maintaining cash neutrality ensures this objective is met. If, for example, there is a major announcement on an economic indicator after only a few hours of trading, the Portfolio Manager may want to halt all trading to revisit his positions. An imbalance in the sell-to-buy ratio could mean less cash is available to react to new opportunities. Furthermore, any imbalance at the end of the day can become costly in terms of settlement.
2. **Manage risk to avoid a variance in performance** An imbalance poses great market risk for the trader. By controlling the execution of buys vs. sells in a cash-neutral fashion, the trader is able to manage the market risk of his

unexecuted list. For example, if a trader submits a $100 million sell and $100 million buy list to a dark pool and $20 million sells get executed very quickly, the trader is left with a residual imbalance of $80 million buys. If the market quickly moves up, the trader could take a considerable hit as a net buyer. Even if a day of trading ends with the same positions as at the start, any change in market prices could create a costly implementation shortfall.

3. **Manage risk to reduce principal bid** There are instances where a trader may want to take a list to a broker for execution, but the combination of its liquidity and risk profile is too costly. As such, the trader may expose the list to a dark pool for a limited time with the hope of executing some of the more illiquid stocks. By doing so, the trader can improve the liquidity profile of the remaining list and reduce the principal bid. However, because the risk profile of the list is also a factor, it is important for that trader to carefully control the cash imbalance of the list while exposing it to the dark pools.

KEY CHALLENGES IN MAINTAINING CASH NEUTRALITY WHILE ACCESSING DARK POOLS

Maintaining cash neutrality in traditional market trading is relatively easy. Many brokers offer algorithms that can execute trade lists in a dollar-neutral fashion, while accessing *displayed* liquidity in the market. But how easy is it to maintain cash neutrality while accessing *hidden* liquidity in dark pools—particularly while using dark pool aggregation algorithms?

Until recently, it has proven difficult for a variety of reasons:

1. **Hidden pools do not guarantee execution.** Market orders routed to Electronic Communication Networks (ECNs) that are displaying a quote, or exchanges, are guaranteed at least partial executions. But, dark pools offer no such guarantee. When accessing hidden liquidity, traders have no idea if, when, and how much of an order will be filled. A key benefit of a dark pool is the ability to get a large block done without market impact. However, without a guarantee that a trader will get his block done, how can he ensure that he maintains cash neutrality? For example, it's possible to commit a large block of shares to a dark venue and only have a portion of sells completed, creating a substantial implementation shortfall. Ultimately, it is much easier to maintain cash neutrality when a trader knows exactly what will be filled.

2. **Most dark pools do not support cash constraints.** Although dark pools accept various order instructions and order types, such as limits, minimum execution size, limit relative to bid/ask, etc., most are based on single orders, as opposed to groups of orders. Currently, only ITG's POSIT® and POSIT Now℠ accept group-based order instructions, such as cash and sector constraints. With all other pools, the trader must manage the cash neutrality manually.

3. **No indication of symbol liquidity.** No dark pool, by definition, displays how much liquidity is available for buying and selling a stock. Some dark pools, such

as BLOCKalert[SM], Liquidnet and Pipeline provide indications for liquidity in a stock, although they typically do not indicate details such as side, size and price.

Unfortunately, indication of liquidity in a stock can make these pools susceptible to gaming. Therefore, these dark pool providers must typically establish extra policing to control the quality of the participants that have access to these pools. Because of these reasons, there are fewer of these pools available.

IS IT POSSIBLE TO MAINTAIN CASH NEUTRALITY WITH DARK POOLS?

There are clear challenges to maintaining cash neutrality while accessing dark pools. But is it possible? To find the answer, we interviewed many traders and discovered two prevalent ways they address the problem:

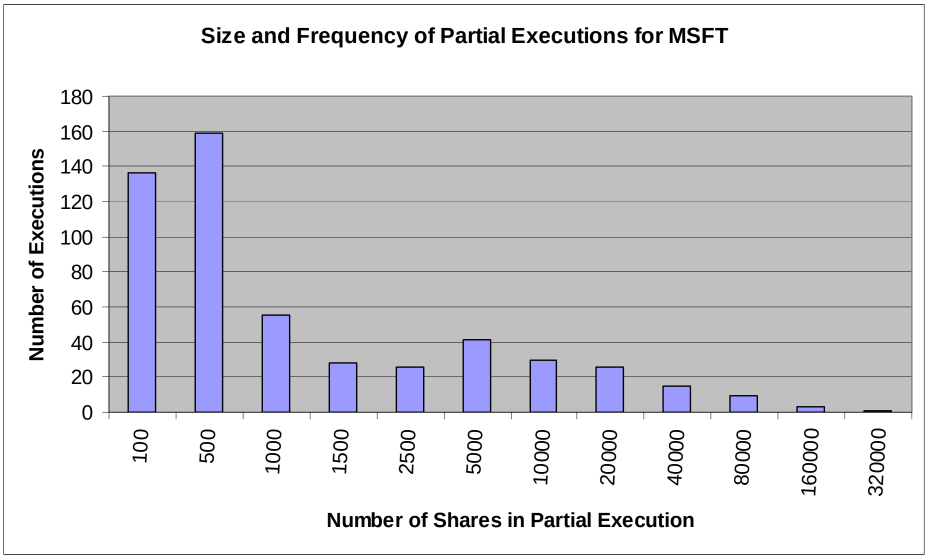
1. **Submit small slices of both buy and sell orders**

Some traders will send small slices on both the buy and sell side for every stock to avoid an imbalance on either side. As they get fills on one side, they adjust by sending orders in for the other side. So, for example, if a trader has a maximum tolerance of $1 million cash imbalance, he can send $1 million of each side of the list to the dark pools and adjust it on a real-time basis as the executions arrive from either side. The downside with this approach, other than the administrative implications, is that traders don't benefit much from the block liquidity that is potentially available.

2. **Submit larger slices to dark pools and send offsetting orders to ECNs/Exchanges**

In this approach, the trader sends larger amounts to dark pools and then sends imbalance-offsetting orders using aggressive order types to exchanges or ECNs. This approach works well when smaller discrepancies result. However, the trader is still exposing himself to considerable risk at short intervals. And if left with a large disparity, sending it to an exchange could cause the one thing he was trying to avoid in the first place—market impact.

The following chart shows a possible range of values for trading Microsoft stock in a dark venue. With such a large range, traders clearly take on a large amount of portfolio-level imbalance risk by not allocating correctly to even a single dark destination.



Source: ITG

Accessing single dark pools provide enough challenges to maintaining cash constraints, but many traders are now using aggregators to access multiple pools simultaneously. This automated approach pulls orders in and out of dark pools dynamically on the trader's behalf, without getting him or her over-bought or over-sold. But how does an aggregator fare when it comes to maintaining cash constraints across multiple pools?

While aggregation certainly improves execution outcomes, when it comes to maintaining cash neutrality, most aggregators can only take advantage of the features available in each distinct pool. If all pools do not accept cash constraints—which they do not—traders are left to their own devices, accessing pools one at a time to preserve the integrity of their lists.

DARK LIST: AUTOMATICALLY MAINTAINS CASH NEUTRALITY ACROSS MULTIPLE DARK POOLS

Investment Technology Group (ITG) recently delivered Dark List, a unique algorithm that not only provides dark pool aggregation but also is designed to automatically maintain cash neutrality across numerous disparate venues, regardless of the features available in each pool.

Dark List uses a unique approach to maximizing fills and maintaining cash neutrality, which includes:

1. **Maximizing liquidity opportunities** Many traders and portfolio managers like to maximize fills from dark pools so that they can minimize market impact and minimize opportunity cost. As such, aggregators are an efficient mechanism for accessing multiple pools. Dark List is a broker-neutral aggregator, seeking out liquidity across over ten venues, including POSIT, POSIT Now, and BLOCKalert, Level, BIDS, ISE and NASDAQ Cross. Dark List consistently allocates to all available venues (unless a trader chooses to turn one off) in an effort to maximize opportunities for execution.

2. **Utilizing the distinct features of each dark pool where possible** If a dark pool accepts cash constraint instructions or provides indications, it is much easier to tap liquidity while maintaining cash neutrality. Dark List utilizes these features

where available, currently only in POSIT, POSIT Now and BLOCKalert. It puts a large portion of all stocks in POSIT and POSIT Now, with cash constraints, and reacts to indications from BLOCKalert —only if that action does not violate stated cash constraints. As an example, let's say that a trader is executing a $200 million list and allowing $1 million maximum imbalance. The buy fills have been $20 million and sell fills have been $21 million. If Dark List receives a BLOCKalert indication for a stock that is a buy order, it will cancel orders for that stock (up to $ 1 million) in other dark pools and send it to BLOCKalert.

3. **Utilizing a statistical model that updates in real time to compute probability of fill and expected fill size in a given time horizon for each pool**

Dark List incorporates a proven statistical model that analyzes both historical and *real-time* trade data for each symbol in each dark pool. Using this model, the algorithm computes the optimal method of exposing each list to each pool while staying within the specified cash constraints. Without this model, traders trying to stay within cash tolerances *may* misallocate shares to one dark pool and miss out on liquidity in another.

Statistics of interest

In developing this model, we analyzed statistics of interest using ITG's database of dark pool executions from ITG Algorithms[SM], including:

- Average trade size in a time interval—or in other words, the average print executed during a particular time period, per symbol and per dark pool

* Expected hit rate in a time interval—or how often one can expect to get an order filled, per symbol and per time period
* Conditional hit rate (based on prior hits) in a time interval—or the subsequent hit rates for a particular symbol after an initial hit during a particular time period

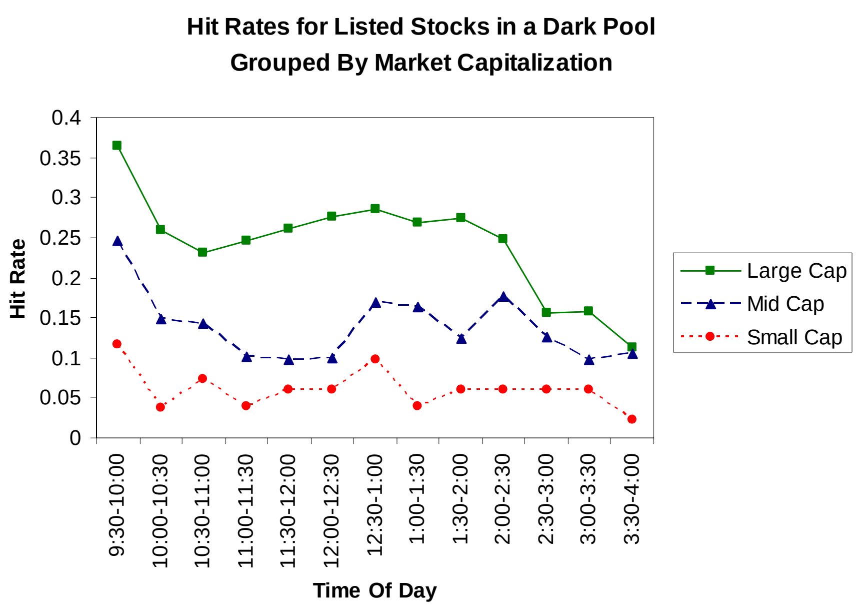
The historical data is certainly important to the model, but it is the conditional hit rate that is particularly interesting. Liquidity expectations drastically change in a dark pool as hits occur. It is natural that liquidity is sticky—whether new liquidity comes into play or is leftover from a previous order. After one hit occurs the expectation of another hit increases and activity ensues. Dark List is uniquely able to capitalize on this activity, changing its model dynamically based on actual real-time activity. For more details see Appendix 1.

Factors to determine these statistics

The following factors were useful in determining these statistics:

* Time of day
* Duration of order
* Prior hit rate and average trade size in the dark pool
* Market cap
* Listed vs. OTC
* Liquidity in the market

We were also able to model a stock-specific component, in addition to these factors. For stocks with insufficient execution data, we forecasted these statistics with peer group data. The following chart explains how some of these factors can drastically determine the liquidity available in each of these dark pools. This chart is for one of the dark pools used within ITG's Dark List algorithm:



Source: ITG

Model:

We allocated the order to different dark pools, based on the expected imbalance of all orders in each pool. Note that this allocation method is only for the dark pools that do not accept cash imbalances and do not provide indications of liquidity. We allocate in such a way that the expected imbalance in that time interval is zero.

Expected Imbalance for portfolio = $\sum_{i \in Sells} ExpectedFill_i - \sum_{j \in Buys} ExpectedFill_j$ * *Sell to Buy Ratio*

4. **Utilize real-time agents to control the allocation:**

   The variation of the expected fill in each of the dark pools is considerable. This can be easily seen with the following chart, which reflects the frequency of various ranges of trade sizes in one of these dark pools for MSFT, as an example. So the allocation model described earlier is combined with many real-time agents, including:

   1. Allocate some shares of each stock for each dark pool, regardless of the model results, so that it can help with the real-time discovery of new liquidity in that dark pool
   2. Update the model as we get hits within a dark pool and reallocate the shares within these dark pools based on the new model results
   3. Do not allocate more than a particular fraction of the total tolerance for the imbalance for any stock. For example, if the maximum tolerance for imbalance is $1 million, then do not allocate more than one third of that amount for any stock in a particular dark pool
   4. As the imbalance amount starts reaching tolerance levels, readjust the allocations to avoid hitting the maximum tolerance levels

Conclusion

Given the absence of quotes and guaranteed executions, the opaque nature of trading in dark pools requires that portfolio managers and traders minimize risk whenever possible. These risks are compounded when trying to also maintain the structural integrity of a list. One way to mitigate them is by utilizing a dark pool aggregator that automatically maintains cash constraints. With ITG's Dark List algorithm, traders can now use the combination of a statistical model and real-time agents to overcome the challenges in maintaining cash constraints across multiple dark pools.

With its unique and innovative approach, ITG's Dark List algorithm models expected fill size in dark pools based on historical liquidity, hit rates, average trade size and the liquidity on the same day. And, it uses probing techniques to discover liquidity in these dark pools and adjusts the allocations in real time based on the statistical model and newly found liquidity. Ultimately, ITG's Dark List algorithm helps minimize the risks associated with trading in dark pools by solving the problem of maintaining cash imbalances while maximizing the executions.

Appendix 1:

Average Trade Size (i,j,k,z) = Average Executed Value for Stock i on dark pool j during time of day k if the order is left for z minutes.

Average Trade Size gives us a gauge of the average—not the maximum—liquidity that we might expect in the next z minutes. We found that modeling the mean is far more stable than modeling the maximum amount of liquidity. Furthermore, we are able to estimate the probability that a fill, including partials, will occur when trading. This is known as the hit rate for a stock.

Hit Rate (i,j,k,z) = Probability of a fill for Stock i on dark pool j during time of day k if the order is left for z minutes.

Conditional Hit Rates:

In our data analysis, we determined that the hit rate increases for a stock, and all else is equal, if a complete fill in the same stock had just occurred.

Conditional hit rate (i,j,k,y,z) = (Hit Rate(i,j,k,z) | an execution in stock i in dark pool j just occurred in the past y minutes during time of day k)