

Algorithm Selection: A Quantitative Approach

JIAN YANG AND BRETT JIU

April 25, 2006

Abstract

The widespread use of algorithmic trading has led to the question of whether the most suitable algorithm is always being used. We propose a practical framework to help traders qualitatively characterize algorithms as well as quantitatively evaluate comparative performance among various algorithms. We demonstrate the applicability of the quantitative model using historical data from orders executed through ITG Algorithms.

BRETT JIU is a senior research analyst at ITG Solutions Network, Inc., 44 Farnsworth Street, Boston MA 02210, Tel: (617)-692-6741; E-mail: bjiu@itginc.com

JIAN YANG is a senior vice president at ITG Solutions Network, Inc., 44 Farnsworth Street, Boston MA 02210, Tel: (617)-692-6860; E-mail: jyang@itginc.com

The authors wish to thank Milan Borkovec, Gabe Butler, Vitaly Serbin, Xiangyang Wang, James Wong, Henry Yegerman and Ian Domowitz all of ITG Inc., as well as Yingchuan Wang for their support and comments. The information contained herein is for informational purposes only. Nothing herein is investment advice as defined by the Investment Advisers Act of 1940. ITG Inc. does not guarantee its accuracy or completeness and ITG Inc. does not make any warranties regarding results from usage. Any opinions expressed herein reflect the judgment of the authors at the time of publication and are subject to change without notice and may not reflect the opinion of ITG Inc. This communication is neither an offer to sell nor a solicitation of an offer to buy any security or financial instrument in any jurisdiction where such offer or solicitation would be illegal. All trademarks not owned by ITG are owned by their respective owners.

The relentless pursuit of lower transaction costs has led to increasing demand for sophisticated trading tools and algorithms, which in turn has led to an explosion in the number of algorithmic products offered in the marketplace today. Yang and Borkovec [2005] predict that this trend will continue as more investment management firms embrace best execution as a top priority.

Having more algorithms at their disposal offers traders both opportunities and challenges. On the up side, a trader now has the opportunity to pick the suitable algorithm that will most likely achieve the trading objective for each order. On the down side, the number of algorithm choices can be so large as to make it difficult to make a quick and correct choice.¹

Adding to the algorithm selection challenge is the fact that algorithms offered by sell-side vendors usually come in the form of a “black box,” with inner workings hidden to the end users. Because of this lack of transparency, users may find it difficult to clearly understand the performance characteristics of a particular algorithm, which, in turn, further complicates the algorithm selection decision.

Instead of looking inside an algorithm, we propose a systematic, quantitative approach to evaluating an algorithm’s historical performance by identifying the determining factors of relative performance across alternative algorithms, and we present a framework for algorithmic selection based on these underlying factors. Our methodology is easy to implement in practice and provides a quantitative framework for conducting performance attribution on algorithmic products. We will demonstrate how we perform empirical analysis on the algorithm performance and how we turn historical data-based model parameters into forward-looking algorithm selection criteria. Our proposed approach can also help investment managers and traders become more proactive in selecting algorithms that are of the highest value to them, and help to ensure the alignment of algorithmic trading with their investment objectives.

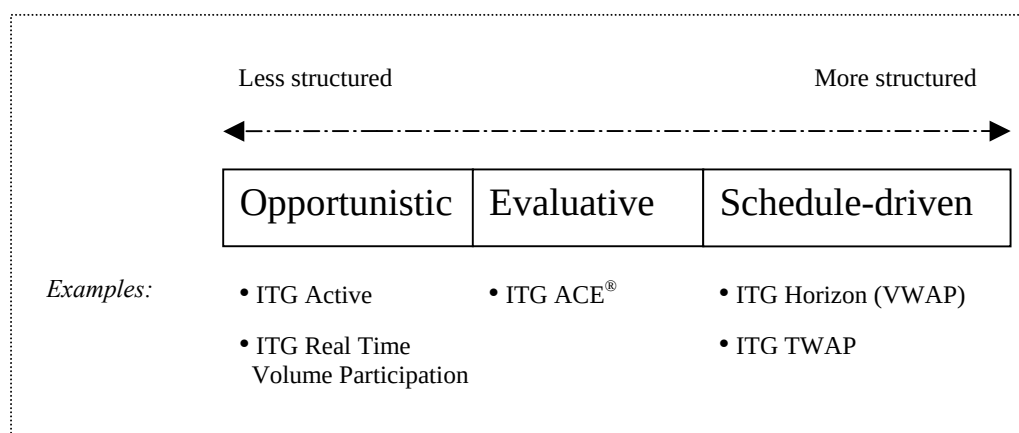
ALGORITHMIC STRATEGY SPECTRUM

The significance of conducting pre-trade “homework” on algorithms is well understood. The need to understand the nature of an algorithm starts at the point when an algorithm is offered by a third-party vendor. We begin our discussion of algorithm choice with a look at how algorithms can be categorized.

At its core, a trading algorithm takes an order, or trade list, and structures a sequence of trades that aim to achieve the objectives of the user, e.g., minimizing cost (vis-à-vis a specific benchmark), maximizing fill rate, or minimizing execution risk. Domowitz and Yegerman [2005a] suggest that, at the most abstract level, the different kinds of algorithms can be thought of as occupying a trade structure continuum, ranging from the less structured to the very structured. In Exhibit 1, we divide this range into three categories.

EXHIBIT 1

Spectrum of Algorithmic Strategies



On the less structured side, we find strategies that can be called *opportunistic*, in the sense that these strategies do not have pre-defined execution schedules; instead, they utilize real-time information to actively search for optimal times when trades can be executed. These strategies create execution schedules as they go along. At the beginning of an order, a trader does not know what the execution schedule will look like. An example is ITG Active (formerly known as ITG activePeg® to clients), an algorithm that employs sophisticated agent-like logic to continuously search for liquidity opportunities.

At the other extreme – on the more structured end – are algorithms that follow precisely-defined execution schedules; we call these algorithms *schedule-driven* strategies. The schedules are based on historical data, pre-programmed into the strategy’s logic and, save for small updates which incorporate real-time information, are followed precisely in optimizing trade entries. All VWAP- and TWAP-based strategies, for example, can be categorized this way. The realized

trade schedule will be similar to the pre-defined one, absent significant, unusual changes in liquidity over the order horizon.

Between these two ends is a category that we call *evaluative* strategies. Not surprisingly, these strategies combine approaches of both opportunistic and schedule-driven algorithms. At the macro level, these algorithms suggest how to optimally slice a large order in different time intervals, for example, half-hour bins. At the micro level, intelligent rules – often quantitative in nature – are employed to execute each part of the original order while balancing the tradeoff between cost and risk. Oftentimes these micro rules require the input of substantial real-time information, which makes them similar to opportunistic strategies. The trader will have a good idea of what the execution trajectory may look like, but the *ex post* trajectory may differ little or greatly from the *ex ante* prediction. An example is ITG ACE, a highly quantitative strategy that actively evaluates the potential price impact of each slice and continuously adjusts how and when each slice of the big order is executed in order to minimize the impact.

While our three-part categorization of algorithms is only a guide,² dividing algorithms into different categories is a necessary first step in deciphering the nature of the myriad strategies available. It is important to see beyond general descriptions and get a clear sense of what kind of strategy any given algorithm is at its core.

ALGORITHM SELECTION: A QUANTITATIVE FRAMEWORK

Given the availability of a basket of algorithmic strategies, attention is now turned to order-specific pre-trade analysis. Specifically, there are two questions concerning algorithm selection:

1. Is the order at hand suitable for algorithmic trading?
2. If so, which algorithm is the optimal one for trading this order?

It is well known that not all orders can be traded using an algorithmic approach. This is because, essentially, algorithms are pre-programmed logic run on computers. As such, algorithmic trading is not, and will never be, the magic bullet that solves all transaction cost-related problems. This is an important pre-trade analysis issue that is beyond the scope of this paper; instead, we focus on the optimal algorithm selection question.

We assume algorithmic suitability has been established and that the appropriate benchmark has been determined. The next step is deciding which algorithm, among the many available, should be used to trade a particular order. For example, even if VWAP is determined to be the best strategy, there may be a number of VWAP strategies from different vendors that can be used. One still needs to pick the best specific VWAP strategy.

We propose a simple quantitative framework that affords a quick but rigorous comparative analysis of algorithmic performance. The workflow of this pre-trade analysis consists of four steps:

1. Specify the model structure which links algorithm performance to a basket of factors that comes from order requirements, stock characteristics, or market conditions.
2. Derive the estimable form of the structural model and obtain performance attribution parameters using historical data. These parameters allow us to characterize how an algorithm's performance responds to changes in the factors.
3. Forecast the performance for each candidate algorithm using factor values known at order entry time.
4. Finally calculate a selection "score" for each algorithm. At this point, selecting the optimal algorithm is as simple as picking the one with the highest score.

We focus on the first step, model structuring, in this section.

While a trader's specific trading objective may vary, implementation shortfall has become a popular cost benchmark. First proposed by Perold [1988], implementation shortfall measures the price distance between the final, realized trade price and a pre-trade decision price. In practice this pre-trade decision price can be different to different people, e.g., the price at which a portfolio manager wishes to enter or exit a position, a previous day's closing price, today's open price, etc. For our purposes, we expand Perold's original definition to include limit orders and define implementation shortfall as *the difference between the share-weighted average execution price and the mid-quote at the point of first entry for market or discretionary orders, and the difference between the average execution price and the limit price of the order for limit orders*.³ The nature of the order – limit or non-limit – is taken from the very beginning; whether this nature changes during the course of trading the order is not considered.⁴ In addition, the im-

plementation shortfall is *sided* so its sign is consistent for both buy orders and sell orders. (A negative value signifies a price improvement.)

To account for the large variability in stock prices, it is the usual practice to express implementation shortfall in relative terms, i.e., the normalized difference between the share-weighted average execution price and the mid-quote immediately before the order started executing for market orders and the user-specified limit price for limit orders.

As a first cut, one may imagine using a simple “horse race” approach to selecting the best algorithm: whichever strategy that potentially achieves the lowest implementation shortfall cost wins. In this approach, it is only necessary to analyze the historical cost of each algorithm and apply the coefficients obtained from the analysis to forecasting its cost for the order at hand. This approach has the appeal of being very easy to implement, assuming historical trading data is readily available.

This simple approach suffers one significant drawback: it tells the trader nothing about how confident he or she will be in achieving the cost the model says an algorithm can achieve *on average*. A second-degree measure is needed to augment the selection process so the trader can compare algorithms along two equally important dimensions: cost advantage and confidence level. We propose a simple solution in the form of estimating the cost and its variance simultaneously. In other words, we will estimate both *how much* an algorithm’s implementation shortfall cost will be and *how closely* the realized cost will come to this *ex ante* estimate. The structural model specifies the conditional mean and variance of the cost as functions of relevant market- and stock-specific factors.

Our key innovation here is to propose that the cost and variance functions be jointly estimated, which provides a set of intimately linked performance attribution parameters for each algorithm. The analyst’s job, then, is to translate the structural functions into econometrically sound estimable specifications and employ the right econometric tools to carry out the estimation. For example, one can use time series, cross sections or panel data to do the estimation; all that is required is the appropriate econometric technique be used given the chosen specifications. The rest of this paper describes our empirical investigation, which can serve as an example to the reader who wishes for a “cookbook” guide to implementing quantitative algorithm selection.

Our proposed framework is completely broker-neutral and can be applied to *any* algorithm. It can be used to compare strategies across the algorithmic spectrum as discussed earlier in

the paper, or even within the same general type of algorithms (e.g., VWAP from vendor X vs. VWAP from vendor Y).

Next, we describe the dataset we work with before discussing our empirical implementation of the general framework.

DATA

Our aim is to obtain parameter estimates of the structural model from historical performance data. Our algorithm-related dataset contains over 100,000 single-name, completely filled client orders handled by the ITG SmartServer[®] suite of algorithmic strategies, also collectively known in the industry as ITG Algorithms. Here, a client *order* means an explicit instruction from a user to buy or sell a certain number of shares in a stock over a pre-specified period of time; it is the algorithm's job to break this order into individual trades or executions. The orders in the dataset cover U.S. stocks traded from February through June 2005. All the orders in our sample were either completed or canceled by close, so there are no multiple-day orders in our sample. We focus on three particular ITG algorithmic strategies: ITG Active, an opportunistic strategy (less structured); ITG ACE, an evaluative strategy (middle of the road); and ITG Horizon Smartserver[™], a VWAP strategy (more structured).

Each record in our dataset includes the following order-specific variables:

- Ticker
- Size (in number of shares)
- Side (buy or sell)
- Market or limit
- Limit price (if a limit order)
- Starting time and ending time for the entire order
- Share-weighted average fill price.

Exhibit 2 gives the sample statistics of the orders data after suitability filtering. The three algorithmic strategies vary significantly in average order size, whether measured as a percentage of MDV (median daily volume) or duration volume. ITG Active (the opportunistic strategy) tends to receive, on average, smaller-sized orders than ITG Horizon (the VWAP strategy), which

in turn gets smaller orders than ITG ACE (the evaluative strategy). This presents a potential sample selection bias problem across the algorithms, because it seems sensible that smaller orders tend to have lower IS cost. If the sample selection bias is indeed present, the implication is that when we estimate the model, ITG Active may get more favorable parameter estimates relative to the other two strategies not because it is inherently “better” but because it handles smaller, “easier” orders. In our case, this turns out not to be a serious problem because while size is a significant factor for the level of IS across orders under each algorithm, it does not correlate significantly with cost *across the strategies*. This is apparent when we examine the “Relative IS” column in Exhibit 2. Even though ITG Horizon handled larger orders than ITG Active, it had a lower average cost. Even though ITG ACE handled larger orders than ITG Horizon, it had a lower median cost. Furthermore, when we divided the range of relative order sizes into many brackets, we found that there were statistically sufficient sample points in each bracket for all three servers; meaning, our estimation results would not be significantly skewed by relative sample size in different order size baskets.

EXHIBIT 2

Sample Statistics

ITG Strategy	# orders	Order Size (shares)		Order Size (%MDV)		Order Size (%dur. vol.)	
		Average	Median	Average	Median	Average	Median
ITG Active	24,341	2,214	400	0.31	0.04	1.74	0.26
ITG ACE	4,635	8,305	2,300	2.04	0.56	4.99	2.08
ITG Horizon	28,926	5,765	1,500	0.93	0.24	1.97	0.60

ITG Strategy	Intraday Volatility (%)		Implementation Shortfall (cents)		Relative I.S. (bps)	
	Average	Median	Average	Median	Average	Median
ITG Active	0.023	0.016	-0.33	0	-2.60	0.00
ITG ACE	0.032	0.021	1.87	1	9.86	3.13
ITG Horizon	0.030	0.020	-0.68	1	-4.16	3.43

It should be emphasized that even when there is sample selection bias, it is an econometric estimation problem, not an inherent issue with our model. Econometric techniques exist to handle this bias. The key lesson here is, when performing quantitative analysis, one must take care not to fall victim to biased estimates due to the presence of sample selection bias or other data-related issues. An experienced econometrician can help solve this problem and provide statistically robust parameter estimates.

Another interesting issue with the data is the large difference between average relative implementation shortfall and the median value for all three strategies. In the cases of ITG Active and ITG Horizon – the two “extreme-end” strategies – the average relative IS is much lower than the median. For ITG ACE, the relationship is reversed. All these suggest the presence of significant skewness in the values. We’ll come back to this issue later when we discuss how we scale the IS value to obtain a distribution that is closer to normal.

In addition to the order-specific dataset, we also obtained stock-specific data that includes essential characteristics such as historical intraday volatility, historical volume profile, and historical ADV. We merge these stock-specific variables into the orders dataset by date and stock. The final dataset contains implementation shortfall for each order as well as various factor variables taken from both order and stock characteristics.

EMPIRICAL ESTIMATION

The structural model must be reduced to a form that can be statistically estimated. In deriving the corresponding reduced form equations, we make the explicit assumption that the relative implementation shortfall should be scaled by the stock’s intraday volatility. There are reasons for this transformation. First and foremost, a stock’s intraday volatility appears to be a significant driver for order cost.⁵ We can either include it as a regression factor (i.e., independent variable), or via a nonlinear specification. Earlier we mentioned that the implementation shortfall values in our sample exhibit significant skewness. But when we divide IS by the stock’s intraday volatility, we find the ratio having a statistical distribution similar to the normal distribution. The normality of the scaled implementation shortfall is important for us in constructing a single-number rating measure of algorithms in the next section. Scaling IS by volatility also reduces or eliminates heteroskedasticity in the model.

We make no *a priori* assumption regarding whether the relevant factors influencing the cost and its variation are the same or even overlap. It is possible that different factors, some common to both, influence the two functions. For example, time-of-day may be a significant factor for cost estimation, but may not have any effect on the variability of this cost estimate. The choice of factors is a question to be answered empirically.⁶

We also do not assume beforehand whether the cost and its variance functions are linear or nonlinear. In fact, as variances are often nonlinear in nature, imposing a linear functional form on cost variance would be too strict of a constraint and would likely produce inferior results.

We have run different sets of regressions using various factors to determine the final reduced (estimable) form of the conditional cost and variance functions. A few interesting results emerge from our analysis.

First, we found that nonlinear functions of order size relative to the predicted volume over the order time horizon, known as duration volume, to have worked best for estimating both cost and variance. In fact, this relative order size is the only factor that proves consistently significant in both equations. The use of duration volume contains an implicit time-of-day effect: the same order size and required duration have different impact depending on the time of entry, since duration volume will be different.

Second, there is a nonlinear, marginally decreasing effect of relative order size: the larger the order, the less increased marginal effect it has on transaction cost.

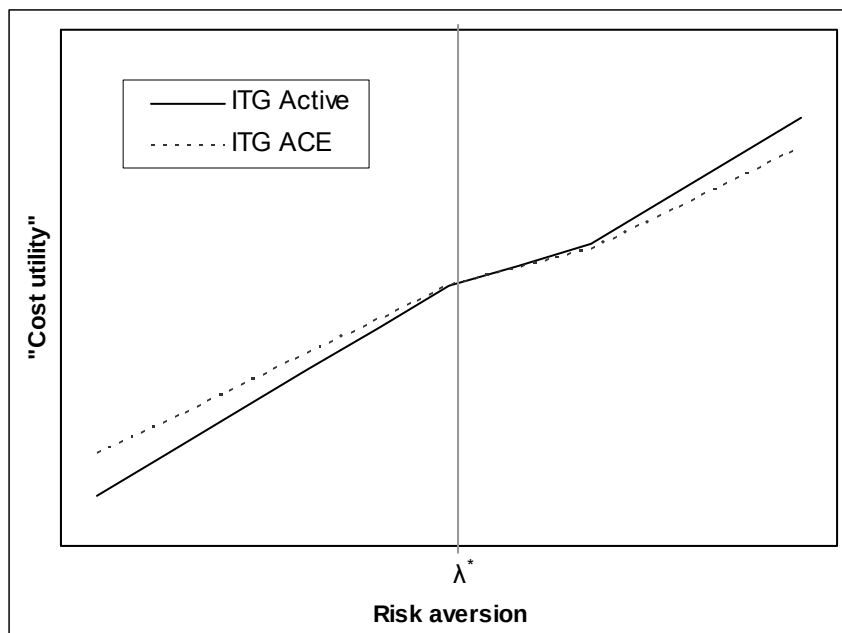
Third, even though market cap and intraday volatility do not enter our final cost equation, we do find that both market cap and intraday volatility have some effect on the magnitude of implementation shortfall. To control the impacts of market cap and intraday volatility, we double sort our samples along the two dimensions into four sub-groups: large-cap and small-cap, and within each market-cap group, high-volatility and low-volatility. The large-cap group roughly corresponds to the Russell 1000 stocks and the small-cap group is similar to the membership in Russell 2000 (plus some micro-caps). This division implicitly assumes the existence of a disjoint, “jump” effect market cap has on cost. The same logic applies to intraday volatility.⁷ One individual model is then estimated for each algorithmic trading strategy and, within each strategy, using each sub-division of the data. In all, we have three algorithmic strategies and four divisions under each strategy, giving us 12 cost-variance pairs of models to estimate.

One interesting pattern in our cost estimation results is that the impact of the relative order size factor on expected cost increases in magnitude as volatility drops from high to low. The reason for this trend has to do with how all of the algorithms presented here work their orders. All three use limit orders for executions to some extent; in fact, they employ an econometric model called the ITG Limit Order Model to forecast the probability of a limit order being hit at any given time. This probability is then used by the algorithms to determine how aggressive or

passive a limit order should be. When intraday volatility is high, the probability of a limit order being hit is high, therefore the individual limit-order trades are likely to be executed regardless of their sizes. In the aggregate, we observe a reduced effect of total order size on the total cost.

EXHIBIT 3

Algorithm Utility as a Function of Risk Aversion



Our estimation results also provide empirical support to our assertion that cost estimate alone does not determine the relative optimality of an algorithm and that cost variance *does* matter. To see this, we hypothesize a utility function that is analogous to the one employed in the mean-variance framework: it is defined as the sum of cost and risk aversion-adjusted cost variance. The optimization goal here is to minimize this utility function given the risk parameter by choosing over a set of available algorithmic strategies. Exhibit 3 plots the utility curves for ITG Active and ITG ACE over different values of risk aversion, assuming a fixed order size (set to 1% of duration volume), a low intraday stock volatility, and that the stock is a small cap name. Along the x-axis, risk aversion increases to the right; equivalently, risk tolerance decreases to the right. The two utility curves intersect at the point where risk aversion is equal to λ^* . To the left of λ^* , risk aversion is low (i.e., high risk tolerance), and ITG Active exhibits lower cost utility than ITG ACE and is therefore the better algorithm to follow. To the right of λ^* , risk aversion is high (low risk tolerance), and ITG ACE has the lower cost utility value and is therefore the winning

strategy in this example. Exhibit 3, in effect, demonstrates the interaction between cost and cost variance in determining which algorithmic strategy is optimal.

FROM ESTIMATION TO SELECTION

Now that we have constructed the models and obtained parameter estimates for each algorithm, we can perform comparative analysis and predict, given a specific order, which algorithm will be the best candidate for trading this order.

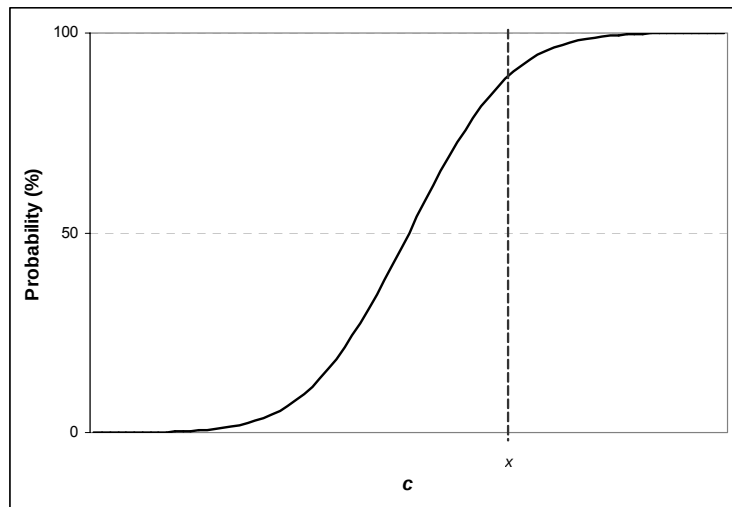
First, we forecast the cost and cost variance of each algorithm using the factor values known at time of order entry. To summarize, the factors that act as input to our estimated models are: (i) number of shares in the order; (ii) the stock's historical duration volume as calculated from (a) expected or required order start time and end time and (b) the stock's historical volume profile and ADV; and (iii) the stock's historical intraday volatility. We simply feed these numbers into the reduced-form models for each algorithm and obtain the predicted values of cost and cost variance for that algorithm.

Exactly how these forecast numbers can be used is up to the user. One obvious approach is using the cost utility function we mentioned in the previous section. The problem with this approach is the additional complexity introduced by the risk aversion parameter: how does one measure one's own risk aversion parameter and in what unit should this parameter be measured?

Instead, we propose another quantitative method which yields a concrete "score" for each algorithm: we can calculate a Value-at-Risk-like number that reflects the probability of keeping the implementation shortfall cost for an order under a certain threshold. Empirically our volatility-adjusted implementation shortfall cost exhibits a sample distribution close to a normal. From the estimates of the cost as well as its variance, the normal cumulative distribution curve can be used to compute the probability of achieving an implementation shortfall that is lower than the specified maximum threshold. This VaR-like value answers the question "What is the probability that implementation shortfall cost can be kept under a specific target?" Exhibit 4 illustrates this application, with c being the volatility-adjusted forecast of implementation shortfall, derived from the model's parameter estimates, and x the volatility-adjusted maximum target. (For example, we base our target on a maximum of 10 basis points in cost.) The intersection of the accumulative probability curve and the dashed line represents the probability of achieving a (volatility-scaled) cost less than or equal to x . When we calculate the VaR probabilities for all

ity-scaled) cost less than or equal to x . When we calculate the VaR probabilities for all the candidate algorithms, the strategy that achieves the highest probability is the optimal strategy for trading the order at hand: by using this strategy to trade the order at hand, we know it has the best chance of *not* incurring a cost higher than the maximum we are willing to tolerate.

EXHIBIT 4 Calculating Probability of Keeping IS Below x



Of course, this is only one example of an algorithm selection criterion, and this is not meant to be a complete guide for such analysis. Our framework is flexible enough to accommodate the construction of any statistic a practitioner wishes to derive from the estimated models. Our feeling is that the criteria should be easy to understand and also easy to implement.

CONCLUDING REMARKS

The rising popularity of algorithmic trading has led to the mushrooming of algorithm products in the marketplace today. A buy-side trader often has a large array of algorithmic choices available. Some of these algorithms may have come from in-house R&D, while others have been acquired from a third-party vendor and are likely to be of the “black box” type.

As we have amply demonstrated in this paper, using algorithms is not a simple task. The main advantages of using an algorithm, when used correctly, are two-fold: first, it gives the trader a systematic, disciplined way to trade an order that is consistent with the trading objective;

second, it generates an optimal trading trajectory that can maximize the chance of achieving the trading objective. To ensure algorithms are properly used, a trader must keep the following checklist of issues in mind when considering the use of algorithmic trading:

- *Nature of algorithmic strategy.* A thorough analysis should be done on the nature of each algorithm before the algorithm is ever used. At a minimum, a trader should cast the algorithm in the three-category paradigm we described. This paradigm helps the trader conceptualize the underpinnings of each strategy so he or she can later quickly call on the appropriate strategies for an order.
- *Suitability of algorithmic trading.* Some orders are less suitable for execution via an algorithm and may be better handled (and closely monitored) by humans. These are typically very large orders, orders for stocks with difficult liquidity conditions, or those with very specific requirements.
- *Fit between order and algorithms.* Even if an order is a “normal” one and can be algorithmically traded, the trader must determine which available algorithms are suitable for this particular order. Algorithms are not all the same. Some are better under certain circumstances while others prevail under other circumstances. When offered an algorithmic trading product, the trader must question the vendor regarding the “optimal” operating conditions of the product. For instance, what is the tradable order size range (in shares, in bps relative to daily volume, or in bps relative to duration volume)? Does the algorithm handle extraordinarily low or high volatilities? Is the algorithm time-of-day-dependent?
- *Dependence on benchmark.* Traders often have less flexibility in selecting the benchmarks as benchmarks are usually part of the desk’s trading policy, but it is still worth asking whether a given benchmark is the appropriate one under the circumstances. Additionally, it is important to have a good idea of how the benchmark is actually calculated inside the algorithm.

For the algorithm selection problem, we propose a quantitative approach that requires no knowledge of the internal mechanisms of the algorithm. Our approach focuses on performance attribution using historical data and provides parameters that help forecast the potential perform-

ance of the algorithms in the context of the specific order and the prevailing market circumstances.

Our proposed framework is general and is broker-neutral. We demonstrate, by example, how to turn the framework into a reduced form that can be estimated and how to use the estimation results in algorithm selection. The key takeaway is, it is not enough to just consider the comparative point estimates of the performance measure among the algorithm candidates. By considering the performance variance one can gain additional insight into how well each algorithm will likely perform given the order at hand. In our study, we estimate a simple single-factor model that is both intuitive and easy to implement; it also requires little computation time to generate useable *ex ante* selection scores for the algorithms.

Quantitative pre-trade analysis of algorithms is an essential part of algorithmic trading and should not be omitted from the trader's algorithmic toolkit. The extra time and effort needed to conduct the analysis will more than pay for itself, for each trade and in the long run, by helping to ensure that the best algorithm be used in achieving the trading objective.

END NOTES

¹In addition to the problem of choosing from a large number of algorithms, one must also consider whether the order at hand is suitable for algorithmic trading. We will not address this second issue in this paper. The interested reader can see, for example, Domowitz and Yegerman [2005a, 2005b].

²Yang and Borkovec [2005], in contrast, use a two-category approach and characterize evaluative algorithms as a special case of structured strategies.

³Some practitioners call the measure vis-à-vis mid-quote at entry (which is Perold's original definition) "realized market [or price] impact."

⁴For example, opportunistic and evaluative strategies may dynamically adjust the order type of each trade to liquidity conditions.

⁵This may simply be a feature specific to the strategies we study; it is possible that some algorithmic strategies in the marketplace can stay volatility-neutral.

⁶Any factor that is found to be significant empirically should also have sound economic justification behind it.

⁷In addition, this grouping approach can also be taken in regard to discrete factors such as exchange membership or industry sector classification.

REFERENCES

Domowitz, I., and H. Yegerman. "Measuring and Interpreting the Performance of Broker Algorithms." ITG Inc. research report, August 2005a.

Domowitz, I., and H. Yegerman. "The Cost of Algorithmic Trading: A First Look at Comparative Performance." In Brian Bruce, ed., *Algorithmic Trading: Precision, Control, Execution*, New York: Institutional Investor, 2005b.

Perold, A. "The Implementation Shortfall: Paper versus Reality." *The Journal of Portfolio Management*, vol. 14, no. 3 (Spring 1988), p. 49

Yang, J., and M. Borkovec. "Algorithmic Trading: Opportunities and Challenges." *Financial Engineering News*, No. 46 (November/December 2005), pp.14-15.