[redacted] Algo Due Diligence Report

Version 1.0

November 10, 2014

Prepared by [redacted]

[redacted].com

This document presents the findings of an independent assessment of five algorithmic trading strategies ("algos") coded and running under version 9.1 of the TradeStation (TS) platform.

We started with a set of optimized backtesting results for the period of January 2003 through current date. We did not have access to the source code, but did have to all variables that allow us to properly carry out the assessment. The assessment in good part focused on the optimization and backtesting methodologies used to arrive at the ECs that were presented to us.

There is no area in algo development of greater contention than that of backtesting and optimization. True, the process is fraught with "gotchas," but when we have an understanding of the nuances and subtleties - and how to put them in perspective - it can be an invaluable tool.

Optimization In-Sample Period

The presented optimized input variables were the results of the popular "Exhaustive" optimization technique where an EC is computed for each combination of the input variables over a user-defined range of values for each variable. The input values associated with the highest EC value are used as the "optimized" set of variables.

This iterative process is performed over a user-specified optimization period (interchangeably referred to in the literature as "in-sample period," "optimization window" or "training set"). In our case, that period is 2003-present. While it is generally accepted that "the more data, the better," we need to keep in mind that there are points of diminishing returns:

The optimization process looks at all data points in the optimization period and determines the best inputs parameter values that yield the largest ending value of the EC. But those values would not have been known to the algo realtime since they are a result of analysis of *all* data points over the period.

In other words, as the algo ran in the past, the data points used to derive the best parameter values were not yet available - they were in the future. This is known as a "*post*dictive" error, and is a subtle but consequential pitfall of backtesting.

What we were provided then is the set of values that yielded the best EC over the entire12-year test period. Three questions become apparent:

1 – If the optimized set of values were not available when the algos ran realtime, how would we have determined the optimized values to use back then?

2 – How would those values have performed, and what would the cumulative impact on the resulting EC?

3 – How effective is the optimization over the entire period in predicting the performance into the future?

The first two questions are addressed under the "Next-Steps" section below.

As to the third question:

The length of the in-sample data for the optimization is a key metric in the same way it is for the length (number of bars) in moving averages and linear regression: the longer the length, the more insensitive to more recent moves but the more representative of the major trend; conversely, the shorter, the more sensitive but the more likely to whipsaw.

To carry that paradigm to optimization, we can think of it as: the longer the in-sample window, the more likely the optimization is to incorporate specific market behavior in the past, and therefore the less likely to adapt to a changing market. (Indeed, "old habits die hard," not too different from human behavior in that regard.)

Below is the table "Algo Product Matrix.xls" provided by the algo authors summarizes the configurations we tested (Click to view).

<REMOVED TO PROTECT IP FOR PUBLIC DISPLAY>
<ALGOS EXAMINED: T1 Burst, S1 Breakdown, O1 Overnight Gap, B1 Breakout, P1 Push Pull>

Backtesting Limitations in TradeStation Platform

We next look at the assumptions made by the TS Strategy Engine (SE) when in backtesting mode and how they can lead to quite misleading results if we are not aware of their presence. While TS does not underscore what amount to pitfalls ("backtesting artifacts" in TS's parlance), those experienced are most cognizant.

Backtesting can be carried out in one of the following four combinations: IntraBar Order Generation (IBOG) ON or OFF, and Look-Inside-the-Bar (LIBB) resolution ON or OFF. Depending on the combinations, the results can be quite different.

Backtesting OHLC Default Chronology

Given bars of any time length (1, 10, 120 minutes etc.,) depending of the IBOG/LIBB settings, the SE evaluates the strategy so that decisions are made either a) once, at the end of the bar, or b) four times, at each of the Open, High, Low, Close values. The SE "guesses" which came first - the High or the Low - by their proximity to the Open: the closer one is assumed to occur first. (The Open and Close are known to be first and last.)

We know that assumption is not always true, and clearly, which comes first can make all the difference when the algo employs stop and limit orders on the same bar. Because of this and other related technical phenomena, it turns out the net effect of the assumptions very often yield overly optimistic results.

LIBB Chronology

In order to improve on the above issue, the concept of LIBB is utilized. When LIBB is selected, instead of only having access to the OHLC of the bar, we specify that the evaluations be performed on smaller bars within the larger bar. For example, a 120-minute bar can be broken into its comprising 120 1-minute bars, and then perform the strategy OHLC evaluations on each of those 120 bars.

Of course, higher resolution data is required, and backtesting time is greatly increased since we are computing 120 times as often but the payback is that the assumptions are now that much more accurate.

We have run backtests with all four combinations and evaluated results from the above perspective with knowledge of the exit order types utilized (stop/limit) in each of the algos and the code requirement that entries always be end-of-bar. ***We have concluded that running with IBOG off and LIBB on with 1-minute resolution yields a realistic setting where we can have confidence our results are to a great degree isolated from the pitfalls of the platform's backtesting mechanism.***

The attached table "InterSearch Performance vs. Backtesting Resolution.xls" summarizes the results of all four IBOG/LIBB combinations for all algos tested (Click to view).

<REMOVED TO PROTECT IP FOR PUBLIC DISPLAY>
<ALGOS EXAMINED: T1 Burst, S1 Breakdown, O1 Overnight Gap, B1 Breakout, P1 Push Pull>

Parameter Sensitivity

Another issue that needs to be kept in mind is the degree to which small parameter changes (perturbations) affect the resulting ECs. (In technical terms, the "flatness of the parameter space".) The basis here is:

a) If results are markedly different, "curve fitting" is implied and the algos' results are not likely to hold into the future since they are exceedingly tuned to the past, without allowance to changes going forward, or

b) There is in general a correlation between the resilience to changes in the underlying market going forward and the sensitivity to small parameter changes over the optimization period.

We performed this analysis by taking the parameter values optimized over the test period and then using them in an optimization with a parameter range 10% higher and 10% lower than the optimized values derived over the test period. The ECs resulting over all perturbations are seen to be largely insensitive across the +10%/-10% range.

Below is the "Stability Analysis - Input Perturbation Ranges.xls" table showing the foregoing (Click to view).

<REMOVED TO PROTECT IP FOR PUBLIC DISPLAY>
<ALGOS EXAMINED: T1 Burst, S1 Breakdown (EMD), O1 Overnight Gap, B1 Breakout, P1 Push Pull>

Slippage and Commissions

The tested algos were resistant to slippage and commissions over the entire test period.

Conclusions

Based on our analysis contemplating the above, we can arrive at the following conclusions, with the noted qualifications:

With a 75% confidence level

All five algos passed the above tests and we can conclude that they are tradable for the symbols evaluated, and should continue to perform in the *near-term* as implied by the backtesting using the optimized parameters as long as we keep optimizing periodically with an optimization window of the same length as has been used until now.

We note that we did not have available any substantiating documentation of actual live performance or realtime forward-testing results on a live account. We are assuming that live performance (over at least 30 trades for each algo) is in line with the backtesting metrics. This confirmation is a requisite and its absence would raise a red flag.
<COMMENT FROM AlgorithmicTrading.net: Statements are available should the evaluator wish to reconcile live trades with back-tested, we have 9+ Months of live returns on the NQ Burst, NQ Overnight Gap and NQ Breakout from 2 separate customer accounts>

Suggested Next-Steps

There are other steps we can take to add support to our findings so far, and hence raise our "confidence level" both for the backtesting and as we go forward:

- Refinement of the optimization process to make it more sensitive to recent market characteristics.
- Assess if it is possible to reduce the number of optimizable variables without giving up much in performance. This reduces the "degrees of freedom" which in general makes optimization more robust.

Walk-Forward Methodology

Walk-Forward techniques have become more popular recently in part because of the availability of faster multi-core processors make it feasible to undertake what is a sizeable computing task.

The concept behind walk-forward testing is to view the history available for backtesting in terms of a smaller "In-Sample" (IS) window. The window is "walked forward," and with each step forward, the window is optimized. As each step is optimized, a backtest using the resulting variables is performed on the data immediately following the IS, referred to as the "Out-of-Sample" (OOS) window, which is data still unseen by the algo at that point.

The OOS window is generally much shorter than the IS. As we walk forward, the IS window is stepped up by the width of the OOS window just tested so now the former OOS window becomes the tail-end of the new IS window. The process is then repeated until all data is exhausted.

Hence, our final EC is the "splicing" of the OOS ECs.

As an example, let's take our historical data from 2003 through 2014. Instead of doing one optimization over the entire 12 year period, we start with the first six months (Jan-Jun 2003) as our first IS window and use July 2003 as our first OOS set. Once we optimize and apply the found variables over the OOS, we step forward by one month so that our new IS is Feb-July 2003 and new the new OOS August 2003.

This methodology has immediate benefits that are major components in the development and testing of a robust trading algo:

- More data points over same data: We are now performing 138 six-month optimizations over the 12 years of data, instead of a single optimization over the same period. Although there is data overlap, the data is being analyzed under differing conditions which provides a higher statistical significance and better indication of realtime performance.

- Greater sensitivity to changes in the market: Because the IS windows are of smaller, optimizations are able to more quickly adapt to changing market environments; shorter optimizations are freer to adapt than longer optimizations that have to accommodate earlier environments (See comments under the first paragraph of "As to the third question").

The Work-Forward methodology finds immediate application for suggested next-steps as follows:

- 1 - Perform Walk Forward Analysis (WFA) to assess that, had the algo run live over the 12-year test period, the results would at least have been comparable to what we see when optimized over of the entire 12-year period. Theoretically, the performance derived from this analysis may better the results we have been presented.

- 2 - Document a process for optimization via TradeStation's Walk Forward Optimization (WFO) to be used to periodically re-optimize as we see new data in the future.

- 3 - Do a tick-by-tick replay of the algos over periods in the past using a technique that in essence generates higher resolution data so the results are very close to live execution.

- 4 - Compare the predictive power of the 12-year window, versus a shorter one with the use of WF cluster analysis.

To summarize, the use of WFA will allow us to:

- 1 - Determine actual past performance
- 2 – Periodically re-optimize going forward
- 3 – Provide substantiating documentation of performance for the purpose of sales and marketing

<COMMENT FROM AlgorithmicTrading.net: We agree that WFA could be done to further evaluate our algorithms. Certainly, for a new IP it is something that could become part of our design check-off criteria. However, at this time we are content with leaving our optimizations as-is given the success we've seen on the 3 core algorithms over the past year. While we agree that WFA can be a great tool for new IP developed and don't leave out the possibility of implementing such an optimization methodology, it is our current opinion that bearish periods such as 2008 should have equal weighting in the optimizations vs. a strategy that would weigh higher a rolling 6 month period>

Other Recommendations

- Request from algo author all realtime results available to support the claims of recent performance, ideally audited
  <COMMENT FROM AlgorithmicTrading.net: This data is available on our website, however would require substantial effort to reconcile each trade with the corresponding algo trade placed>

- Request from algo author an inspection of all available operational and coding documentation
  <COMMENT FROM AlgorithmicTrading.net: We will evaluate any additional requests for data carefully on a case-by-case basis. Our primary concern is the protection of the IP>

- Set up a trading account for the purposes of establishing auditable performance metrics. This is an account free of any transactions other than the trades provided by the algos. It need not be allotted more capital than what is required to trade 1 contract
  <COMMENT FROM AlgorithmicTrading.net: We currently have 9+ months of statements from accounts which have traded our algorithms live.>

- Place code and documentation in escrow and update it periodically
  <COMMENT FROM AlgorithmicTrading.net: We will evaluate any additional requests for data carefully on a case-by-case basis. Our primary concern is the protection of the IP>

About the Author, [redacted], [redacted].com

As a certified TradeStation specialist for almost 10 years, I have been engaged by many traders in search of robotic strategies. In the process, I have tested numerous trading ideas and concepts. The one thing that time and again has proven crucial is the ability to backtest accurately - yet there are so many ways to be misled by the process. While there is disagreement as to the efficacy of backtesting, I am a strong believer in the process and have found that those who differ in opinion is often because they are unaware of the subtleties, or because their environment is just plain difficult to backtest.

Earlier on I was an independent consultant in IT and internetworking. In the 90's I worked on the build-out of the Internet to Central and South America, working with communications carriers and ISPs to offer Internet access to the region over satellite and transoceanic optical fiber.

<u>Education</u>
MS in Electrical Engineering/Computer Science, Columbia University, New York

<u>Publications</u>
Published major articles in Technical Analysis of Stocks and Commodities
and the IEEE Transactions on Circuit Theory