

DEVELOPING TRADING STRATEGIES WITH GENETIC ALGORITHMS

Abstract

In this work we test the feasibility of using genetic algorithm (GA) to develop a trading strategy by combining a fixed subset of signals chained by logical operators. The work presented here is a Lean-based implementation of the technique described in the paper *High Frequency Foreign Exchange Trading Strategies Based on Genetic Algorithms*.

The project was implemented in the open source engine LEAN, using the genetic algorithm library [GeneticSharp](#) integrated with Lean by James Smith. Nine common technical indicators were selected with their default parameters.

After a running of 32 generations, the best out-of-sample trading strategy developed by the genetic algorithm showed a Sharpe Ratio of 2.28 in trading of EURUSD with 25 trades in the out-of-sample period of January – April 2017.

But more important that the results itself, are the layout of a framework flexible enough to test a wide range of strategies and the proof of concept of what is possible with two powerful open sources tools as Lean and GeneticSharp.

Introduction

Terminology

For the purpose of this work the following terms will have a specific meaning:

- **Technical indicator signal:** an event derived from the behavior of one or more technical indicator. In turn, the signal has a trading direction based in the expected behavior. E.g. the crossing of two moving averages is a signal. And, when the faster moving average crosses the slower moving average the signal direction is long because, following the technical indicator logic, this signal means the price will keep raising.
- **Trading rule:** a rule derived from a set of technical indicators signals and entails a trading action. The trading rule takes the form:
Indicator1 | LogicalOperator1 | Indicator2 | LogicalOperator2 | ... | IndicatorN => TradingAction.
Where:
 - *IndicatorX* is a statement which its true value depends on a standard technical indicator signal.
 - *LogicalOperatorX* is a logic operator (AND/OR) that join the indicators signal to generate a single statement, that in case of be true, fires the trading action. The trading rule can be entry to the market (only long) or closing an open position.
- **Trading Strategy:** a combination of two trading rules, one to entry to the market and the another to close an open position.

Technical Indicator Signals

For the purpose of this implementation, two generic technical indicator signals are considered:

- **Crossing moving averages:** this signal is generated when a faster moving average crosses the slower one from below (long signal) or from above (short signal).
- **Oscillator:** this signal is generated when an oscillator crosses its defined thresholds:

- o The long signal is generated when the oscillator crosses the lower (oversold) threshold from below.
- o The short signal is generated when the oscillator crosses the upper (overbought) threshold from above.

Trading Rule

For the sake of clarity, let's see an example of a Trading Rule composed of three Technical Indicator Signals defined as follows:

- **Technical Indicator Signal 1:** Crossing Moving Averages: **SMA(50) vs SMA(200)**. Direction: **Long**.
- **Logical Operator 1:** AND
- **Technical Indicator Signal 2:** Oscillator: **RSI(14)**. Direction: **Short**
- **Logical Operator 2:** OR
- **Technical Indicator Signal 3:** Oscillator: **Stochastic (14)**. Direction: **Long**

In this case, the Trading Rule is as follows:

IF SMA(50) crosses SMA(200) from below

AND RSI(14) crosses the upper threshold (80) from above

OR Stochastic(14) crosses the lower threshold (20) from below

THEN Trading Action.

The Trading Action can be taking a long position in the underlying FOREX pair or close an open position. A fully defined Trading Strategy need two trading rules, one for entry to the market and the other to close the open positions.

Hypothesis

The fundamental underlying assumption of using technical analysis is the past behavior of an asset price contains predictive information. And in turn, that information can be systematically exploited by rules from technical indicators. Based on that assumption, the hypothesis of this work is that a genetic algorithm can be used as meta-optimizer able to develop Trading Strategies as defined in this work.

Lean Implementation

The technical indicators used in the source paper are: MA, MACD, Slow Stochastic, RSI, CCI, Momentum Oscillator, Price Oscillator, Larry Williams, Bollinger Bands and OBV. However, the papers have no specific details about the indicator parameter nor the signals setups. For that reason, this implementation uses the standard indicator parameters and set ups.

The detailed implementation for each technical indicator signal can be found in the [SetUpIndicatorSignal](#) method. That method returns an instance of and [ITechnicalIndicatorSignal](#) interface, that instance can be a [CrossingMovingAverages](#) object or an [OscillatorSignal](#) object. Finally, in the [TradingRule](#) class the signals from the technical indicator and logical operators are evaluated [dynamically](#). Please note the simplicity of the [ITechnicalIndicatorSignal](#) interface, it only has one Boolean property (check if the instance is ready) and the *GetSignal* method that also returns a Boolean. This simplicity allows implement any kind of criteria with a trigger or flag, not only technical indicators.

The [QCAAlgorithm](#) is the implementation of a Trading Strategy, therefore, it requires a set of two trading rules composed of five Technical Indicator Signal. Each trading rule needs 14 parameters to be fully defined:

- 5 technical indicators
- 5 indicator signal direction
- 4 logic operators

And in turn, each Trading Strategy needs 28 parameters to be fully characterized.

In the source paper, the only pair considered is EURUSD with a resolution of 5 minutes. In this study trading only EURUSD was adopted, but with 1-minute resolution.

Genetic Algorithm

The individual in this genetic algorithm is the Trading Strategy, from the point of view of the GA, each individual is a realization of a chromosome with 28 genes. The population is initially 90 with a maximum of 150. The fitness used in the source paper is the Sterling ratio:

$$SterlingRatio = \frac{TotalReturns}{MaxDrawDown}$$

but to avoid negative fitness issues, the fitness used in this work is two to the power of the Sterling ratio.

$$fitness = SterlingRatio^2$$

The source paper in-sample period length is 20 days, and the out-of-sample is 15 days. For this implementation, the in-sample period was 6 months from July 1st 2016 to December 31st 2016 and four overlapping out-of-sample periods were considered, from one to four months starting the January 1st 2017.

Results

The training session was of 32 generations, 1873 backtests¹ ran in 26 hours non-stop with 16 instances in parallel. The training session was stopped after a nine generation fitness stagnation.

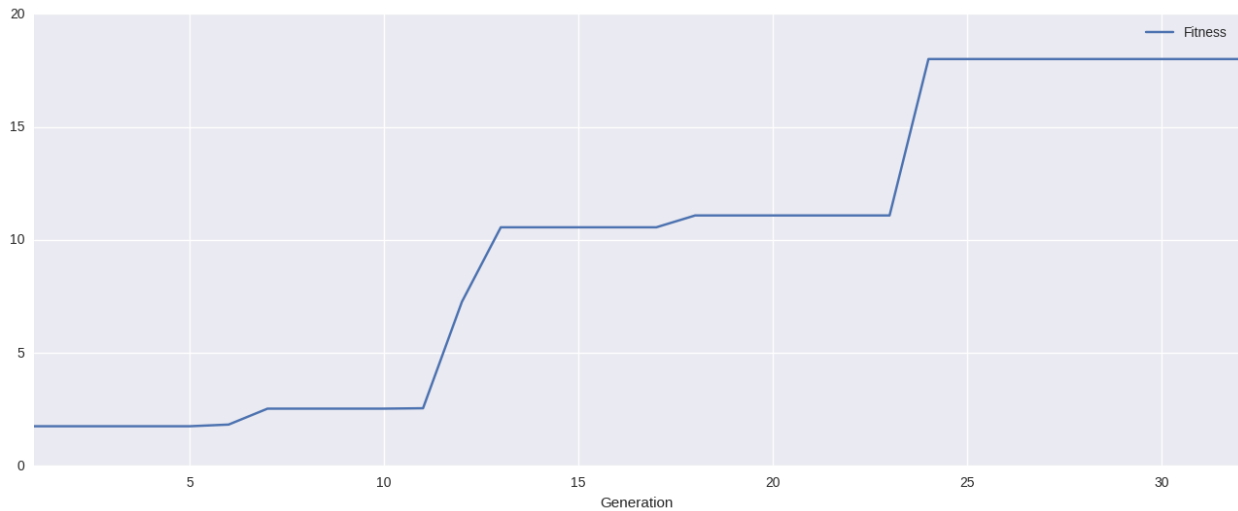
Attached is the backtest with performance of the best out-of-sample individual. You will note the backtest shared from the QuantConnect platform is different from the one in the GitHub repository. This is because the use of the [DynamicExpresso](#) library, which for security reasons is not whitelisted in the QuantConnect platform.

Fitness by Generations

The fitness shows a typical pattern in genetic algorithms, that is, many generations of fitness stagnation until an innovation reaches new highs.

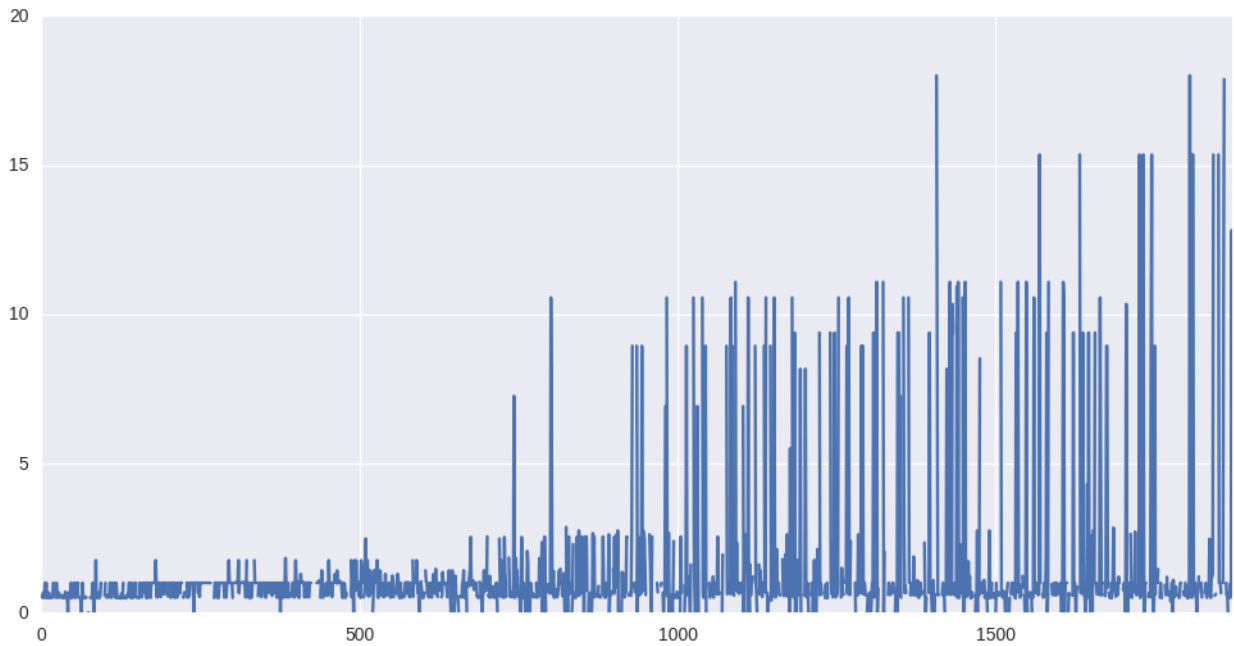
Figure 1 Fitness by Generation

¹ A compressed file with all the data analyzed in this section can be found [here](#).



Fitness by Individuals

Figure 2 Fitness by Individuals



The blanks in the plot corresponds to the 456 cases where the drawdown is equal to zero, thus the fitness cannot be estimated. From those cases, 249 corresponds to cases where the individual was unable to generate signals; the remaining 207 cases are consequence of some flaws in the experiment design and implementation. First, the algorithm used to run the evaluations had a very low market exposure, when an entry market order is sent, only the 10% of the total portfolio value is used. The reason for that decision was trying to keep the portfolio in positives values, even when many bad trades were made by the individuals (some individuals made more than 10000 trades!). The problem with the low exposure is that the drawdowns are very small. Second, the Drawdown statistic from the backtest results is parsed from a string in percentage format with two decimals; thus, the small drawdowns are rounded to zero. In other words, many individuals made trades but as the drawdown is rounded to zero, the GA ignore them in the Selection process. Some of those ignored individuals had good behavior in the in-sample period, the best one had an in-sample Sharpe Ratio of 1.9.

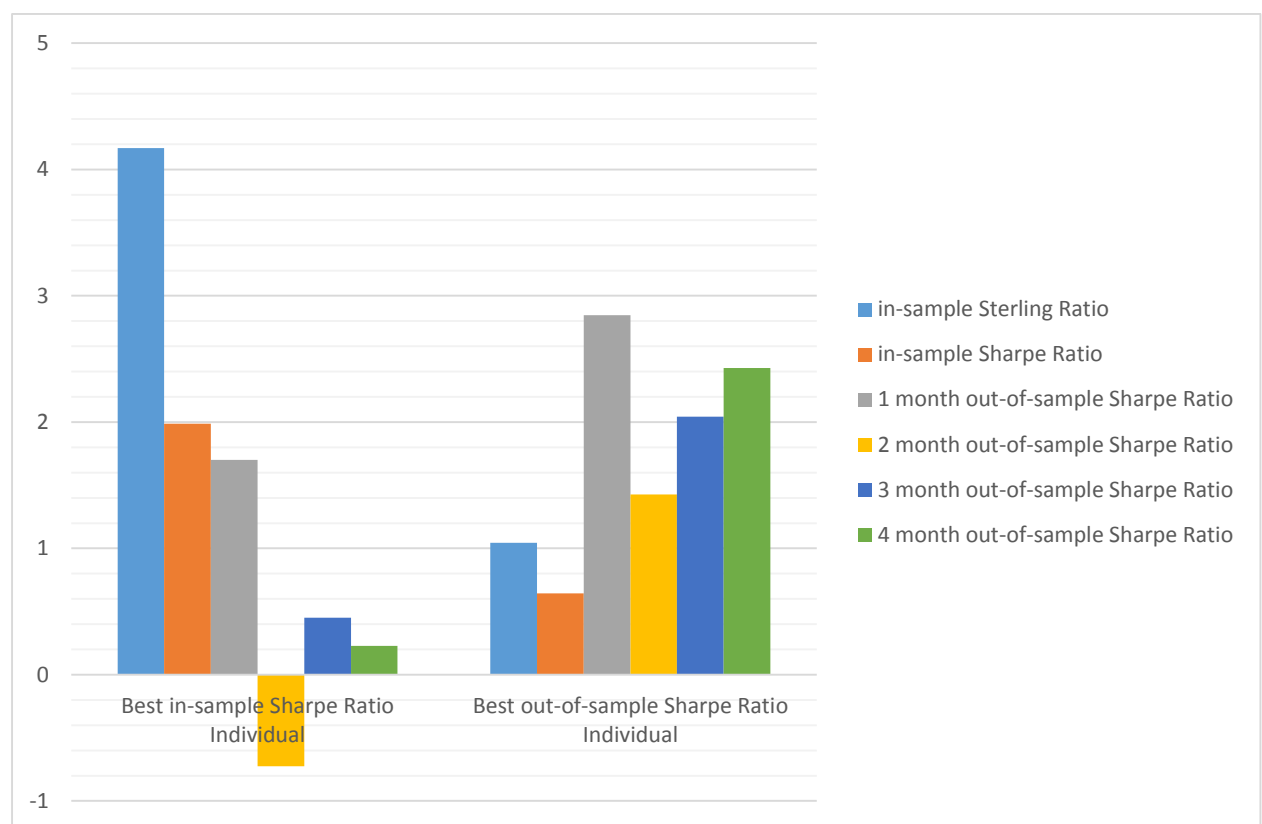
In-sample vs Out of Sample

Even when the statistic used as fitness is the Sterling Ratio, we will use the Sharpe Ratio as proxy for the out-of-sample performance. There are two main reasons, Sharpe Ratio is one of the most widely statistic used as performance. Second, this work is not just a machine learning experiment, its goal is to test if the GA can develop profitable strategies (as defined in this context).

From the 1873 individuals only 162 with Sterling ratio greater than one were considered to run the out-of-sample analysis. For each one of the 162 individuals four out-of-sample backtest were ran, all starting January 1st 2017, but with different length, from one to four months.

The Figure 3 compares the performance of the individual with the best in-sample fitness vs. the performance of the individual with the best out-of-sample Sharpe Ratio.

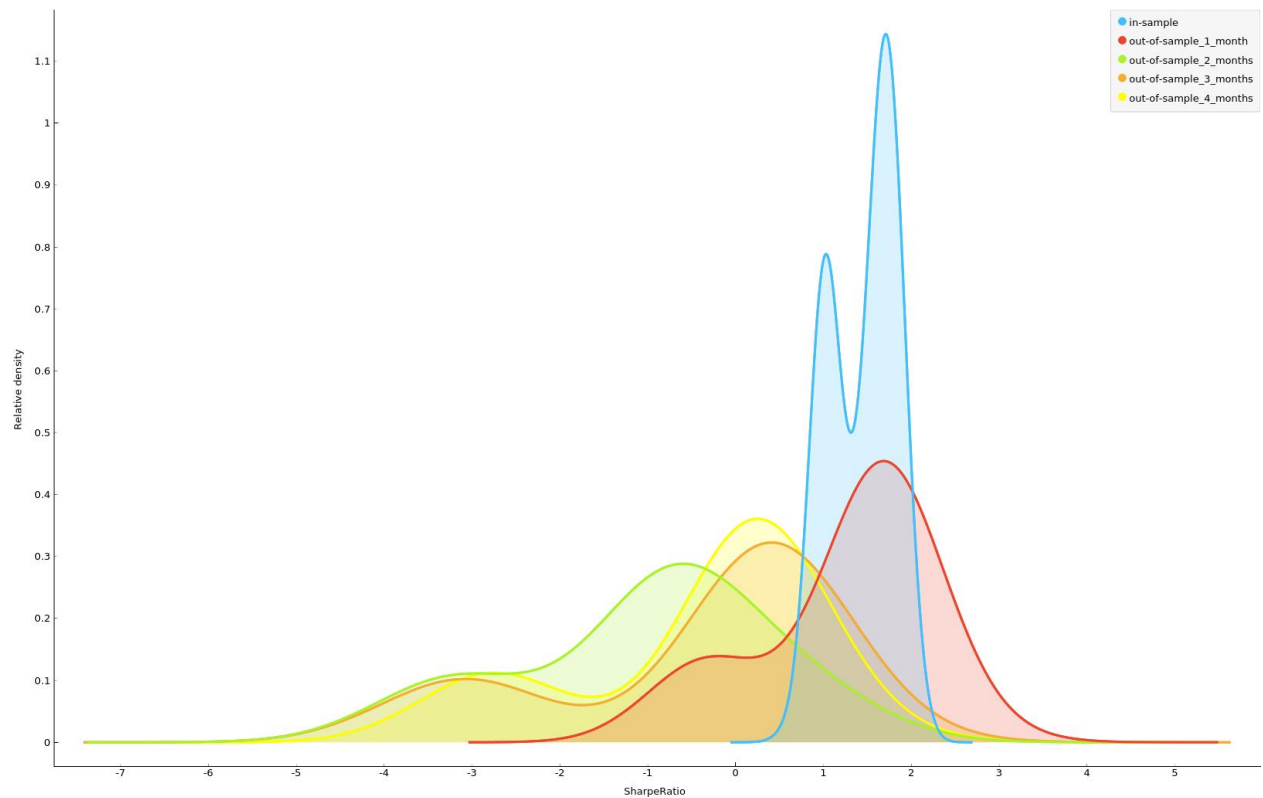
Figure 3 Out-of-sample Sharpe Ratio selected Individuals



The best in-sample performance reached a Sterling Ratio of 4.17 and a Sharpe ratio of 1.988 with 84 trades. The best out-of-sample Sharpe Ratio was 2.4 with 25 trades in the four months out-of-sample period. However, this individual had an in-sample Sharpe Ratio of 0.6 with 41 trades and an in-sample Sterling Ratio of 1.045. From this plot seems that there is not a strong correlation, between the in-sample and the out-of-sample performance.

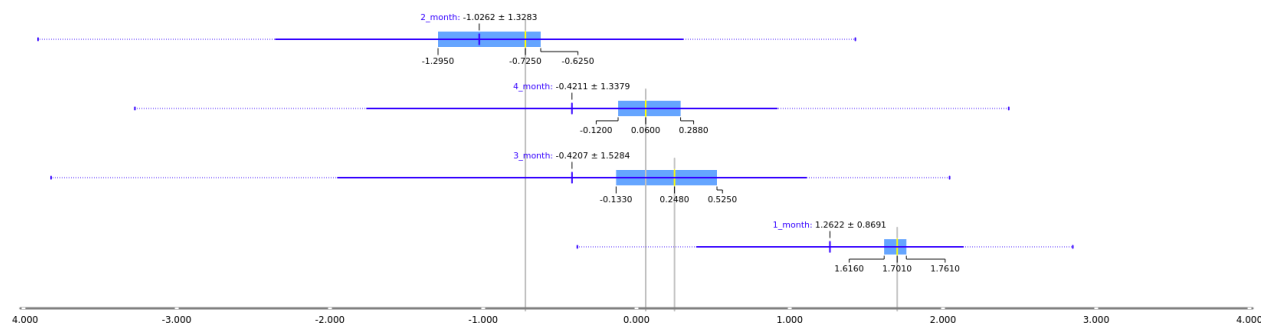
The Figure 4 shows the Sharpe Ratio densities of the in-sample and the four out-of-sample periods. As expected, the out-of-sample kernel densities have a lower kurtosis and are positively skewed. However, the loss in the out-of-sample performance is not progressive.

Figure 4 Sharpe Ratio in-sample vs. out-of-sample kernel density



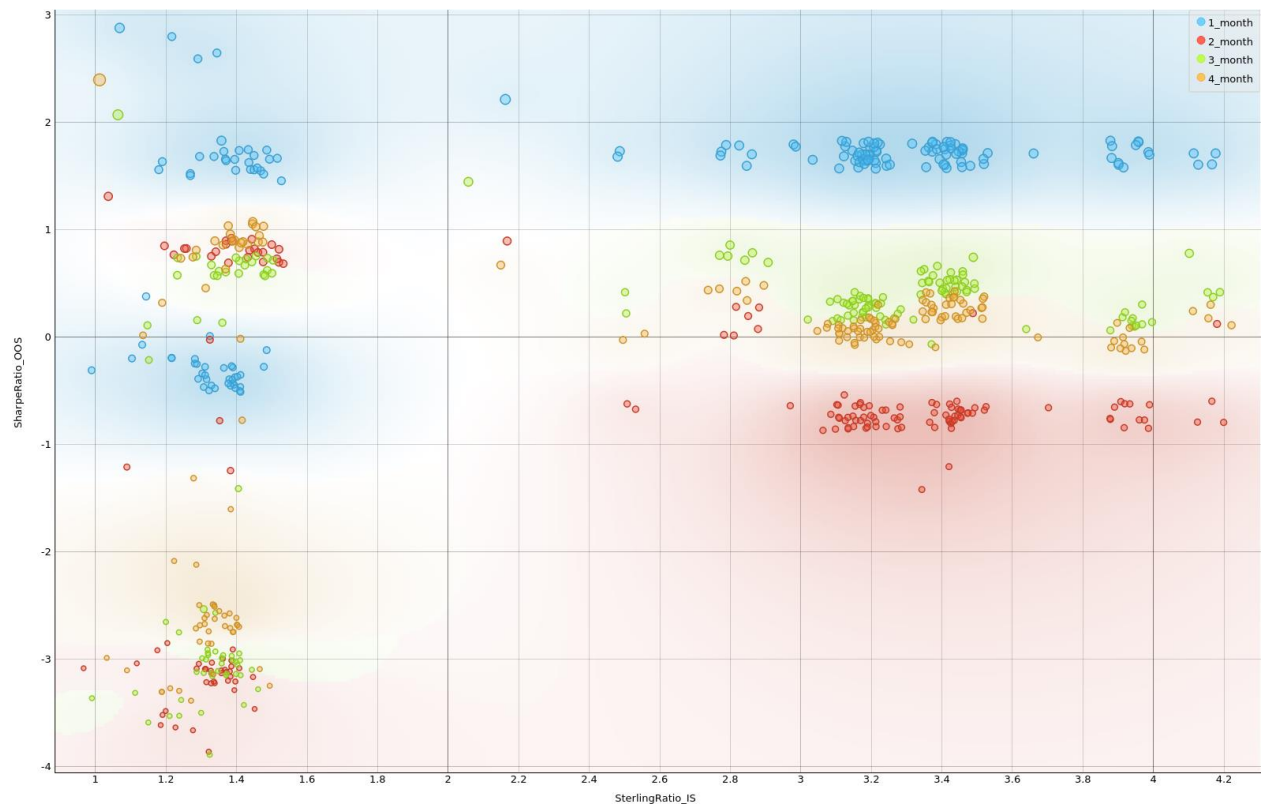
Complementing the previous plot, the Figure 5 shows the Sharpe Ratio by out-of-sample period length. Here is clear the second out-of-sample period (January – February 2017) have the worst performance and the following months the performance starts to improve slightly but fading out. This can mean some kind of short term edge for the strategies developed in this experiment.

Figure 5 Sharpe Ratio by out-of-sample Period length



Finally, Figure 6 shows the in-sample Sterling Ratio vs the out-of-sample Sharpe Ratio, the colors represents the out-of-sample length and the size the out-of-sample Sterling Ratio. Again, is clear the negative effect of February for all the strategies selected and the slight improvement in March and April.

Figure 6 Out-of-Sample Sharpe Ratio vs. in-sample Sterling Ratio



To test if the in-sample performance can explain the out-of-sample behavior, a series of regression were run (available [here](#)) with out-of-sample Sharpe Ratio as dependent variable and in-sample Sterling ratio as independent variable. Below, the results:

Out-of-sample Period	Coefficient	Significant at 95%	R ²
January 2017	0.5051	True	0.356
January - February 2017	0.4248	False	0.108
January – March 2017	0.9133	True	0.376
January – April 2017	0.6696	True	0.264
All periods	0.6282	True	0.174

The fact that the coefficients are positive and significant means in-sample behavior provides information about the out-of-sample performance. In other words, the GA can *be used as optimizer to develop trading strategies*, despite the very basic set of available tool it has (standards technical indicators).

Now, the poor correlation is expected, given the random nature of the genetic algorithms and the complexity of the experiment domain. GA are noisy by definition. On the other hand, the low correlation raises questions about the robustness of the Sterling Ration as fitness. To test the if the Sterling Ratio is a good measure of fitness, a meta-analysis should be done. Many training sessions must be run and check if the in-sample fitness vs the out-of-sample performance shows the same behavior consistently.

Conclusion

The statistical analysis shows the feasibility of using genetic algorithm (GA) to develop a trading strategy by combining a fixed subset of signals chained by logical operators, despite the set of very

basic (dull) signals used. This could potentially be improved further by using a set of weak but more profitable signals.

Although the Sterling Ratio shows some power as fitness, a meta-analysis should be done in order to address its robustness. However, using a more complex/comprehensive fitness formula is another possible improvement.

The experiment has some flaws in design and implementation. One of these shortcomings, detailed in the results section, is the issue of low market exposure plus parsing the backtest statistics form fixed two decimal strings. This issue affected the Selection process itself, potentially hurting the GA outcomes. In addition, just one training session was considered, to fully test the genetic algorithm power, multiple training session should be running.

Finally, the Lean-centric framework has two very strong advantages:

- The training evaluation can be as complex as needed (including ask-bid spread, fees, commissions, slippage model, risk management, etc.) to enhances the training by exposing the individuals to realistic environments.
- The QCAAlgorithm used by the genetic algorithm to evaluate the individuals can be used to trade in live paper mode and even in real trade. Therefore, a profitable set up developed by the genetic algorithm can be tested in real time or put to trade immediately.