Generating Directional Change Based Trading Strategies with Genetic Programming

Jeremie Gypteau, Fernando E. B. Otero, and Michael Kampouridis

School of Computing
University of Kent, Canterbury, UK
{jg431,F.E.B.Otero,M.Kampouridis}@kent.ac.uk

Abstract. The majority of forecasting tools use a physical time scale for studying price fluctuations of financial markets, making the flow of physical time discontinuous. Therefore, using a physical time scale may expose companies to risks, due to ignorance of some significant activities. In this paper, an alternative and novel approach is explored to capture important activities in the market. The main idea is to use an intrinsic time scale based on Directional Changes. Combined with Genetic Programming, the proposed approach aims to find an optimal trading strategy to forecast the future price moves of a financial market. In order to evaluate its efficiency and robustness as forecasting tool, a series of experiments was performed, where we were able to obtain valuable information about the forecasting performance. The results from the experiments indicate that this new framework is able to generate new and profitable trading strategies.

Keywords: directional changes, financial forecasting, trading, genetic programming

1 Introduction

The global financial system, recently rocked by the crisis, is open 24 hours a day, 7 days a week and can be defined as a complex network of interacting agents (e.g., corporations, retail traders). With an average daily turnover of 3-4 trillion USD [1] and with price changes nearly every second, its activity varies at different times of a day and reacts on the announcement of political or economic news. As a consequence, financial time series are unevenly spaced and make the flow of physical time discontinuous [2]. The majority of traditional methods to observe price fluctuations in financial time series are based on physical time changes and, nowadays, it has become very difficult and challenging to observe price movements through this scale. For example, what researchers and practitioners tend to do is to use snapshots of the market, taken at fixed intervals. For instance, they first decide how often to sample the data, and then they take snapshots at the chosen frequency. Therefore, these snapshots create an interval-based summary, e.g. daily closing prices. However, this lacks realism. In everyday life, we record history by identifying key events.

To model the discontinuous fluctuations in term of volatility, new time scales were introduced in [3], called intrinsic time. Intrinsic time is an alternative approach that replaces the notion of physical time scale and looks beyond the constraints of the physical time within financial data and constitutes an event-driven approach. In the same paper, the concept of directional changes was introduced. A directional change event is characterized by a fixed threshold of different sizes and time in price time-series, eliminating any irrelevant details of price evolution. In this way, the price fluctuations are described by the frequency of directional change event over a sampling period, which provides an alternative measure of the risk.

While directional changes have already been used for summarising data in an event-based manner, they have not been used for forecasting purposes. Thus, in this paper our aim is to demonstrate that the new paradigm of DC leads to effective and profitable trading strategies, which can potentially outperform the traditional physical-time strategies that are currently widely used. In order to do this, we use a genetic programming algorithm to automatically generate trading strategies that make use of DC thresholds—strategies are created by combining the output of multiple thresholds in a single expression. The GP-generated trading strategies are compared against the use of fixed DC threshold to evaluate whether GP solutions explore the advantage of using multiple thresholds simultaneously or not.

The rest of this paper is organised as follows: Section 2 presents related work in the field of financial forecasting, and Section 3 gives an overview of the concept of directional changes. Section 4 then discusses how we used the genetic programming algorithm to generate trading strategies. Section 5 presents our experiments, and Section 6 concludes the paper and presents future work.

2 Related Work

There are numerous works that attempt to create trading strategies to be used in financial markets. Some recent examples include [4–7]. In [4], the authors studied the evolution of trading strategies for a hypothetical trader who chooses portfolios from foreign exchange technical rules that are derived from daily data. On the other hand, [5] presented evidence of time-of-day effects in foreign exchange returns, by using hourly data for the EUR/USD currency. Moreover, [6] used Genetic Network Programming to derive trading rules from the Tokyo stock market, by using daily closing prices. As we can observe, in all of the above works time has been taken in fixed intervals (daily, hourly). The same happens with other works in the literature. In fact, to the best of our knowledge, there are no forecasting/trading works that use non-fixed intervals. Lastly, [7] proposed the use of the so-called zig zag technical analysis indicator in two scenarios to predict trend reversals: using a fuzzy logic model and a neural network model. The zig zag indicator shares similarities with the concept of directional changes (i.e., both use a threshold to filter price movements), although directional changes includes the concepts of overshoots as well, as it will be discussed in Section 3.

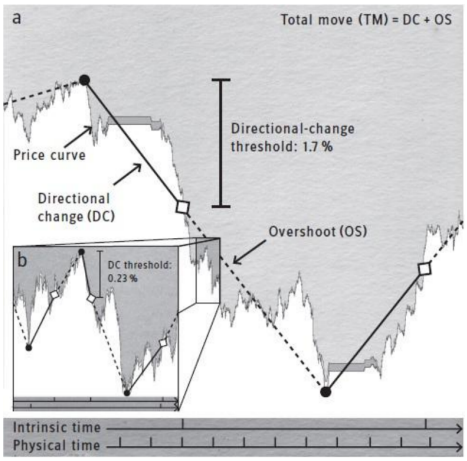


Fig. 1: Projection of a (a) two-week, (b) zoomed-in 36 hour price sample onto a set of DC events defined by a threshold (a) $\theta = 1.7\%$, (b) $\theta = 0.23\%$. The DC events (diamonds) act as a natural dissection points, decomposing a total-price move between two extremal price levels (bullets) in to DC (solid lines) and OS (dashed lines) sections (source:[2]).

The first works to use the concept of directional changes were proposed in [3] and [2]. By doing so, new empirical scaling laws in foreign exchange data series were discovered. Then, [8] was the first work to formally present all definitions related to the paradigm of directional changes. Furthermore, [9] demonstrated the effectiveness of directional changes in capturing periodic market activities. To the best of our knowledge, the above are the only works that have used the concept of directional changes. However, none of these works have used DC for generating and evaluating trading strategies. Instead, they have only focused on theoretical aspects of directional changes—e.g. discovering new empirical scaling laws. This thus motivated us to use DC to derive new trading strategies. In the following section, we present in detail the concept of directional changes.

3 Directional Changes

A directional change (DC) event is identified by a change in the price of a given stock. This change is defined by a threshold value, which was in advance decided by the trader. Such an event can be either an upturn or a downturn event. After the confirmation of a DC event, an overshoot (OS) event follows. This OS event finishes once an opposite DC event takes place. The combination of a downturn event and a downward overshoot event represents a downward run and, the combination of an upturn event and an upturn overshoot event represents an upturn run. In other words, a downward run is a period between a downturn

Algorithm 1 Pseudocode for generating directional changes events (source: [9]).

Require: Initialise variables (event is Upturn event, $p^h = p^l = p(t_0), \Delta x_{dc}(Fixed) \geq 0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} = t_0$)

if event is Upturn Event **then**
  if $p(t) \leq p^h \times (1 - \Delta x_{dc})$ **then**
    $event \leftarrow DownturnEvent$
    $p^l \leftarrow p(t)$
    $t_1^{dc} \leftarrow t$ // End time for a Downturn Event
    $t_0^{os} \leftarrow t + 1$ // Start time for a Downward Overshoot Event
  else
    if $p^h < p(t)$ **then**
      $p^h \leftarrow p(t)$
      $t_0^{dc} \leftarrow t$ // Start time for Downturn Event
      $t_1^{os} \leftarrow t - 1$ // End time for an Upward Overshoot Event
    end if
  end if
else
  if $p(t) \leq p^l \times (1 + \Delta x_{dc})$ **then**
    $event \leftarrow UpturnEvent$
    $p^h \leftarrow p(t)$
    $t_1^{dc} \leftarrow t$ // End time for a Upturn Event
    $t_0^{os} \leftarrow t + 1$ // Start time for an Upward Overshoot Event
  else
    if $p^l > p(t)$ **then**
      $p^l \leftarrow p(t)$
      $t_0^{dc} \leftarrow t$ // Start time for Upnturn Event
      $t_1^{os} \leftarrow t - 1$ // End time for an Downward Overshoot Event
    end if
  end if
end if

event and the next upturn event and an upturn run is a period between an upturn event and the next downturn event.

Figure 1 presents an example of how a physical-time price curve is transformed to the so-called *intrinsic time* [2] and dissected into DC and OS events. Furthermore, Algorithm 1 presents the high-level pseudocode for generating directional changes events.

This new concept provides traders with new perspectives to price movements, and allows them to focus on those key points that an important event took place, blurring out other price details which could be considered irrelevant, or event noise. Furthermore, DC have enabled researchers to discover new regularities in markets, which cannot be captured by the interval-based summaries [2]. Therefore, these new regularities give rise to new opportunities for traders, and also open a whole new area for research.

One of the most interesting regularities that was discovered in [2] was the observation that a DC of threshold θ is on average followed by an OS event of the

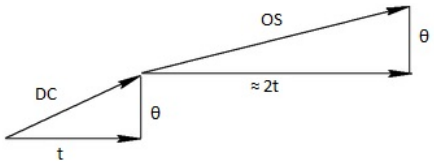


Fig. 2: An example of a scaling law presented in [2], which shows that (a) a DC event of threshold θ is followed by an OS event of also threshold θ, and (b) the OS event lasts about the double amount of time that it took for the DC event to take place.

same threshold θ. At the same time, it was observed that if on average a DC takes t amount of physical time to complete, the OS event will take an amount of $2t$. This observation is summarised in Figure 2, and *was only made under DC-based summaries*. Furthermore, this astonishing observation was made on all of the 13 different currency exchange rates that the authors of [2] experimented with (findings were published in a prestigious financial journal). This thus leads us to hypothesise that such statistical properties could lead to profitable strategies, if appropriately exploited, mainly because such properties are not well-known to traders yet. Therefore, the DC area is a rich research area that could potentially lead to significant discoveries.

4 Generating DC-based Trading Strategies

As we discussed in Section 3, a DC event is identified by a change in the price by a given threshold value. The use of different DC thresholds provides a different view of the data: smaller thresholds allow the detection of more events and, as a result, actions can be taken promptly; larger thresholds detect fewer events, but provide the opportunity of taking actions when bigger price variations are observed. In this section we present a trading strategy that combines the use of different threshold values using a GP algorithm. The aim is to automatically generate expressions that produce outputs based on multiple threshold values in an attempt to take advantage of the different characteristics of smaller/larger thresholds.

The proposed algorithm follows the standard tree-based GP configuration, summarised in Table 1. GP individuals' trees are created from nodes from a terminal and function sets. The function set is composed by boolean functions: binary functions {`AND`, `OR`, `NOR`, `XOR`} and the unary function {`NOT`}. The terminal set is composed by nodes representing the output of (randomly-generated) DC thresholds as boolean values: `TRUE` if the detected event is either an *Upward* or *UpwardOvershoot* event; `FALSE` if the detected event is either a *Downward* or *DownwardOvershoot* event. The values of the thresholds are randomly chosen at the start of the algorithm given user defined range. Figure 3 illustrates the

Table 1: Configuration of the proposed GP algorithm.

Configuration	Value
Individual structure	tree
Function set	boolean functions {`AND`, `OR`, `NOR`, `XOR`, `NOT`}
Terminal set	randomly generated boolean terminals representing different DC threshold values
Tree initialisation	ramped half-and-half
Genetic operators	subtree mutation, one-point crossover and reproduction
Selection	tournament selection
Termination criteria	maximum number of generations

structure of a GP individual tree. Note that an event is detected only after a confirmation point (end time of a downturn/upturn event); therefore, terminal values correspond to the current active event and this changes when a new event is detected.

In order to evaluate the output of an expression represented by the GP tree, the algorithm provides a price value to the terminal nodes, which enables the different threshold to detect a DC event. Based on the DC event detected, the terminal nodes output a boolean value.

4.1 Fitness Function

As each individual represents a trading strategy, the fitness function measures how good a candidate strategy is in terms of generating profit. The profit is determined by computing the amount of cash that each trading strategy could earn from the historic market activity (training data). In summary, the fitness function iterates over the training days and based on the output of the individual, takes the action of selling or buying a stock. Algorithm 2 presents the high-level pseudocode of the fitness function.

The evaluation of an individual is controlled by two parameters: the initial amount of money (cash) and number of stocks (stock balance). An individual can only buy a stock if it has available cash; similarly, it can only sell a stock if it has available stocks. The operation of buying a stock decreases the cash by the stock price and increases the stock balance by one; the operation of selling a stock decreases the stock balance by one and increases the cash by the selling price.

Recall that an individual potentially includes multiple DC thresholds—each threshold might represent different DC events given the same point in time—and its output is the result of the combination of the output of these thresholds. The rationale of using a combination of multiple thresholds is that it should help to buy or sell stocks at more favourable moments than using just a fixed threshold as an investment strategy.

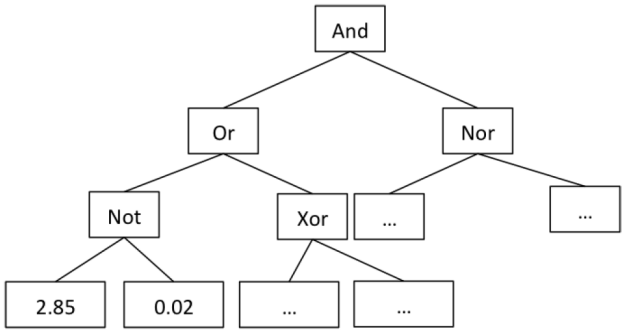


Fig. 3: A sample GP individual tree: internal nodes are represented by boolean functions, while leaf nodes correspond to different DC thresholds. Given a price value, terminal nodes output a boolean value according to the DC event detected.

Algorithm 2 Pseudocode for the fitness function evaluation of a GP individual.

Require: : Initialise variables (index = 0, cash = *initial cash*, balanceStock = 0)
for *index* < *number of training days* **do**
  buy ← *evaluate*(*individual*)
  if (buy is true **and** cash ≥ *current price*) **then**
    balanceStock ← *balanceStock* + 1
    cash ← *cash* − *current price*
  else if (buy is false **and** balanceStock ≥ 0) **then**
    balaneStock ← *balanceStock* − 1
    cash ← *cash* + *current price*
  end if
  index ← *index* + 1
end for
$fitness \leftarrow cash + (balanceStock \times last\ price)$

According to Algorithm 2, the fitness function iteratively evaluates the output of an individual by iterating through the training data. At each iteration, the current price information (data point) is used as an input for each DC threshold node. Based on the detected event, the expression represented by an individual evaluates to a boolean value that indicated the action to be taken: buy one stock at the current price (`TRUE`); sell one stock at the current price (`FALSE`). From the sale and purchase of stocks, the fitness of an individual corresponds to the profit during the training period, calculated as

$$fitness = cash + (stockBalance \times last\ price)\ , \tag{1}$$

where the *last price* is the value of a stock at the last data point. Therefore, the fitness function takes into consideration both buying and selling activities to calculate the profit of the trading strategy represented by the individual.

Table 2: Parameters of the GP algorithm used in the experiments.

Parameter	Value
Max Tree Initial Depth	5
Max Tree Depth	8
Generations	300
Population size	300
Tournament size	2
Reproduction probability	0.01
Crossover probability	0.97
Mutation probability	0.01
Elitism probability	0.01
Max DC threshold	10
Min DC threshold	0

5 Computational Experiments

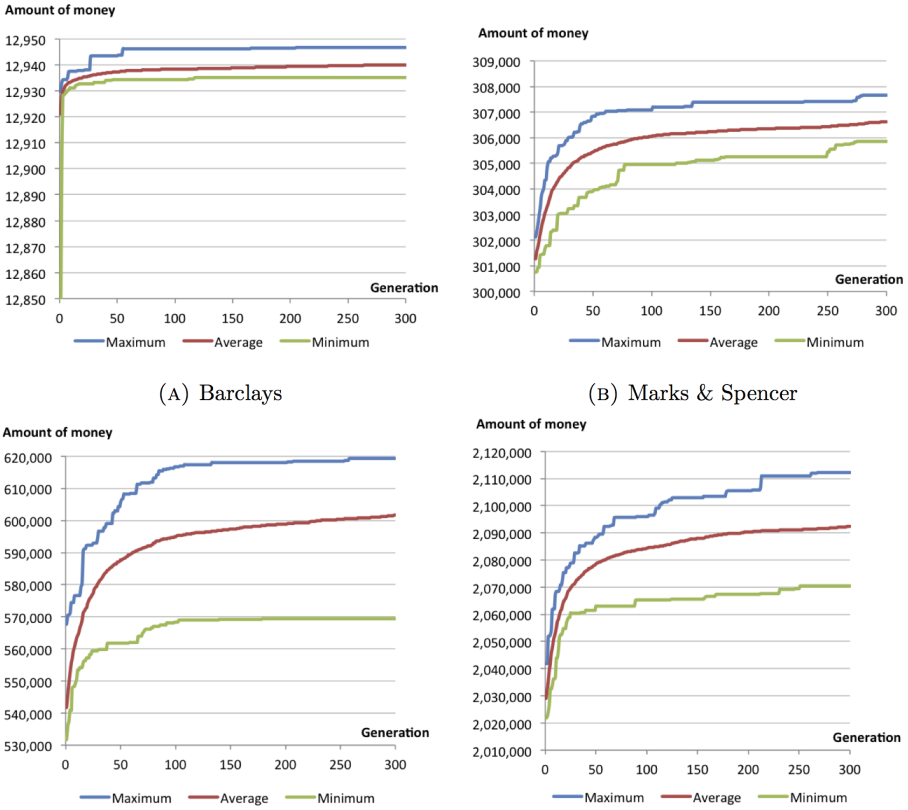
The goal of our experiments is twofold: (i) demonstrate that the paradigm of DC returns profitable strategies, and (ii) provide evidence that the strategies generated by the GP are more profitable than using a fixed threshold.

We run tests on 4 datasets, two stocks from the UK FTSE100 market (Barclays, Marks & Spencer), and two international indices (NASDAQ, and NYSE). Data consist of daily closing prices, where the training period is 1000 days long, and the testing period is 500 days. Given that stock prices are different across the datasets, the initial budget for each dataset is different—these will be presented in Subsection 5.1. The rest of the experimental parameters are presented in Table 2. The GP algorithm is run 30 times on each dataset and the results presented correspond to the average value over the 30 executions; fixed thresholds are run just once per dataset, since they represent a deterministic strategy.

Finally, we mentioned above that we will be comparing our GP results with results generated by fixed thresholds. These fixed thresholds are: 0.02, 0.05, 0.10 0.20, 0.50, 1.0, 1.50, 2.00, 2.50, 3.00, 4.00, 4.50, and 5.00. The comparison will involve two steps: first, all fixed thresholds are evaluated in the training period; then, for each dataset, the best threshold (the one that provided the highest profit in the training data) is selected to be against the GP in the test period.

5.1 Results

The average of the training fitness over 30 runs with the GP algorithm are presented in Figure 4 for the four following markets: Barclays (4A), Marks & Spencer (4B), NASDAQ (4C) and NYSE (4D). The individuals chosen for this observation were the ones that had the highest performance in the training period. We can observe that the GP search can produce more profitable solutions over the generations. Another interesting observation is that the best fitness



(A) Barclays

(B) Marks & Spencer

(C) NASDAQ

(D) NYSE

Fig. 4: Average training fitness for the Barclays, Marks & Spencer, NASDAQ and NYSE markets. The training period is 1000 days. Initial amount of cash for fitness evaluation for each dataset: (a) 7,000; (b) 300,000; (c) 500,000; and (d) 2,000,000 monetary units, respectively.

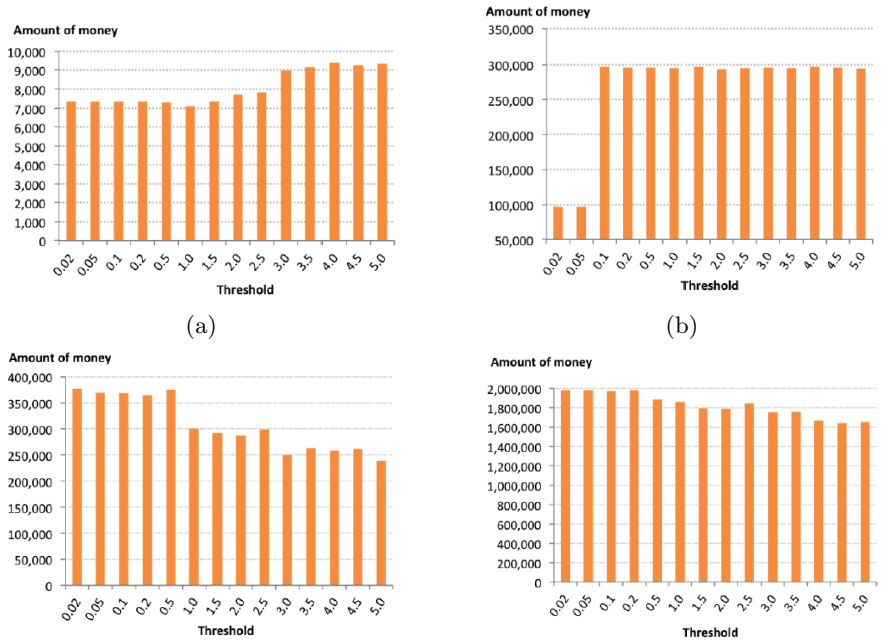


Fig. 5: Training fitness values for the different fixed thresholds for Barclays, Marks & Spencer, NASDAQ, and NYSE. The initial amount of budget for each dataset was (a) 500,000, (b) 250,000, (c) 2,000,000 and (d) 1,000,000 monetary units, respectively.

Table 3: Profit generated by the GP and fixed DC thresholds in the test data. The test period is 500 days. Initial amount of cash for fitness evaluation for each dataset: (a) 3,500; (b) 150,000; (c) 250,000; and (d) 1,000,000 monetary units, respectively. Fixed DC threshold used: Barclays (4%), Marks & Spencer (0.1%), NASDAQ (0.02%) and NYSE (0.05%).

Dataset	GP	Fixed DC
Barclays	**3,663.47**	3,007.55
Marks & Spencer	**150,832.33**	149,501.30
NASDAQ	**262,344.39**	254,450.72
NYSE	**1,073,190.19**	1,048,095.45

value (profit) is considerably higher than the initial budget and also higher than the profit generated by the use of fixed threshold values, presented in Figure 5.

In order to evaluate the trading strategy evolved by the GP, we tested the solution on a separate set of data (test data), which was not used during training. The testing period correspond to 500 days. In the case of the GP, we report the average over the 30 runs of the solution created by the GP; in the case of fixed thresholds, we selected the threshold with the highest profit in the training data for each dataset (see Figure 5)—therefore, different fixed thresholds are used for each dataset. Table 3 summarises the results obtained in the test set.

From Table 3 we can make several observations. First, the GP returns profit for all 4 datasets tested in this paper. For Barclays, there was an increase in profit by 163.47 monetary units (4.67%), for Marks & Spencer an increase by 832.33 units (0.55%), for NASDAQ an increase by 12,344.39 units (4.93%), and for NYSE by 73,109.19 monetary units (7.31%). The second observation we can make is that this is not always the case with the fixed thresholds, where Barclays and Marks & Spencer returned losses. Finally, another observation we can make is that the GP results outperform the fixed threshold results under all 4 datasets tested in this work. Therefore, these results provide evidence that the GP evolved strategies are doing more than just using a fixed threshold. Overall, we consider these results promising: the GP successfully explored the use of multiple DC thresholds to evolve trading strategies that combine different thresholds to buy/sell stocks at more favourable moments.

6 Conclusion

In this paper we presented an approach to automatically generate trading strategies using a GP algorithm. Traditionally, financial forecasting tools are based on physical time to forecast the future price movement and probe the market activity at a fix interval of time. In the proposed approach, we argue that it is better to use an intrinsic time scale to forecast the market activity through the

use of event based on Directional Changes (DC). To the best of our knowledge, DC has not been previously used to generate and evaluate trading strategies.

The proposed GP algorithm uses different DC threshold values as terminal nodes; the output of the DC nodes are combined into a single output using boolean functions. The rationale of allowing the combination of multiple thresholds is that it should help to create strategies that buy or sell stocks at more favourable moments. Computational experiments showed that the strategies evolved by the GP are more profitable than the use of a fixed threshold value, providing evidence that DC can be used for forecasting and that combining multiple thresholds is beneficial. On the one hand, the use of smaller DC thresholds detects more events, therefore actions can be taken promptly; on the other hand, the use of larger DC thresholds detects less events, but actions can be taken after observing bigger price variations.

There are several interesting research directions that can build on the approach presented in this paper. The first one is to explore different parameters settings of the algorithm with the aim of increasing the profit—e.g., at each buy/sell operation, currently only one stock is involved. There is an opportunity to identify situations where it will be better to buy/sell more stocks in the same operation. Second, we discussed that a DC event takes on average t amount of physical time to complete, followed by an OS event that takes on average $2t$. The current GP does not explore the use of OS events—there is a clear opportunity to use OS events to further improve the evolved strategies, exploring the property that a DC event is followed by a OS event with twice its duration.

References

1. International Monetary Fund: (Global financial stability report (2009))
2. Glattfelder, J., Dupuis, A., Olsen, R.: Patterns in high-frequency FX data: Discovery of 12 empirical scaling laws. Quantitative Finance **11 (4)** (2011) 599–614
3. Olsen, R.B., Muller, U.A., Dacorogna, M.M., Pictet, O.V., Dave, R.R., Guillaume, D.M.: From the bird's eye to the microschope: A survey of new stylized facts of the intra-day foreign exchange markets. Finance and Stochastics **1 (2)** (1997) 95–129
4. Neely, C.J., Weller, P.A.: Lessons from the evolution of foreign exchange trading strategies. Journal of Banking & Finance **37**(10) (2013) 3783 – 3798
5. Breedon, F., Ranaldo, A.: Intraday patterns in FX returns and order flow. Journal of Money, Credit and Banking **45**(5) (2013) 953–965
6. Chen, Y., Mabu, S., Hirasawa, K., Hu, J.: Genetic network programming with sarsa learning and its application to creating stock trading rules. In: Proceedings of the IEEE Conference on Evolutionary Computation, Singapore (2007) 220–237
7. Azzini, A., Pereira, C., Tettamanzi, A.: Modeling turning points in financial markets with soft computing techniques. In: Natural Computing in Computational Finance. Volume 293 of Studies in Computational Intelligence. Springer (2010) 147–167
8. Tsang, E.: Directional changes, definitions. (Working Paper WP050-10, Centre for Computational Finance and Economic Agents (CCFEA), University of Essex (2010))
9. Aloud, M., Tsang, E., Olsen, R., Dupuis, A.: A directional change events approach for studying financial time series. Economics **36** (2012) (online)