Evolving Directional Changes Trading Strategies with a New Event-based Indicator

Michael Kampouridis and Adesola Adegboye and Colin Johnson

School of Computing, University of Kent, UK

Abstract. The majority of forecasting methods use a physical time scale for studying price fluctuations of financial markets, making the flow of physical time discontinuous. An alternative to this is event-based summaries. Directional changes (DC), which is a new event-based summary method, allows for new regularities in data to be discovered and exploited, as part of trading strategies. Under this paradigm, the timeline is divided in directional change events (upwards or downwards), and overshoot events, which follow exactly after a directional change has been identified. Previous work has shown that the duration of overshoot events is on average twice the duration of a DC event. However, this was empirically observed on the specific currency pairs DC was tested with, and only under the specific time periods the tests took place. Thus, this observation is not easily generalised. In this paper, we build on this regularity, by creating a new event-based indicator. We do this by calculating the average duration time of overshoot events on each training set of each individual dataset we experiment with. This allows us to have tailored duration values for each dataset. Such knowledge is important, because it allows us to more accurately anticipate trend reversal. In order to take advantage of this new indicator, we use a genetic algorithm to combine different DC trading strategies, which use our proposed indicator as part of their decision-making process. We experiment on 5 different foreign exchange currency pairs, for a total of 50 datasets. Our results show that the proposed algorithm is able to outperform its predecessor, as well as other well-known financial benchmarks, such as a technical analysis.

Keywords: directional changes; algorithmic trading; financial forecasting; genetic algorithms

1 Introduction

The majority of traditional methods to observe price fluctuations in financial time series are based on physical time changes, e.g., daily data summaries. However, important price movements (and thus potential profit) might be lost due to the creation of such artificial price summaries. For example, if we are using daily prices, we would not be able to observe the 6 May 2010 Flash Crash, which was a US trillion-dollar stock market crash that lasted for approximately 36 minutes.[1]

[1] http://blogs.wsj.com/marketbeat/2010/05/11/nasdaq-heres-our-timeline-of-the-flash-crash/ Last access: 13 May 2017.

Directional Changes (DC) is based on the idea that an event-based system can capture significant points in price movements that the traditional physical time methods cannot. Instead of looking the market from an interval-based perspective, DC record the key events in the market (e.g., changes in the stock price by a pre-specified percentage) and summarise the data based on these events. Under this new paradigm, a threshold θ is defined, expressed by a percentage of the price. The market is then fragmented and summarised into upward and downward trends.

As a result of DC summaries, new market regularities have been observed. One such regularity, is the observation regarding the duration of events. Such knowledge is beneficial to traders, because it can allow them to anticipate trend reversal and thus increase their profitability margin. In this work, we exploit this regularity, by building a new event-based indicator, which predicts the expected duration of DC events. We provide more information about this in Section 3. We use this indicator as part of a genetic algorithm based trading strategy. This strategy combines multiple DC thresholds, and uses the genetic algorithm to optimise the parameters of the above multi-threshold strategy. Our goal is to show that the proposed indicator, under the DC paradigm, can lead to profitable strategies that outperform popular financial benchmarks. We test our proposed algorithm on 50 different datasets from 5 different foreign exchange (FX) currency pairs, and compare its results to a technical analysis based trading strategy, and also buy and hold.

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

2 Related Work

The first works to use the concept of directional changes were proposed in [16] and [7]. In these works, new empirical scaling laws in foreign exchange data series were discovered. These scaling laws aimed to establish mathematical relationships among price moves, duration and frequency. Then, directional changes and the scaling laws from the above works were used to develop new trading models in [6]. However, those models were not used for any financial forecasting purposes and were only used to derive statistics from potential trading. Furthermore, [1] demonstrated the effectiveness of directional changes in capturing periodic market activities. In addition, [8] presented an approach to forecasting the daily closing price of financial markets by combing directional changes and genetic programming. The work in [17] introduced new trading indicators for profiling markets under directional changes. Lastly, [2, 3, 13] were the first works that presented extensive experiments on algorithmic trading by utilising the DC paradigm. As we can observe, initial works had been focusing on theoretical aspects of directional changes—e.g. establishing mathematical relationships and developing new indicators. More recently, there have been attempts to generate trading strategies based on the DC concept. The current paper builds on these attempts, and particularly on [13], by presenting a new event-based indicator that predicts the expected duration of an event and comparing its trading performance to its predecessor's. We also compare the trading results to popular financial benchmarks. More information about the new indicator follow in the next section. First, Section 3.1 presents an overview of the DC methodology. Then, Section 3.2 presents the new indicator.

3 Directional Changes

3.1 Overview

The directional change (DC) approach is an alternative approach for summarising market price movements. A DC event is identified by a change in the price of a given financial instrument. 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.

Figure 1 presents an example of how a physical-time price curve is dissected into DC and OS events. As we can observe, two different thresholds are used, and each threshold generates a different event series. Thus, each threshold produces a unique series of events. The idea behind the different thresholds is that each trader might consider different thresholds (price percentage changes) as significant. A smaller threshold creates a higher number of directional changes, while a higher thresholds produces fewer directional changes.

Looking at the events generated by a threshold of $\theta = 0.01\%$ (events connected via solid lines), we can observe that any price change less than this threshold is not considered a trend. On the other hand, when the price changes above that threshold, then the market is divided accordingly, to uptrends and downtrends. DC events are in red lines, and OS events are in green lines. So an downturn DC event starts at Point A and lasts until Point B, when the downturn OS events starts. The downturn OS lasts until Point C, when there is a reverse in the trend, and an uptrend starts, which lasts until Point D. From Point D to E we are in an upturn OS event, and so on.

It is important to note that the change of a trend can only be confirmed retrospectively, i.e. only after the price has changed by the pre-specified threshold θ. For example, under $\theta = 0.01\%$ we can only confirm that we are in a upward trend from Point D onwards. Point D is thus called a *confirmation point*. Before Point D, the directional change had not been confirmed (i.e. the market price had not changed by the pre-specified threshold value), thus a trader summarising the data by the DC paradigm would continue believing we are in a downward trend, which started from Point A. So what becomes important here is to be able to anticipate the change of the trend as early as possible, i.e. before Points C and

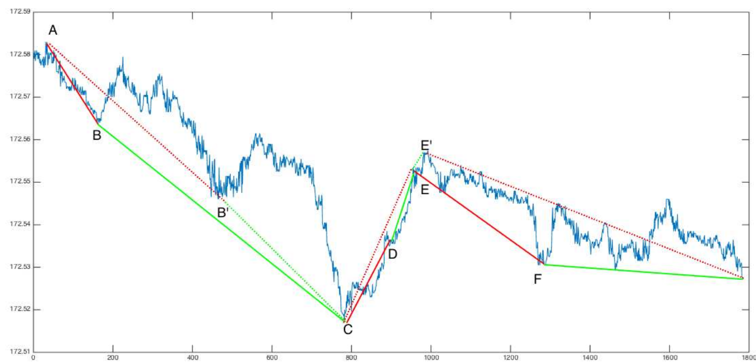

Fig. 1. Directional changes for tick data for the GBP/JPY currency pair. The solid lines denote a set of events defined by a threshold $\theta = 0.01\%$, while the dotted lines refer to events defined by a threshold $\theta = 0.018\%$. There red lines indicate the DC events, and the green lines indicate the OS events.

E have been reached. In addition, since different thresholds generate different event series, we hypothesise that the combined information from these series would lead to profitable trading strategies.

The advantage of this new way of summarising data is that it 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 even noise. Furthermore, DC have enabled researchers to discover new regularities in markets, which cannot be captured by the interval-based summaries [7]. Therefore, these new regularities give rise to new opportunities for traders, and also open a whole new area for research.

3.2 A new event-based indicator

One of the most interesting regularities that was discovered in [7] was the observation that on average a DC takes t amount of physical time to complete, the OS event will last twice, i.e., $2t$. This observation *was only made under DC-based price summaries*, and not under phycical-time summaries.

The main advantage of the above observation is the fact that we can anticipate when trend is going to reverse, since we can expect when the OS event will end. However, this observation is only an approximation and it only applies to the specific currency pairs it was tested with, and only under the specific time periods it was tested. This thus makes it inflexible and rather static. We propose to have more tailored expected OS durations, by looking into each currency pair and time period separately. Therefore, we calculate the average time of each OS event for every period and dataset we experiment with. This makes this duration indicator more dynamic, as its duration estimates adapt to each dataset we experiment with. We create two variables, expressed as the average ratio of the OS event length over the DC event length. These two variables are r_u and r_d, where r_u is the average ratio of the upwards OS event, and r_d is the average ratio of the downwards OS event.

After obtaining these ratios, we are able to anticipate the end of a trend (approximately) and as a result make trading decisions once an OS event had reached the average ratio of r_u or r_d. Of course, in reality things are not that simple. The r_u and r_d ratios are just average approximations, so many times the OS event might last longer or shorter than anticipated. In an attempt to address this issue, we have created two user-specified parameters, namely b_1 and b_2, which define a range of time within the OS period, where trading is allowed. For instance, if a trader expects the OS event to last for 2 hours, then we can define an *action* range of $[b_1, b_2] = [0.90, 1.0]$, which effectively means we are going to trade at the last 10% of the 2 hours duration, i.e. in the last 12 minutes. By introducing b_1 and b_2, we are essentially attempting to anticipate the approximation errors that might have been created during the calculation of r_u and r_d. Equation 1 presents the formulas for these starting and ending for upward and downward OS periods:

$$\begin{aligned} t_0^U &= (t_1^{dc} - t_0^{dc}) \times r_u \times b_1 \\ t_1^U &= (t_1^{dc} - t_0^{dc}) \times r_u \times b_2 \\ t_0^D &= (t_1^{dc} - t_0^{dc}) \times r_d \times b_1 \\ t_1^D &= (t_1^{dc} - t_0^{dc}) \times r_d \times b_2 \end{aligned} \quad , \tag{1}$$

where t_0^U, t_1^U are the start and end times for upwards overshoot period, respectively, and t_0^D, t_1^D are the start and end times for downwards overshoot period, respectively. In addition, t_0^{dc} and t_1^{dc} are the start and the end times of the current DC event, after the confirmation of the event has taken place at time t_1^{dc}. Their difference $t_1^{dc} - t_0^{dc}$ returns the length of the current DC event. Also, r_u and r_d are the average ratios of the upwards and downwards OS period lengths, respectively, over the current DC period. Lastly, b_1 and b_2 are the two parameters defining the action range within the OS periods, as explained above.

Although b_1 and b_2 define a window for trading, a problem that exists with high-frequency data is that there can still be hundreds of points to trade, even if that trading window is very narrow. This could be problematic, because trading at multiple price levels will not return the highest profit. What is more effective is to sell (buy) at a price as expensive (cheap) as possible. To achieve this, we introduced another variable b_3, which prevents traders from doing expensive trades. To ensure this, we only allow the system to sell at the most expensive (peak) price P_{peak} and buy at the cheapest recorded price (trough) P_{trough}, or in prices in close range. This range is determined by the value of b_3. Therefore a trader would sell when the price is equal to $P_{peak} \times b_3$, or buy when the price is equal to $P_{trough} \times (1 - b_3)$. Essentially, b_3 is a value within the range of $[0, 1]$ and defines the range of prices close to P_{peak} andP_{tough} that the system will perform an action.

4 Generating GA-based Directional Changes

4.1 Step 1: Creating a multi-threshold DC trading strategy

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. We will thus combine the use of different threshold values in an attempt to take advantage of the different characteristics of smaller and larger thresholds.

From the proposed duration indicator in the previous section, we know that under a specific threshold we should buy towards the end of a downtrend and sell towards the end of an uptrend (i.e. towards the end of the respective OS events). Since now we are dealing with multiple thresholds, each threshold summarises the data in a unique way. For example, at one point in time the trading strategy under one threshold could be recommending a buy action, while under a different threshold recommend a sell action.

In order to decide which recommendation to follow, we associate each DC threshold to an equal weight of $\frac{1}{N_\theta}$, where N_θ is the total number of thresholds used. Therefore, $W_1 = W_2 = W3 = ... = W_{N_\theta} = \frac{1}{N_\theta}$. As a result, at any point in time the trading strategy is able to make a buy/sell/hold recommendation based on the combined recommendations of all thresholds. As we already know, each threshold produces DC events; thus each threshold is able to make this buy/sell/hold recommendation. Since we have N_θ thresholds, this means that at any point in time we receive N_θ recommendations. In order to decide which recommendation to follow, we sum the weights of the thresholds: if the sum of the weights for all thresholds recommending a buy (sell) action is greater than the sum of the weights for all thresholds recommending a sell (buy) action, then the strategy's action will be to buy (sell). The hold action is a special case of both buy and sell and it happens when we are outside the price range recommended by b_3, or when there is not enough quantity to act.

In addition, the multi-threshold trading strategy is able to make recommendations on the trading quantity Q_{trade}. The decision for this quantity is a dynamic decision, taken by the number of DC thresholds that are advising to sell (buy) at a certain point in time: if many thresholds are advising to sell (buy), then the algorithm sells (buys) a higher quantity of the given currency pair. Equations 2a and 2b present the relevant formulas, for buy and sell, respectively:

$$Q_{trade} = (1 + \frac{N_\downarrow}{N_\theta}) \times Q \tag{2a}$$

$$Q_{trade} = (1 + \frac{N_\uparrow}{N_\theta}) \times Q \tag{2b}$$

where Q_{trade} is the quantity to trade, $N_\downarrow$ and $N_\uparrow$ are the number of thresholds recommending to buy and sell, respectively, N_θ is the total number of thresholds used in our experiments, and Q is a user-specified quantity, which is fixed through tuning and controls the trading quantity. As we can see, by taking into account the recommendations given by the DC thresholds, we are giving more or less weight to the Q quantity, resulting to a new quantity Q_{trade}. Lastly, it should be mentioned that our trading strategy allows short selling. However, in order to avoid excess short selling, which can lead to significant losses, we have introduced a stop loss mechanism that is called *short selling allowance*. This allowance is a percentage of our budget and allows short selling activities up to this pre-specified percentage. This percentage is decided during parameter tuning.

4.2 Step 2: Optimising multi-threshold strategies via a genetic algorithm

While the multi-threshold strategy presented above has the advantage of combining recommendations from different thresholds, a problem that exists is that we do not know how much weight we should give to each threshold and how to update them in time. Some thresholds might be more useful than others, hence we should give them more weight. Thus, we use a genetic algorithm (GA) to evolve *real values* for the weight of each DC threshold. In addition, we also evolve some other DC parameters that are crucial to the success of the trading strategy. All these are discussed next, where the GA representation, operators and fitness function are presented.

Representation Each chromosome consists of $4 + N_\theta$ genes, where N_θ is the number of different threshold values of the multi-threshold strategy. The number 4 denotes that in addition to the thresholds, there are also 4 parameters to be optimised: Q (first gene), b_1 (second gene), b_2 (third gene), and b_3 (fourth gene). Q, b_1, b_2 and b_3 refer to the DC-related parameters presented in Sections 3.2 and 4.1. A reminder that b_1 and b_2 are directly linked to the proposed duration indicator, as they control our expectations about trend reversal and the specific time period we should act. Each remaining gene in the chromosome (positions 5 to $[4{+}N_\theta]$) represents the weight associated to a given threshold.

As a result, at any point in time a GA individual is able to make a buy/sell/hold recommendation based on the combined recommendations of all thresholds by using the majority vote mechanism we presented in the previous section. An example of an 8-gene GA chromosome is presented in Figure 2.

Based on this example, the GA recommends buying/selling a quantity of Q equal to 10, and only acting in the period $[0.9, 1.0]$ of the estimated duration of the OS event (i.e., in the last 10% of the length of the OS event). In addition, the fourth gene recommends to only consider prices that are within a 20% range (the value of b_3 is 0.8, so $1.0-0.8 = 0.20$ or 20%) of the highest (lowest) recorded price P_{peak} (P_{trough}). In addition, to decide the trading action, we would check the recommendation of each individual threshold. For this example, let us assume that the first threshold recommends buy, the second threshold recommends sell, the third threshold recommends buy, and the fourth threshold recommends hold.

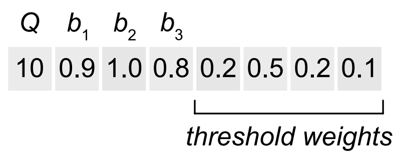

Fig. 2. An example of a 8-gene GA chromosome. The first four genes are : Q, b_1, b_2 and b_3, respectively. The remaining four genes are the weights for the DC thresholds: W_1, W_2, W_3, and W_4.

We would then sum up the weights of the thresholds, according to each action. Therefore, the weight for buying W_B is equal to $W_1 + W_3 = 0.2 + 0.2 = 0.4$, and the weight for selling W_S is equal to $W_2 = 0.5$.[2] Since $W_S > W_B$, the GA's recommendation would be to sell.

Operators We are using elitism, uniform crossover and uniform mutation.

In elitism, the best-performing individual (in terms of fitness) is copied to the next generation. In uniform crossover, two parents are selected via a tournament selection. In this type of crossover, the genes between the two parents are swapped with a fixed probability of 0.5. In addition, we ensure that the value of the third gene is always greater than the value of the second gene, i.e. b_2 always has to be greater than b_1. Lastly, for the uniform mutation operator a single parent is selected, again by tournament selection. With a probability of 0.5, each gene of the chromosome is mutated, and a different value is obtained. It should be clarified here that for the first gene (quantity Q), the mutated value can be any integer up to a pre-specified maximum quantity value; whereas for the remaining genes (i.e., b_1, b_2, b_3 and all weights W), the mutated values are real numbers randomly drawn between 0 and 1, where $b_2 > b_1$.

Fitness function Several different metrics have been used in the literature as fitness function in algorithmic trading. Some examples are: wealth, profit, return, Sharpe ratio, information ratio [4, 5]. In this paper, we set our fitness equal to the total return minus the maximum drawdown, presented in Equation 3:

$$\begin{aligned} ff &= Return - \alpha \times MDD \\ MDD &= \frac{P_{trough} - P_{peak}}{P_{peak}} \end{aligned} \quad , \tag{3}$$

where $Return$ is the return of the investment, MDD is the maximum drawdown, and α is a tuning parameter. Maximum drawdown is defined as the maximum

[2] As explained earlier, the hold action is an exceptional case that is considered as an alternative to buy and sell actions.

cumulative loss since commencing trading with the system. It is used to penalise volatile trading strategies in terms of return. Its value is given as the percentage of $\frac{P_{trough}-P_{peak}}{P_{peak}}$, where P_{trough} the trough value of the price, and P_{peak} is the peak value of the price. Lastly, the tuning parameter α is used to define how much risk-averse the strategy is. The more risk-averse in terms of wishing to avoid a catastrophic loss, the higher the value of α.

5 Experiments

We use 10-minute interval high frequency data for the following currency pairs[3]: EUR/GBP, EUR/USD, EUR/JPY, GBP/CHF, and GBP/USD. The period is June 2013 to May 2014. Every month is split into its own dataset, with the first 70% of the data being the training set, and the remaining 30% being the testing set. We should also note that r_u and r_d (ratios for OS over DC duration) are only calculated for the training period during pre-processing; the resulted values are then used during the evolution of the GA individuals.

Our goal is to demonstrate that the proposed duration indicator, under the GA-optimised multi-threshold DC paradigm, can lead to profitable trading strategies that can also outperform popular financial benchmarks. We will be presenting experimental results for two variations of the DC strategy. The first will be using the static duration indicator [7], which assumes that the OS length is on average twice as long as the DC length. We denote this as $DC{+}GA_S$. The second DC algorithm will be using the new dynamic indicator, which uses tailored OS lengths for each dataset, denoted as $DC + GA_D$. We will also be presenting results from two common financial benchmarks: buy and hold (BH), and technical analysis. For the latter, there are numerous indicators that one can use. We use a genetic programming [14] algorithm, named EDDIE, to combine different indicators and formulate trading strategies [12, 10, 11, 9]. This algorithm has shown in all of the above works its ability to generate profitable strategies.

5.1 Experimental parameters

We used the I/F-Race package [15] for parameter tuning. I/F-Race automatically configures optimisation algorithms by finding the most appropriate settings, given a set of instances of an optimisation problem. It should be noted that BH is a simple process with no parameters that require tuning.

In order to avoid biased results, we used the first two months of our data (June and July 2013) for each currency pair for tuning purposes. Thus, I/F-Race was applied to the data of June and July 2013. The remaining ten months (August 2013–May 2014) were used only with the tuned parameters, after I/F-Race was complete. At the end of the tuning process, we picked the best parameters returned by I/F-Race. These parameters constitute the experimental parameters for our algorithms. These parameters are presented in Table 1. The buy and hold setup did not have any parameters, so it is not present in Table 1.

[3] All data was purchased by OlsenData: http://www.olsendata.com

Table 1. Experimental parameters determined using I/F-Race.

Parameter	EDDIE	$DC + GA_S$ / $DC + GA_D$
Population	500	1000
Generations	30	35
Tournament size	2	7
Crossover probability	0.90	0.90
Mutation probability	0.10	0.10
Number of thresholds	N/A	5
Short selling allowance	0.25	0.25
MDD weight	0.20	0.20

5.2 Results

Table 2 presents the mean return for EDDIE, $DC+GA_S$ and $DC+GA_D$ under the 10-minute interval datasets, over 50 individual runs. We should also note that BH's average return was 0.01274%. The first observation we can make is that the DC paradigm outperforms technical analysis, as all best mean returns (boldface) come from either $DC+GA_S$ or $DC+GA_D$. EDDIE has a negative mean return of -0.00873%; it is also worth noting that for all five currency pairs EDDIE's mean return is negative. On the other hand, $DC+GA_S$ has a positive return for three currency pairs: EUR/GBP, EUR/USD, and GBP/CHF. However, overall, $DC + GA_S$'s mean return is negative: -0.00930%. This mainly because of the algorithm's very bad performance for the EUR/JPY currency pair. With respect to $DC + GA_D$, there's again 3 currency pairs with positive average returns (EUR/GBP, EUR/JPY, GBP/USD), and 2 pairs with negative average returns (EUR/USD, GBP/CHF). But these negative returns are minimal and thus, the mean return for all 5 currency pair is positive, at 0.01046. In addition, by looking into the standard deviation values, which are also presented in Table 2 inside the brackets, we can observe that $DC + GA_D$ has the lowest average standard deviation, making it the least volatile algorithm.

To further investigate the algorithms' performance, we applied Friedman's non-parametric statistical test to compare multiple algorithms. We present the results in Table 3. For each algorithm, the table shows the average rank according to the Friedman test (first column) over the 50 datasets, and the adjusted p-value of the statistical test, when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to the Hommel post-hoc test (second column). The ranks presented in the table confirm that $DC + GA_D$ has the best overall performance, with a rank of 1.40. $DC + GA_S$ ranks second, and EDDIE ranks third. However, as we can observe from the p-value Friedman test (0.1250), the test was close to reject the null hypothesis at the 10% significance level; however, the p-value was slightly higher, which means that the differences in the ranks are not statistically significant. Neverheless, the fact remains that $DC+GA_D$ was ranked first across the majority of the tests. More importantly, $DC + GA_D$ had a positive mean

Table 2. Mean return results for EDDIE, $DC + GA_S$ and $DC + GA_D$. 10-minute interval data. BH's average return (not included in the table) was 0.01274%. Results shown in % values. Best return value per currency pair is shown in bold. Standard deviation is presented inside the brackets.

	EDDIE	$DC + GA_S$	$DC + GA_D$
EUR/GBP	-0.00141 (0.007)	**0.00341** (0.008)	0.00063 (0.004)
EUR/JPY	-0.01644 (0.357)	-0.07723 (0.055)	**0.05387** (0.210)
EUR/USD	-0.00840 (0.018)	**0.02455** (0.276)	-0.00125 (0.009)
GBP/CHF	-0.01114 (0.015)	**0.00903** (0.027)	-0.00388 (0.014)
GBP/USD	-0.00628 (0.011)	-0.00580 (0.018)	**0.00293** (0.025)
Mean	-0.00873 (0.082)	-0.00930 (0.077)	**0.01046** (0.053)

Table 3. Statistical test results according to the non-parametric Friedman test with the Hommel's post-hoc test. 10-min interval data.

Friedman p-value	0.1250	
Algorithm	Average Rank	Adjusted p_{Homm}
$DC + GA_D$ (c)	1.40	-
$DC + GA_S$	2.10	0.26838
EDDIE	2.50	0.16398

return over the 50 datasets it was tested, while both of the other two algorithms had a negative mean return. This thus makes $DC+GA_D$ a much more attractive algorithm and also a promising algorithm for future experimentation. To sum up, our results demonstrate two things: (i) the DC paradigm can be a profitable one when tuned appropriately, and (ii) our proposed method of having tailored OS length estimates improves the mean return results of the trading algorithm.

6 Conclusion

To conclude, this paper presented a new tailored event-based indicator, which was used within the context of directional changes. DC is a new way of summarising physical-time data. After creating different summaries, based on different DC thresholds, we used a genetic algorithm to optimise their recommendations. Our experiments, over 50 datasets from 5 different FX currency pairs showed that our approach was able to yield positive returns in the majority of datasets tested, and outperformed both its predecessor, and also a technical analysis based trading algorithm. It also performed similarly to buy and hold.

We believe that this is a very positive result and that more research should go towards this direction. For example, it would be interesting to use a genetic programming algorithm for symbolic regression, to produce new equations describing the relationship of the length of DC and OS events. Also, we plan to test our algorithm with more datasets for generalisation purposes.

References

1. Aloud, M., Tsang, E., Olsen, R., Dupuis, A.: A directional change events approach for studying financial time series. Economics 36, (online) (2012)
2. Bakhach, A., Tsang, E., Ng, W.L., Chinthalapati, V.L.R.: Backlash agent: A trading strategy based on directional change. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (2016)
3. Bakhach, A., Tsang, E.P.K., Jalalian, H.: Forecasting directional changes in the fx markets. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (2016)
4. Brabazon, A., O'Neill, M.: Biologically inspired algorithms for financial modelling. Springer (2006)
5. Bradley, R., Brabazon, A., O'Neill, M.: Dynamic high frequency trading: a neuro-evolutionary approach, pp. 233–242. Springer (2009)
6. Dupuis, A., Olsen, R.: High frequency finance: using scaling laws to build trading models. In: James, J. (ed.) Handbook of exchange rates, chap. 20. Wiley, NJ, USA (2012)
7. Glattfelder, J., Dupuis, A., Olsen, R.: Patterns in high-frequency FX data: Discovery of 12 empirical scaling laws. Quantitative Finance 11 (4), 599–614 (2011)
8. Gypteau, J., Otero, F.E., Kampouridis, M.: Generating directional change based trading strategies with genetic programming. In: Mora, M.A., Squillero, G. (eds.) Applications of Evolutionary Computation: 18th European Conference. pp. 267–278. Springer (2015)
9. Kampouridis, M., Alsheddy, A., Tsang, E.: On the investigation of hyper-heuristics on a financial forecasting problem. Annals of Mathematics and Artificial Intelligence 68 (4), 225–246 (2013)
10. Kampouridis, M., Tsang, E.: EDDIE for investment opportunities forecasting: Extending the search space of the GP. In: Proceedings of the IEEE World Congress on Computational Intelligence. pp. 2019–2026. Barcelona, Spain (2010)
11. Kampouridis, M., Tsang, E.: Investment opportunities forecasting: Extending the grammar of a GP-based tool. Int. Journal of Computational Intelligence Systems 5(3), 530–541 (2012)
12. Kampouridis, M., Otero, F.E.B.: Heuristic procedures for improving the predictability of a genetic programming financial forecasting algorithm. Soft Computing pp. 1–16 (2015)
13. Kampouridis, M., Otero, F.E.: Evolving trading strategies using directional changes. Expert Systems with Applications 73, 145 – 160 (2017)
14. Koza, J.: Genetic Programming: On the programming of computers by means of natural selection. Cambridge, MA: MIT Press (1992)
15. Lopez-Ibanez, M., Dubois-Lacoste, J., Stutzle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration (2011), Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium
16. 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), 95–129 (1997)
17. Tsang, E., Tao, R., Serguieva, A., Ma, S.: Profiling high-frequency equity price movements in directional changes. Quantitative Finance 2016, (online) (2016)