Applied Soft Computing 43 (2016) 170–186

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Heuristic based trading system on Forex data using technical indicator rules

Murat Ozturk [a], Ismail Hakki Toroslu [a,*], Guven Fidan [b]

[a] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
[b] Argedor Information Technologies, Ankara, Turkey

ARTICLE INFO

Article history:
Received 3 July 2015
Received in revised form
22 December 2015
Accepted 27 January 2016
Available online 26 February 2016

Keywords:
Forex
Technical analysis
Technical indicator
Trading rule
Heuristic methods
Genetic algorithm
Trading system

ABSTRACT

Technical indicators are widely used in Forex and other financial markets which are the building blocks of many trading systems. A trading system is based on technical indicators or pattern-based approaches which produces buy/sell signals to trade in the market. In this paper, a heuristic based trading system on Forex data, which is developed using popular technical indicators is presented. The system grounds on selecting and combining the trading rules based on indicators using heuristic methods. The selection of the trading rules is realized by using Genetic algorithm and a greedy search heuristic. A weighted majority voting method is proposed to combine the technical indicator based trading rules to form a single trading rule. The experiments are conducted on 2 major currency pairs in 3 different time frames where promising results are achieved.

1. Introduction

Forex (FX in short) which stands for foreign exchange is the biggest financial market in the world with a daily transaction exceeding $5 trillion [1]. In FX, the currencies are exchanged simultaneously between 2 parties [2]. The participants in FX are widespread including banks, corporations, brokers/dealers, individuals, etc. EUR/USD is the most traded currency pair in FX market.

Many practitioners and scientists are closely interested in price forecasting in FX. In this context, the analysis approaches are divided into two groups: fundamental and technical analysis. Fundamental analysis deals with the macroeconomic factors to explain and forecast the changes in price. Technical analysis aims to forecast the price changes using historical market data. Technical analysis approaches can be grouped as chart analysis and technical indicator based price analysis. Chart analysis focuses on the price charts with the aim of finding recurrent patterns in price. Technical indicators transform the historical time series price data to another time series data to detect patterns, identify trends, measure volatility in price and define the relationship between price and volume.

The unstable and chaotic structure of price in FX market complicates forecast analysis. This leads to the usage of optimization methods. There are many generic heuristic methods, such as genetic algorithm (GA), simulated annealing (SA), etc. to solve optimization problems. GA is one of the most popular generic heuristic optimization method that generates solutions which evolve in time [3,4]. GA is based on evolution and genetics. Heuristic methods yield nearly but not necessarily optimal solution with reasonable computational effort and time.

In this paper, a GA based trading system which is developed using trading rules based on technical indicators is described. The system is based on testing the technical indicator based trading rules for qualification, selection among these qualified rules and combining the selected rules. GA is used in the qualification test of the trading rules. The selection of the qualified rules is realized using both GA and a greedy search heuristic. The latter one is used to generate baseline results. A weighted majority voting method is proposed to combine the trading rules. Training data is used in all these phases and the system is tested using test data. The experiments are conducted on 2 major currency pairs in 3 different time frames.

Combining technical indicator rules of various types, such as crossover and pattern based (i.e., Bollinger Bands based and divergence), is not common in the literature. There are two important challenging problems related to this combination process:

- Crossover type rules' signals can be calculated at each time instant by just using technical indicator values at that time. However, for the pattern based rules, technical indicator values of some time frame should be processed in order to discover the pattern of the signal.
- Different indicators may generate opposing decisions at the same time frame.

Our paper proposes novel approaches to overcome these two problems as follows:

- Although crossover signals are calculated for each time instance separately, it is common to observe the same signal is generated in consecutive time instances in trend behaviors. However typically, pattern signals occur just at single time instance when the required pattern has been observed. Therefore, usually pattern signals do not exhibit trend behaviors. Therefore, in our model, when a pattern corresponding to a pattern based signal has been discovered, that signal is not produced only just for that final time instance, but that signal has been assumed

* Corresponding author. Tel.: +90 3122105585; fax: +90 3122105544.
E-mail addresses: mozturk@ceng.metu.edu.tr (M. Ozturk), toroslu@ceng.metu.edu.tr (I.H. Toroslu), guven.fidan@argedor.com (G. Fidan).

http://dx.doi.org/10.1016/j.asoc.2016.01.048
1568-4946/

to be generated for a short predefined time period. This approach made it possible to produce several signals at a time instance and combine various types of signals for decision making.

- In order to overcome the second problem, first, the strengths of the rules are calculated. Then, a rule selection method is applied, and finally, a weighted combination method has been utilized to combine the decisions of various, potentially opposing rules.

We have tested our proposed rule selection and combination methods using a real data set. Especially when GA based rule selection mechanism is used, our system has produced stable and high profit for all test cases. In financial domain, it is not easy to construct a stable mechanism that always generates profit.

The rest of the paper is composed of 5 sections. Section 2 presents the related studies on using technical indicators to forecast in FX market. Section 3 introduces the fundamentals of FX and trading as well as technical indicators and trading rules. Section 4 presents and elaborates on the proposed trading system. Section 5 presents the results of the experiments conducted on 2 currencies in 3 time frames. Section 6 gives a summary of the study, discusses the results and concludes with the future directions.

2. Related work

There are numerous studies on using technical analysis and technical indicators in forecasting financial markets and specially FX market. One of the earliest study is given in [5] where a significant excess return is obtained. In the literature, many forecasting/trading models are proposed using technical indicators in combination with various machine learning methods, where GA is of special interest. The hybrid GA-technical indicators models are applied on stock data in [6–9] and on equity data in [10].

Neely et al. in [11] used GA to optimize the randomly created trading rules in their pioneering work. The trading rules are created randomly using price data in combination with basic arithmetical and logical operations and represented as trees. Following, the GA operations are applied on these trading rules and best of these rules are experimented.

In [12], the authors propose a genetic algorithm based approach to combine technical indicator rules. The proposed method uses a large number of technical indicator rules as well as construct new rules which are a linear combination of existing rules. The method searches for an optimal trading expert which is a weighted average of technical indicators by using genetic algorithm.

An evolutionary algorithms based constraint-guided method (CGM) is proposed to handle both hard and soft constraints in optimization problems in [13]. The proposed method is applied in economic problems including financial forecasting. The method uses basic technical indicators and rule connectives to construct technical indicator rules where genetic algorithm is utilized to discover best of those rules in terms of return.

In [14], technical indicators are combined using GA and reinforcement learning to form a hybrid trading system on EUR/USD intraday data. In the proposed model, the training data is split into two groups. The trading rules based on technical indicators are optimized by GA on the first group and the best of these rules are selected. The combination of the selected rules is then optimized using reinforcement learning on the second group and tested.

In [15], Dempster and Jones developed a system which uses genetic programming to combine multiple technical indicators on GBP/USD currency pair. The authors split the rules into buy and sell rules and used genetic programming to combine the individual technical indicator-based trading rules with an array of Boolean operators to form a system rule. They also used an active cash management system which exits a trade when the trading loss exceeds the user defined threshold.

A hybrid method is proposed which combines Multiple Kernel Regression and GA with the overbought/oversold indicators in [16]. Multiple Kernel Regression is applied to obtain the exchange rate predictions where technical indicators are supplied as input to the Multiple Kernel Regression. The trading rule is constructed using GA where it is applied to the results of the Multiple Kernel Regression with the technical indicator and buy/sell parameters.

Hirabayashi et al. developed a trading system which combines technical indicators using GA in [17]. The technical indicator values with upper and lower buy/sell limits of each and the trade parameters are represented in chromosomes to form a combined trading rule.

In [18], a hybrid method is proposed which combines technical indicators, Recurrent Self Organizing Map (RSOM) and Support Vector Regression (SVR) using GA. RSOM is applied to segment the training data into several regions and SVR is applied to the segmented data to form a trading rule. Trading rules are defined based on technical indicators and the RSOM-SVR rule is combined in GA to produce the best results.

A hybrid model which combines technical indicators, Artificial Neural Network (ANN) and GA is proposed by Thinyane and Millin in [19]. Firstly, trading rules based on technical indicators are defined. The buy/sell signals of the trading rules are then fed into an ANN to form a combined trading rule. Following, the combined trading rule is optimized using GA.

A combined model is proposed in [20] where technical indicators and GA are used. The trading rules which are based on technical indicators are defined in two categories: rules to open and exit position. These rules are then represented in a chromosome and optimized to adapt to give the best profits.

Brito and Oliveira proposed a method which combines technical indicators using GA in a comparative study [21]. The authors use 4 technical indicators to form 15 trading rules and they optimize all the rules individually using GA. They experiment the 15 trading rules on 9 most traded FX currencies and compare the results with a hybrid system including Support Vector Regression and Self Organizing Map.

In [22], the authors propose a method which combines technical indicators, Multiple Kernel Learning (MKL) and Differential Evolution (DE). The proposed method uses 3 different currency pairs including EUR/USD to form a trading rule applied to EUR/USD pair. MKL is used to predict changes in the target currency pair and DE is used to form a trading rule in combination with the technical indicators which is then combined with MKL.

Deng et al. developed a model based on technical indicators which are combined using GA in [23]. The authors set buy and sell parameters for 3 technical indicators and other trading parameters which is represented in a chromosome as a trading rule. The proposed model is applied to the training data and optimized. Following, the optimized trading rules are applied to the test data to obtain the performance.

In [24], the technical indicators are optimized using Artificial Neural Network (ANN) to form a combined trading system. The ANN is composed of four layers: the input layer of the trading signals based on the selected technical indicators, 2 hidden layers and one output layer. The proposed model is trained and the resultant trading rule is applied to hourly data of 9 currency pairs.

Deng and Sakurai proposed a method in [25] which combines Relative Strength Index (RSI) indicator in different time frames with GA on EUR/USD currency pair. 30 min, 1 and 2 h RSI indicator values combined with the various weight parameters are represented as chromosomes which forms the trading rules. The trading rules are then optimized with the GA methods and the best trading rule is selected to be tested.

Since our proposed model is also a hybrid GA based system, it might seem to be similar to the above mentioned GA based hybrid solutions as well. Moreover, to the best of our knowledge, our model is novel due to followings:

- Our proposed system combines a very large number of diverse set of technical indicator based rules, including complex pattern based and simple crossover rules.
- Our system has a weighted combination method which can be used to combine the decisions of rules with opposing direction.

As a result, our model has managed to generate stable profit under all different scenarios that we have tried.

3. Preliminaries: FX, technical analysis and trading

3.1. FX basics and trading

FX is a decentralized market unlike other markets such as stock market. Its decentralized structure makes it available to trade in a 24 h basis which differs from the other financial markets [2]. In FX market, various currencies are traded (exchanged) in pairs between each other.

When an individual decides to trade a currency pair using the broker's software, he/she sees two prices for each currency. The price on the left is called the bid and the price on the right is called the ask price. The bid price is the price which you can sell the base currency whereas the ask price is the price which you can buy the base currency [26]. The difference between the ask and bid prices is called spread. The smallest unit of price in any currency pair is called pip. For example in EUR/USD, the value of 1 pip is $0.0001 [2]. Some brokers/dealer also use fractional pips called pipette where 1 pip equals 10 pipettes [27].

After selecting a currency pair for trading, one should place an order to initiate a trade. An order is an instruction to the broker to take a specific transaction [2]. There are 3 primary order types: Market order, take-profit order and stop loss order. A market order is an order which is executed immediately with the current price. Take-profit and stop-loss orders are pending orders executed after a specified price level is reached which gains profit and stops loss, respectively [26].

In order to start a trade in FX, one should open a position. There are two options to open a position: either buying the base currency and selling quote currency (going long) or selling the base currency and buying the quote currency (going short) [28]. The base and quote currencies are the first and second currencies in a currency pair, respectively. For example, EUR is the base and USD is the quote currencies in EUR/USD currency pair [2]. After a trade is initiated, it can be closed by making a counter trade. As an example, if a trader goes long in EUR/USD (buy EUR and sell USD), he/she should sell EUR and buy USD to close the trade. A trader can use leverage in the trading. Leverage is the ratio which allows to trade large amount with a small amount of money [2]. For example, if one trades $1000 with a leverage 1:100 in EUR/USD, the amount of trading transaction will be $100,000 instead of $1000.

The profit/loss of a trading transaction is calculated by subtracting the final value from the initial value of the currency pair. Suppose the trader goes long $100,000 in EUR/USD with a buying price of 1.2850 and closes his position with a selling price of 1.2870. The difference is 0.0020 which is 20 pips. Because the initial position is long and price increased, the transaction is profitable and the profit is $100,000 \times 0.0020 = \$200$. Therefore the trader wants the price of the currency pair to increase when he/she is long and decrease when he/she is short in order to get profit.

3.2. Forecasting future prices: fundamental and technical analysis

Fundamental Analysis deals with the cause of market movement [29] by focusing on the macroeconomic factors that affects the prices to move higher or lower. These fundamental factors can be listed as follows [26]:

- Economic data reports
- Interest rate levels
- Monetary policy
- International trade flows
- International investment flows

On the other hand, technical analysis is the study of past market action for the purpose of forecasting future prices. Technical analysis deals with the effect of market action on future prices [29]. Technical analysis is based on the following three premises:

- Market action discounts everything: it means any factor that can affect the prices is already reflected in the price.
- Prices move in trends: the purpose of the technical analysis is to detect a price trend in the early phases of development.
- History repeats itself: technical analysis uses patterns which have shown success in the past and assumes they will work in the future [29].

Technical analysis, as stated previously, has two main approaches. Chart analysis involve detection of patterns in price charts. These patterns can be graphical formations such as double bottom, head and shoulders as well as trend lines with the support and resistance levels [30]. There are 4 main types of charts used by traders: Bar chart, line chart, candlestick chart and point and figure chart. A bar chart represents the opening, highest, lowest and closing prices (open, high, low and close in technical analysis jargon, respectively) by vertical bars. A line chart involves only the close prices as points connected to form a line. In a candlestick chart, there are two parts: the thin line (shadow) represents the price range between high and low whereas the wider portion (real body) represents the price range between open and close. If the close is higher than the open, the real body is white which shows an increase in price. If the close is lower than the open, the real body is black which shows a decrease in price [29]. A point and figure chart makes the increase and decrease in prices more visible. The 'x's shows rising prices whereas 'o's represents the declining prices in successive periods.

The other approach involves using technical indicators to forecast the future price action. Technical indicators are the vital tools of technical analysts to forecast future price trends and action. They can be used to clarify the price trend as well as measure volatility and define the interrelationship between price and volume data. They provide means to understand the past market action and use that information to predict future prices. One important advantage of technical indicators is the availability of their usage in many financial instruments including FX, stock market, futures market, etc. [31].

3.3. Technical indicators

Technical indicators consist of mathematical formula(s) which are applied to price time series data to produce another time series data. Technical indicators can be classified into three groups: trend, momentum and volatility based indicators. Trend indicators follow the price action and commonly referred as lagging indicators. Moving average and MACD are examples of trend indicators. Momentum indicators display the rate of change in price and commonly referred to as leading indicators. RSI and Stochastic Oscillator are examples of momentum indicators. Volatility based indicators are based on the rapid changes in volatility in price. Bollinger Bands and Chandelier Exit are examples of volatility based indicators [31,32].

In our trading system, 24 technical indicators are used as the basis of trading rules. These technical indicators are: Moving average, moving average envelopes, TEMA (Triple Exponential Moving Average), Bollinger Bands, %b, bandwidth, MACD (Moving Average Convergence Divergence), RSI (Relative Strength Index), Figurelli RSI, ATR (Average True Range), chandelier exit, physchological line, RVI (Relative Volatility Index), stochastic oscillator, ultimate oscillator, rate of change, DeMarker, relative vigor index, MFI (Money Flow Index), OBV (On Balance Volume), ADL (Accumulation Distribution Line), chaikin oscillator, CMF (Chaikin Money Flow) and EMV (Ease of Movement). Since the indicators in each group are similar with each other, some of the popular indicators are explained in the following sub-sections. The detailed calculations of these indicators are also included in Appendix A.

3.3.1. Moving average

Moving average is an indicator which shows the average price value in a specified period. Moving average is a lagging indicator which smooths the price data and makes the current trend more visible. There are 4 known types of moving averages: Simple, Exponential, Smoothed, Weighted. Simple moving average is the default strategy which is based on the arithmetic average of the prices [32–34].

3.3.2. Bollinger Bands, %b, bandwidth

Bollinger Bands, %b, Bandwidth are three indicators developed by John Bollinger. Bollinger Bands consists of 3 bands: a moving average of price in a specified period (i.e. the middle band) and 2 trading bands placed above and below this moving average (upper and Bollinger Bands, respectively) [35]. The calculation of upper and lower Bollinger Bands is based on the standard deviation of the price in the specified period of the moving average. The upper and lower bands widen or narrow depending on the volatility of the price [32,35].

%b is derived from Bollinger Bands indicator which aims to address the relative position of the price compared with the upper and lower bands of the Bollinger Bands indicator [35].

Bandwidth is derived from Bollinger Bands similar to %b indicator. It is used to identify the width of the distance between upper and lower Bollinger Bands; when the width is narrow, it is a sign of either an uptrend or downtrend. It is also the keystone of The Squeeze rule [35].

3.3.3. Moving average convergence/divergence (MACD)

MACD is an indicator developed by Gerald Appel in late 1970s [36]. There are 3 components used in the calculation of MACD: the shorter and the longer moving average in a specified period, signal line. The difference of shorter and longer moving averages is known as MACD line. The signal line is the moving average of the MACD line in the specified period. Additionally, MACD histogram which is developed by Thomas Aspray, is used as a visual tool; it is the difference of MACD line and signal line. The MACD line, signal line and MACD histogram are illustrated in Fig. 1.

The short and long term moving averages points out two different aspects of price: Short term moving average will reflect the price changes more rapidly while the long term moving average will make the current trend more visible. In this context, MACD indicator shows the strength and reflect the changes in the direction of the current trend [36,32].

3.3.4. Average true range (ATR)

ATR is an indicator developed by Wilder [37]. It grounds on True Range (TR). TR is defined as the greatest of the following:

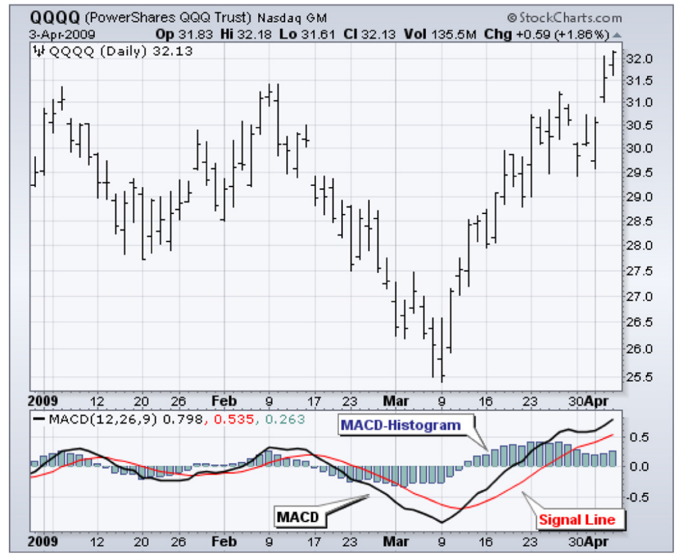

Fig. 1. MACD, signal line and MACD histogram [32].

- Current period's highest price less the current period's lowest price.
- Absolute value of current period's highest price less the previous period's close price.
- Absolute value of current period's lowest price less the previous period's close price.

TR is illustrated in Fig. 2. True Range is a means of measuring the volatility of the price; the volatility (and hence the TR) increases/decreases directly proportional to the market activity. ATR is the smoothed moving average of the TR values [32,37].

3.3.5. Relative strength index (RSI)

RSI is an indicator developed by Wilder [37]. RSI grounds on Relative Strength (RS), which is the ratio of average gain divided by average loss in a specified period. RSI is a momentum indicator which reflects the speed and changes in price. It is also used to identify the overbought/oversold levels of price. RSI values oscillate between 0 and 100; 0 indicates the price is oversold and 100 indicates the price is overbought [32,37].

3.3.6. Money Flow Index (MFI)

MFI is a volume based indicator developed by Quong and Soudack [38]. MFI values may range from 0 to 100. MFI grounds on money flow which consists of typical price and volume; typical price is calculated as the average of the close, highest and lowest

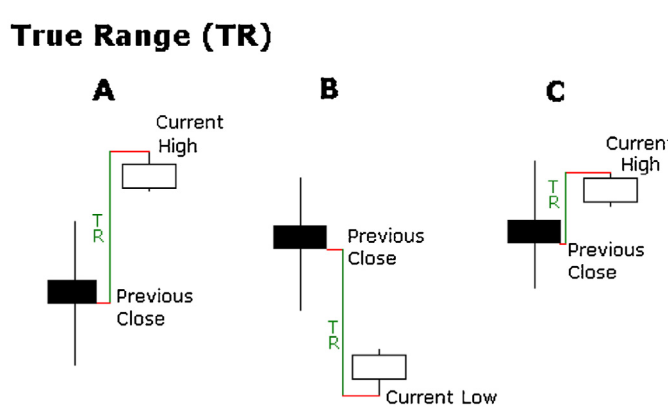

Fig. 2. True Range (TR) [32].

price in a specified period. The authors entitled MFI as volume-weighted RSI when the indicator was published for the first time; hence it has similarities with RSI. RSI is based on RS which is the ratio of average gain divided by average loss. Similarly, MFI uses Money Flow Ratio, which is the ratio of positive money flow divided by negative Money flow.

MFI can be used to detect the overbought/oversold levels of the price; values close to 100 shows an overbought level and signals a sell while values close to 0 signals an oversold level and is a sign for buy. Volume brings in an early chance to detect the overbought/oversold levels since volume leads prices [32,38].

3.4. *Trading rules*

Trading rules are the building blocks of our trading system. A trading rule is simply a rule which is based on the values of indicators and/or price in technical analysis parlance. It generates buy and sell signals according to the steps defined in. A signal is a suggestion to open a position in the market. There are 3 types of signals: Buy, sell and hold. A buy or sell signal is active which suggests to buy or sell while a hold signal is passive which means "do nothing". A trading rule may be straightforward such as comparing the indicator value with a limit value or may be complex such as looking for a special shaped pattern in the price.

A trading rule generates buy/sell signals in different fashions called trading strategy. In our trading system, two trading strategies are entitled: "always in the market" and "First buy, then sell". In "always in the market" strategy, a buy/sell signal is followed by a closing sell/buy signal (i.e. the opposite of the first signal), where the last signal does not only close the trade but also opens a new position. As an example, suppose DeMarker Crossover generates a buy signal. The next sell signal closes the active trade and also opens a short position (i.e. as a result of sell order). The second type of trading strategy "First buy, then sell" is rather simple. In "First buy, then sell" strategy, a buy signal should be generated to open a trade which is followed by a closing sell signal.

In our trading system, 38 trading rules based on 24 technical indicators are used. These trading rules are divided into 3 groups: crossover rules, rules based on Bollinger Bands, %b and bandwidth indicators, divergence rules. All three groups are explained in the following sub-sections. Furthermore, some trading rules among each groups are elaborated in detail in Appendix B.

3.4.1. *Crossover rules*

Crossover rules are trading rules which appear in two situations:

- When a time series (either indicator or price) crosses above/below another time series (either indicator or price).
- When an indicator time series crosses above/below a predefined threshold/limit value.

In our trading system, 22 trading rules are crossover rules. These trading rules are: Moving average price, double moving average, triple moving average, moving average envelopes, TEMA, MACD, RSI, Figurelli RSI, chandelier exit, physchological line, RVI, stochastics oscillator, rate of change, DeMarker, relative vigor index, MFI, OBV, ADL, chaikin oscillator, CMF and EMV crossovers. Among them, the following 6 rules differ in that they use volume information: MFI, OBV, ADL, Chaikin Oscillator, CMF and EMV Crossovers. In FX, volume information is not exactly known like other markets such as stock market, gold/valuable market, etc. The reason is that FX has a decentralized market structure which makes it almost impossible to have the full volume information. On the other hand, many FX brokers provide tick volume. Tick volume is a term which denotes the number of changes in price in a specified period. Tick volume is generally used as volume information by traders in FX. Besides tick volume, 1-day ATR indicator values can be used as volume explained in [39]. In our trading system, tick volume and 1-day ATR values (entitled as ATR volume) are the choices provided as volume in the aforementioned rules. In detail, there are 5 choices for the selection of volume: tick volume (alone), ATR volume (alone), arithmetic mean, harmonic mean and geometric mean of ATR and tick volume.

In all the crossover rules, "always in the market" is the preferred trading strategy.

3.4.2. *Rules based on Bollinger Bands, %b and bandwidth indicators*

The trading rules based on Bollinger Bands, %b and Bandwidth indicators differ from other indicator rules in that they are rather complex and include various patterns. There are 6 rules to be covered in this section: W-Type Bottom Pattern and M-Type Top Pattern are based on Bollinger Bands, Method III-Reversals, %b-MFI and %b(CMF) Crossover are based on %b and The Squeeze and Expansion is based on the Bandwidth indicator.

3.4.3. *Divergence rules*

Technical indicators of type momentum generally reflect and move together with price. Divergence occurs when a significant price movement in a direction (upward or downward) is not confirmed with the indicator movement in the same direction [31,40]. Technically, if the price is trending in upward direction, the highest level of price increases as the trend continues; thus price makes higher highs. Similarly, if the price is trending in downward direction, the lowest level of price decreases as the trend continues; thus price makes lower lows. In this case, the indicator in consideration should make higher highs or lower lows together with the price trend. If this is not the case, this means indicator and price are diverging from each other and is called a divergence. Generally, divergences are classified into two types: regular, hidden. Regular divergence signals the end of a current trend [27,40]. Regular divergence occurs in two cases:

- Price is making lower lows, but the indicator is making higher lows. It means, as the lowest level of price decreases, the lowest level of indicator increases. In this case, a regular bullish divergence occurs.
- Price is making higher highs, but the indicator is making lower highs. It means, as the highest level of price increases, the highest level of indicator decreases. In this case, a regular bearish divergence occurs [27].

On the other hand, hidden divergence confirms the current trend. Hidden divergence occurs in two cases:

- Price is making higher lows, but the indicator is making lower lows. It means, as the lowest level of price increases, the lowest level of indicator decreases. In this case, a hidden bullish divergence occurs.
- Price is making lower highs, but the indicator is making higher highs. It means, as the highest level of price decreases, the highest level of indicator increases. In this case, a hidden bearish divergence occurs [27].

Regular bullish and hidden bullish divergences are illustrated in Fig. 3a and b, respectively [27].

4. Trading system

In this section, we present our hybrid trading system which uses GA and a weighted majority voting method to select and combine

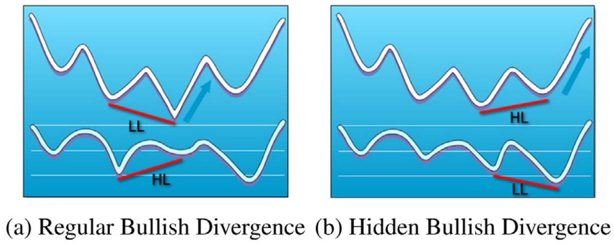

Fig. 3. Regular and hidden bullish divergences [27]. HL and LL stand for higher low and lower low of prices, respectively.

trading rules based on technical indicators discussed in the previous section to generate buy/sell signals. Also, as a baseline system, we propose a simple greedy search heuristic to replace GA based rule selection mechanism. The trading system uses time series price data of any currency pair.

The motivation behind combining trading rules instead of using those rules individually comes from the technical analysis domain. When a trading rule generates a buy/sell signal, the analyst wants to be confident about the generated signal. For this purpose, the analyst checks whether a second signal (preferably based on a different approach) confirms the first rule's decision. A parallel approach in machine learning domain is also observed with the ensemble methods where various types of classifiers are combined to obtain better performance than any constituent classifier used alone.

Our proposed trading system is composed of two phases: in the first phase, each trading rule is tested for qualification and in the second phase, high quality rules among the qualified rules are selected and combined. Training data is used in the construction of the trading system.

In the first phase, the best parameter(s) of each rule explained in Section 3.4 which satisfy the predefined criteria are obtained using GA. The output of the first phase is the set of trading rules with the best parameter(s) which satisfy the predefined criteria.

In the second phase, among the qualified rules obtained from the first phase, a subset of them is selected for decision making. The trading rules to be combined are selected among qualified rules using GA and a greedy search heuristic. The combination of the qualified trading rules is realized with an algorithm based on the weighted majority voting of the constituent rules.

An overview of the framework of the proposed system is illustrated in Fig. 4. In the following sub-sections, the phases of the framework are explained in detail.

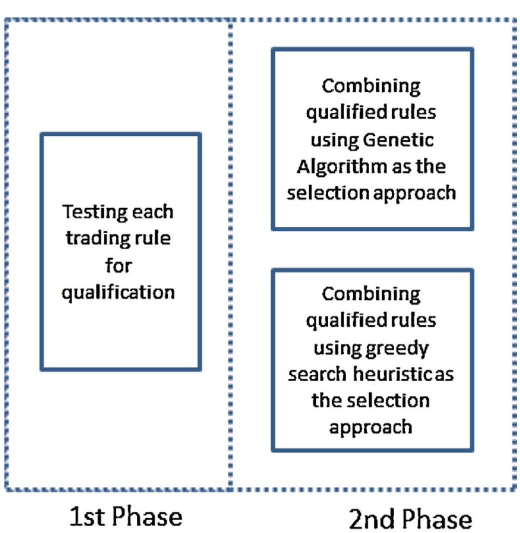

Fig. 4. The framework of the overall trading system.

4.1. Phase 1: testing each trading rule for qualification

The objective in this phase is testing and determining the trading rules to qualify for the second phase. There are 2 modules used in this phase: Trading Simulation and GA Modules. The complete process is given in Fig. 5. In the figure, for each trading rule discussed in Section 3.4, GA module randomly generates chromosomes which represent different parameter combinations of a trading rule. Then, all of these candidate rules with different parameter combinations are simulated over the training data using Trading Simulation module. The output of the Trading Simulation module is Net Profit which is provided as the fitness value to the GA module. After that, the best chromosome having the highest net profit is examined whether it satisfies the predefined criteria. The predefined criteria are thresholds for number of trades and average profit per trade (details given in Section 5.3) which are defined by the user. The rules which do not satisfy the thresholds for these criteria are eliminated. The output of this phase is the set of qualified trading rules with their best parameter(s) considering the net profit values.

4.1.1. Trading simulation module

This module is used to simulate any trading rule on the given time series data to generate buy/sell signals and calculate net profit (in pipettes) as well as other statistics (hit rate, average profit, etc.) at the end of the simulation. In our proposed system, the net profit of the simulated trading rule is used as fitness value for GA module. The simulation and calculation of net profit of a trading rule is given in Algorithm 1.

Algorithm 1. Pseudo code of the trading simulation and calculation of net profit of a trading rule.

Require: *trsignals*:trading rule buy/sell signals data, *price*:price data, *i*:current index
prevSignal : = *null* // The type of the previous signal
prevPrice : =0 // The previous price value
netProfit : =0 // The final net profit
i : =0
while $i \le |trsignals|$ **do**
  if $trsignals[i]$ = "*Buy*" $\wedge$ *prevSignal* = "*Sell*" **then**
    $netProfit := netProfit + (price[i+1] - prevPrice)$
    prevSignal:="Buy"
    $prevPrice := price[i+1]$
  els if $trsignals[i]$ = "*Sell*" $\wedge$ *prevSignal* = "*Buy*" **then**
    $netProfit := netProfit + (price[i+1] - prevPrice)$
    prevSignal:="Sell"
    $prevPrice := price[i+1]$
  end if
    $i := i+1$
end while
return *netProfit*

In Algorithm 1, when the trading rule's signal is a buy/sell signal in the current time period, the previous signal is checked whether it is the opposite of that signal. If the previous signal is opposite of the current period's signal, the trade is closed by subtracting the previous signal's price value from the next time period's price value. Next period's price value is used in the calculation since a buy or sell instruction in real market conditions cannot be executed in the same time with signal generation. The buy/sell strategy is assumed as "always in the market" in the simulation. If the succeeding signals are of the same type, they are not considered in the profit calculation. Also, if the signal is neither buy nor sell, it is implicitly a hold signal and no action is taken.

4.1.2. GA module for parameter selection

In GA, a chromosome represents a solution in a population which consists of genes having alleles (values). In GA, population evolves in time depending on a specified fitness function. The population evolves until the population converges to produce an optimal/near optimal solution. Selection operator is used to select the chromosomes for reproduction. Crossover operator exchanges

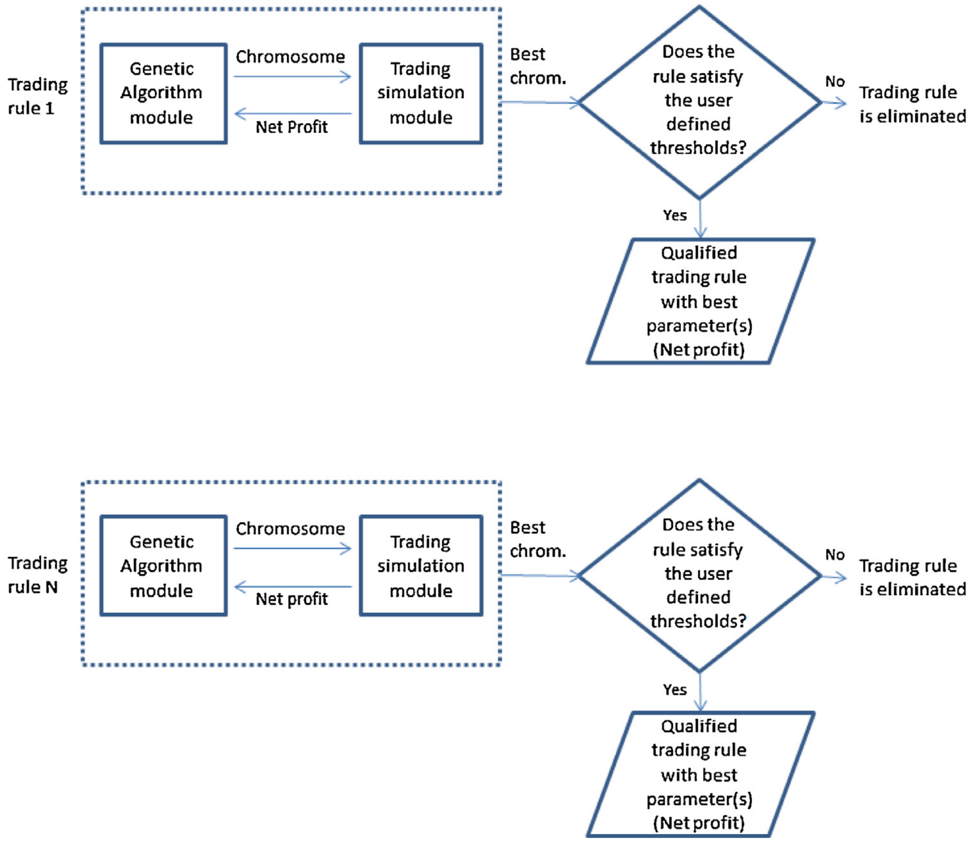

Fig. 5. Testing each trading rule for qualification.

the sub-sequences between chromosomes to generate new chromosomes. Mutation operator randomly changes the value of a gene in a chromosome [3,4].

In our work, the GA module is implemented using a framework for heuristic and evolutionary algorithm called HeuristicLab [41]. In this phase, GA module works in cooperation with Trading Simulation Module to find the best parameter(s) of each trading rule. The steps how GA module works are given as follows:

- Firstly, the possible parameter values in each gene of a chromosome is discretized to be represented by integers and the range of the values are defined. The chromosome representation of RSI crossover rule is illustrated in Fig. 6 as an example.
- Following, the chromosomes which represent the parameter combinations of a trading rule are randomly generated to form an initial population.
- Next, the fitness value of each chromosome is calculated and sent by Trading Simulation Module.
- In the main loop of GA, the following steps are repeated until a predefined number of offsprings are created:
  - Two candidate chromosomes are selected from the current population.
  - Selected chromosomes are crossed over as defined in [42] to form two offsprings.
  - Mutation is applied to the created offsprings with a defined mutation probability. Following, the mutated offsprings are placed in the population.
  - The fitness values of the crossed and mutated offsprings are calculated and sent by Trading Simulation Module.
  - The current population is replaced with the new one and the process starts over.

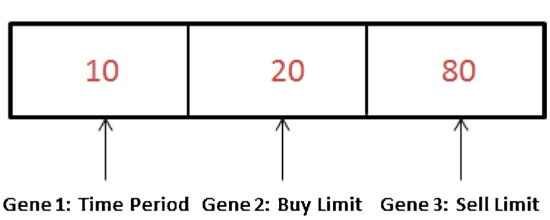

Fig. 6. Chromosome representation of RSI crossover rule.

In order to tune GA parameters, we have made several experiments with various number of parameter values. Since fitness calculation in our problem requires applying selected decisions on a time frame for a chosen training data set, fitness calculation takes significant amount of time for each chromosome. Therefore, in our problem, both the quality of the solution and the time to execute the GA has been very important. In this respect, we have tried various population sizes which range between 100 and 500. Moreover, we have incremented number of elites up to 5. Each time, the best known optimal solution has converged to a stable value and we have not seen any significant improvement. As a result, we have found the following parameter settings for GA quite satisfactory for our problem:

- Population size is 100.
- Number of elites is 1.
- Mutation probability is %5.

The resultant parameters (running time, number of iterations) of the experiments with GA are addressed in the experiments section.

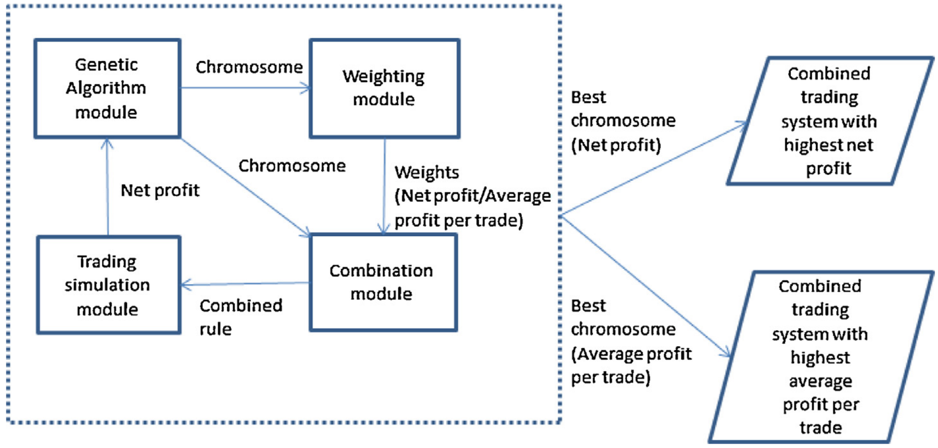

Fig. 7. Combining rules using GA as the selection approach.

The fitness metric is net profit for GA module. The output of this module is a chromosome which has the best fitness value found so far.

4.2. Phase 2: selecting and combining the qualified rules

The output of the first phase is the set of qualified trading rules with best parameter(s) in terms of net profit satisfying the criteria discussed above. In this phase, the qualified rules in the first phase are combined to form a single trading system which represents each constituent rule. There are two tasks in this phase: the selection of the trading rules to be combined and combining the selected rules. Two approaches are used for the selection of the trading rules: GA and a greedy search heuristic where these approaches are implemented in GA and greedy search heuristic modules, respectively. In order to combine the selected rules, a weighted majority voting approach is used which is implemented in Combination Module. Both two tasks, namely selection and combination of the trading rules, require the trading rules to be weighted in terms of their Net Profit/Average Profit Per Trade. The weighting process is performed by the weighting module.

The combined trading system will be different depending on the selection approach. Therefore in this phase, two alternatives which differ in selection approach are proposed to implement the trading system. These alternatives are illustrated in Figs. 7 and 8.

In Fig. 7, GA is the approach to select the trading rules for combination. Firstly, GA module randomly generates candidate chromosomes (solutions) where each of them includes different rule selections. Following, these solutions are sent to weighting module to obtain the weights of each rule in the solution. There are two alternatives to calculate the weights: Net Profit and Average Profit Per Trade. Next, the chromosome and the weights of the rules in that chromosome are sent to Combination module to form the combined trading system. The output of this module is the combined rule which is evaluated by Trading Simulation module for its performance. These processes will loop until the best chromosomes having the highest the Net Profit and Average Profit Per Trade are obtained. The outputs are the combined trading systems having the highest Net Profit and Average Profit Per Trade.

In Fig. 8, greedy search heuristic is the approach to select the trading rules for combination. Firstly, greedy search heuristic module sorts the trading rules in decreasing order considering their Net profit/Average profit per trade. Then, it generates a solution including only the best trading rule at first. Then, in each step it incrementally includes a trading rule to generate a new solution. The weights of each rule in each solution are obtained by weighting module considering their Net Profit/Average Profit Per Trade. The solutions with the weights are sent to Combination module to form the combined trading systems. The outputs of this module are the set of combined trading systems considering Net profit/Average profit per trade as the weights.

In the following paragraphs, GA, weighting, combination and greedy search heuristic modules will be explained in depth.

4.2.1. GA module for trading rule selection

Differently from the first phase, GA module is used to select the rules to be combined in this phase. The steps how GA module works is given as follows:

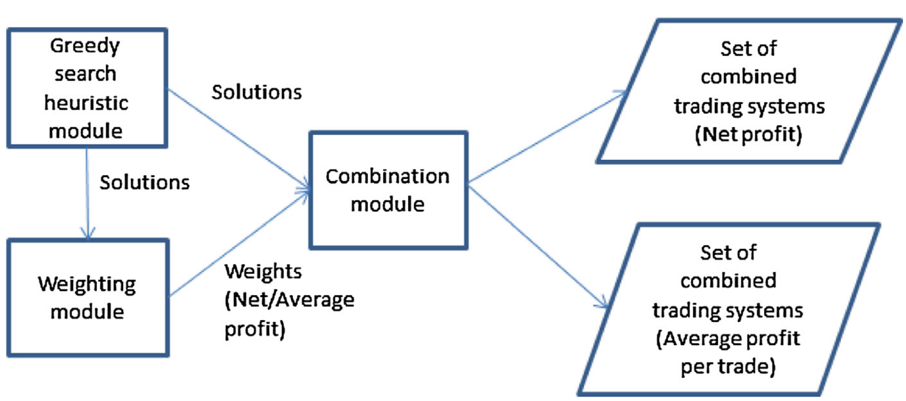

Fig. 8. Combining rules using greedy search heuristic as the selection approach.

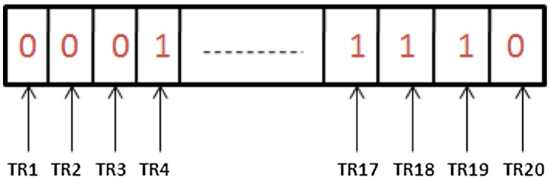

Fig. 9. Chromosome representation of a candidate combined rule.

- Firstly, the possible parameter values in each gene of a chromosome is defined to be either 1 or 0 which denotes whether the trading rule is selected to be combined or not. The representation of a chromosome with 20 trading rules is illustrated in Fig. 9 as an example.
- Following, the chromosomes are randomly generated to form an initial population.
- Next, each chromosome is sent to the Weighting and Combination Modules for the purpose of combining the selected rules in the chromosome.
- The fitness values of each candidate combined rules represented by chromosomes are calculated and sent by Trading Simulation Module.
- The main loop of the GA module is the same as given in Section 4.1.2.

The GA parameter settings are also the same as given in Section 4.1.2. The resultant parameters of the experiments with GA for this phase are similarly addressed in the experiments section.

The fitness metric is either Net Profit or Average Profit Per Trade for GA module depending on the choice. The GA module runs until all the chromosomes converge to a common solution or user interruption. The output of this module is the chromosome which has the best fitness value found so far and represents the final combined system.

4.2.2. Greedy search heuristic module

In this module, the trading rules to be combined are selected using a greedy search heuristic approach. In this approach, the trading rules are sorted in decreasing order in terms of either their Net Profit or Average Profit Per Trade. After that, the rules to be combined are selected starting from the trading rule having highest Net Profit/Average Profit Per Trade value. In each step, the rule with the next highest value is selected in a cumulative manner. In the final step, all the trading rules are selected for the combination. The process is illustrated in Fig. 10. In the figure, TR_{max1} through TR_{maxN} represent the trading rules having the highest through lowest Net Profit/Average Profit Per Trade values.

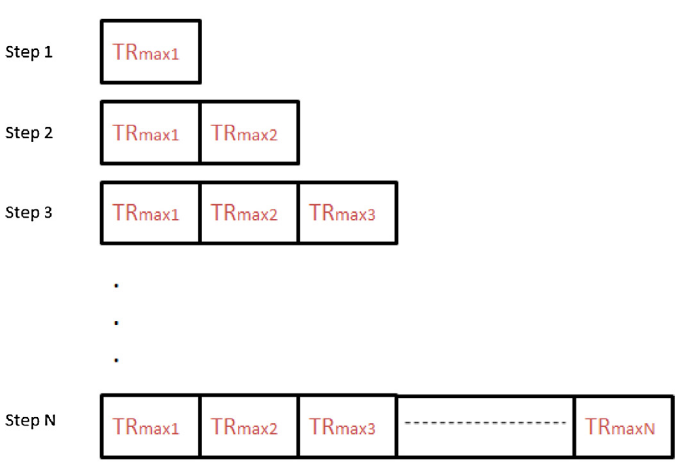

Fig. 10. Selection of trading rules with greedy search heuristic.

4.2.3. Weighting module

This module calculates and assigns weights to the selected rules to be combined either by using Net Profit or Average Profit Per Trade. The weighting process is performed by firstly summing up Net Profit/Average Profit Per Trade values of all the selected rules. Then, each rule's Net Profit/ Average Profit Per Trade value is divided by the total and multiplied by 100. The final values are assigned as the weights of the rules. In the experiments, three thresholds are used which are 25, 50 and 75.

4.2.4. Combination module

This is the key module which combines the selected trading rules to generate buy/sell signals similar to a single trading rule and forms the combined trading system. A weighted majority voting method is used in this module. The whole process is given in Algorithm 2.

In Algorithm 2, the selected trading rules' signals data is used to decide the combined rule's buy/sell decision. The weight total of buy and sell are kept separately. The weight of any trading rule is accumulated to the related total (buy or sell) whenever one or more trading rule generates a buy/sell signal. When the buy or sell weight total is greater than the user defined threshold value, a buy/sell signal is generated and the related total (buy or sell) is set to zero. Successive buy or sell signals are not considered; so if a buy signal is generated by the trading system, it is followed by a sell signal. The buy/sell strategy of the system is "always in the market" which is explained in Section 3.4.

In order to clarify the combination module, an illustration of combining 3 trading rules based on Bollinger Bands, RSI and moving average indicators is given in Fig. 11. In the figure, yellow lines denote the upper, medium and lower bands of Bollinger Bands, red line denotes the moving average, blue line denotes the RSI indicator, and green line denotes the EUR/USD price. The trading rules taken into consideration are W-type bottom pattern, moving average price crossover and RSI crossover. In the figure, the yellow circles show the successive steps of the W-type bottom pattern. The timezone inside the yellow rectangle represents a buy signal

Fig. 11. An illustration of combining trading rules.

for W-type bottom pattern. Similarly, the timezone inside the red and blue rectangles also represent a buy signal for moving average price and RSI crossover, respectively. These 3 trading rules are combined by intersecting the 3 rectangles and the system produces a buy signal for the timezone in the intersection area since all the trading rules agree for a buy signal in this timezone.

As illustrated in this example, the process of determining the occurrence of pattern based rules is a very complex task. Our system handles a number of such complicated rules.

5. Experiments

5.1. Experimental environment

The experiments are conducted on a PC with a 2.20 GHz quad-core processor, 4 GB memory and 500 GB disk space.

The proposed trading system is implemented in C#. MySql version 5.6.13 [43] is used to store the data sets used in the experiments. HeuristicLab version 3.3.10.11173 [41] is used as the GA tool.

5.2. Investment conditions

In all the experiments,

- The unit used to calculate the profit/loss of a trade is pipette (explained in Section 3).
- The initial capital is $10000.
- The investment amount for each transaction (either buy or sell) is $1000.
- The amount of leverage (explained in Section 3) used is 1:100. Therefore, the investment amount is 100 × 1000 = $100, 000 for each transaction.
- The trading strategy is "always in the market" (explained in Section 3.4) for the proposed trading system.

Considering above conditions, the profit/loss calculation of a trade is given in the following example. Suppose the trading system goes long in EUR/USD with a buying price of 1.25603 and closes its position with a selling price of 1.25690. The trade is profitable since the opening position is long and the price is increased. The profit of the trade is 1.25690 − 1.25603 = 0.00087 which is 87 pipettes. Since each transaction is $100,000 (due to leverage), the net profit is 100, 000 × 0.00087 = $87. Therefore, 1 pipette profit/loss in a trade is equal to $1 profit/loss in the experiments.

5.3. Evaluation metrics

Experimental results are obtained using 6 evaluation metrics. These metrics are:

- **Number of trades:** the total number of trades where 1 trade includes 2 transactions either by closing a long (first buy then sell) or short (first sell then buy) position.
- **Gross profit:** the total amount of profits including the profits of all the profitable trades in pipettes.
- **Gross loss:** the total amount of losses including the losses of all the unprofitable trades in pipettes.
- **Net profit:** the difference of gross profit and gross loss in pipettes.
- **Average profit per trade:** the average ratio in pipettes which is obtained by dividing net profit by number of trades.
- **Profitable trades (%):** the percentage of the number of profitable trades among all trades.

Algorithm 2. Pseudo code of trading rules combination.

```
Require: trsignals:array of each trading rules buy/sell signals
data,metric:either Net Profit or Average Profit Per Trade, weights:array of
weights of trading rules according to the given metric, threshold:threshold to
generate buy/sell signal,idata:current data index,itr:trading rule index
  buyFlag : = False
  Initialize buySellSignals : = null        // The array keeping the buy/sell signals
  buyTotal : =0                             // total weight value of buy signals
  sellTotal : =0                            // total weight value of sell signals
  idata : =0
  while idata ≤ max(|trsignals|) do
    for all itr in trsignals do
      if itr[idata] = "Buy" then
        buyTotal:=buyTotal + weights[itr]
      els if itr[idata] = "Sell" then
        sellTotal:=sellTotal + weights[itr]
      end if
    end for
    if buyFlag = True then
      if buyTotal > threshold then
        buySellSignals[idata]:="Buy"
        buyTotal : =0
        buyFlag : = False
      end if
    els if buyFlag = False then
      if sellTotal > thresholdthen
          buySellSignals[idata]:="Sell"
        sellTotal : =0
        buyFlag : = True
      end if
      end if
    end while
    return buySellSignals
```

5.4. Data sets

1-Min EUR/USD and GBP/USD time series close price data sets are used in the experiments. The data sets cover the January 2013–January 2014 (January 2014 excluded) period for EUR/USD and January 2014–July 2014 (July 2014 excluded) for GBP/USD. The data sets are obtained from Metatrader 5 software [33].

Three experiments are conducted on these data sets:

- Experiments on EUR/USD between January 2013–January 2014 (January 2014 excluded).
- Experiments on EUR/USD between January 2013–July 2013 (July 2013 excluded).
- Experiments on GBP/USD between January 2014–July 2014 (July 2014 excluded).

The data sets are split into two sections: training and test data sets. Training data is used to combine the trading rules. Following, the performance of these combined rules are experimented using test data. The ratio of number of data points of training to test data set is 2:1 in the experiments.

5.5. Experiment structure

The experimental results are obtained using:

- GA and greedy search heuristic as the rule selection approaches.
- Net profit and average profit per trade as the weight calculation metrics.
- 25, 50 and 75 as the thresholds to generate buy/sell signals.

which are discussed in Sections 4.2.1 and 4.2.3.

In greedy search heuristic approach, the qualified trading rules are sorted in decreasing order considering their Net profit/Average profit per trade values as stated in Section 4.2.2. The selection is performed starting from the best and incrementally adding a rule in each step through the worst. The rules having the best and worst performance are entitled as 1st and nth considering n qualified rules. Selecting the best rule only in greedy search heuristic approach gives the same results by using the best rule individually. The results of greedy search heuristic experiments are given in summary including the following selections:

- 1st and 2nd rule
- 1st through $(n/2)$th rule (starting from the best through middle)
- 1st through $(n-1)$th rule
- 1st through nth rule (all rules)

The resultant parameters (i.e. number of solutions and running time) of GA for parameter and rule selection are given in a table for each particular experiment.

The results of the experiments are summarized in the following subsections where net profit is shown only among the evaluation metrics. In the experiment results, the unit of net profit values is pipette. The experimental results of three individual qualified rules (indicated with their names) which performed the highest performance in training phase are given to compare the trading system's performance with the individual rules.

5.6. Experiments on EUR/USD 1 year data

The data set covers 1-min time series data between 01.01.2013 and 31.12.2013. The total number of data points in the data set is 368,440. The training data set includes data between 01.01.2013 and 31.08.2013 with a total of 246,792 data points. The test data set includes data between 01.09.2013 and 31.12.2013 with a total of 121,648 data points.

In Table 1, GA resultant parameters for parameter and rule selection are given.

28 out of 38 trading rules are qualified for the second phase. The list of qualified and unqualified rules are given in Table 2.

Table 1
Resultant parameters of GA for parameter and rule selection (EUR/USD 1 year data).

	Total running time (in seconds)	Number of generations
Parameter selection	483	87,342
Rule selection[a]	2080	14,158

[a] Average total running time and number of generations are given.

Table 2
Qualified and unqualified rules (EUR/USD 1 year data).

Rule type	Rule name	
	Qualified rules	Unqualified rules
Crossover	Double moving av., Triple moving av., Moving average env., RSI, Figurelli RSI, Chandelier exit, Psychological line, RVI, Stochastics osc., Ultimate osc., Rate of change, DeMarker, Rel. vigor index, MFI, Chaikin Osc., EMV	Moving average price, TEMA, OBV, MACD, ADL, CMF
Bollinger Bands, %b and Bandwidth Ind. Based	Method III-reversals, %b-MFI	W-type bottom pat., M-type top pat., %b(CMF) crossover, The squeeze and expansion
Divergence	Bullish (RSI), Bearish (RSI), Bullish (Figurelli RSI), Bearish (Figurelli RSI), Bullish (Ultimate oscillator), Bearish (Ultimate oscillator), Bullish (DeMarker), Bearish (DeMarker), Bullish (MFI), Bearish (MFI)	–

Table 3
Experiment results using net profit (EUR/USD 1 year data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	6780	4022	2614
	1st–14th	11,709	7561	−1648
	1st–27th	6012	5318	4191
	1st–28th	5262	10496	2481
Genetic algorithm	12, 17, 17[a]	8998	2534	3880
Best rule	Bullish div. (Fig. RSI)		4619	
2ndBest rule	RSI cross.		5747	
3rd Best rule	%b-MFI		3617	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

Table 4
Experiment results using average profit per trade (EUR/USD 1 year data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	1807	158	4658
	1st–14th	7488	5094	15942
	1st–27th	9496	7236	477
	1st–28th	14,584	977	9013
Genetic algorithm	14, 20, 13[a]	7660	7412	2014
Best rule	Bullish div. (Fig. RSI)		4619	
2nd Best rule	Bullish div. (DeMarker)		968	
3rd Best rule	Bullish div. (RSI)		−98	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

The results of the experiments using net profit and average profit per trade as the weight calculation metrics are given in Tables 3 and 4, respectively.

The results in Table 2 indicate that 16 out of 22 crossover rules, 2 out of 6 Bollinger Bands and related indicators based rules and all the divergence rules successfully qualified for the second phase.

In Table 3, the best results are achieved for GA and greedy search heuristic methods when the threshold value is 25. Greedy search heuristic method produced a negative result but all the other results are positive showing the profitability of the system. Also, it is obvious that the profits of greedy search heuristic decreases when the threshold increases. The trading system's performance (including both greedy search heuristic and GA) outperformed the three best individual rules in six of the cases.

In Table 4, the best results are achieved for GA and greedy search heuristic methods when the threshold values are 25 and 75, respectively. GA outperformed greedy search heuristic when the threshold value is 50. Greedy search heuristic showed a good and stable performance when the first 14 rules (1st–14th) are selected for combination. GA also showed good performance for thresholds 25 and 50. The trading system's performance (including both greedy search heuristic and GA) outperformed the three best individual rules in 10 of the cases.

The performance of the three best individual trading rules decreased dramatically in Table 4 compared with the results in Table 3 while the trading system's performance increased. It can be concluded that using average profit per trade as the weight calculation metric is more preferable to using net profit.

5.7. Experiments on EUR/USD: 6 months data

The data set covers 1-min time series data between 01.01.2013 and 30.06.2013. The total number of data points in the data set is 182,567. The training data set includes data between 01.01.2013 and 30.04.2013 with a total of 121,196 data points. The test data set includes data between 01.05.2013 and 30.06.2013 with a total of 61,371 data points.

Table 5
Resultant parameters of GA for parameter and rule selection (EUR/USD 6 months data).

	Total running time (in seconds)	Number of generations
Parameter selection	302	80,713
Rule selection	1406	12,722

In Table 5, GA resultant parameters for parameter and rule selection are given.

28 out of 38 trading rules are qualified for the second phase. The list of qualified and unqualified rules are given in Table 6.

The results in Table 6 indicate that 15 out of 22 crossover rules, 3 out of 6 Bollinger Bands and related indicators based rules and all the divergence rules successfully qualified for the second phase.

In Table 7, the best results are achieved for GA and greedy search heuristic methods for the thresholds 25 and 75, respectively. Greedy search heuristic produced a very good and stable performance in all the threshold values where the profitability increases as the threshold increases. Greedy search heuristic results also show that including all the rules in the trading system is a good choice resulting high profits. Three individual rules also showed good performance and it can be concluded that the trading system and the individual rules are in competition.

Table 6
Qualified and unqualified rules (EUR/USD 6 months data).

Rule type	Rule name	
	Qualified rules	Unqualified rules
Crossover	Double moving av., Triple moving av., Moving average env., RSI, Figurelli RSI, Chandelier exit, RVI, Stochastics osc., Ultimate osc., Rate of change, DeMarker, Rel. vigor index, MFI, Chaikin Osc., EMV	Moving average price, TEMA, OBV, MACD, Psychological line, ADL, CMF
Bollinger Bands, %b and Bandwidth Ind. Based	W-type bottom pattern, %b(CMF) crossover, %b-MFI	M-type top pattern, Method III-reversals, The squeeze and expansion
Divergence	Bullish (RSI), Bearish (RSI), Bullish (Figurelli RSI), Bearish (Figurelli RSI), Bullish (Ultimate oscillator), Bearish (Ultimate oscillator), Bullish (DeMarker), Bearish (DeMarker), Bullish (MFI), Bearish (MFI)	–

Table 7
Experiment results using net profit (EUR/USD 6 months data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	8021	10039	9540
	1st–14th	6671	7533	9842
	1st–27th	9000	8027	12,678
	1st–28th	8926	10,353	11,222
Genetic algorithm	13, 19, 11[a]	8154	7466	4611
Best rule	%b-MFI		3234	
2nd Best rule	Rate of chan. cross.		12,457	
3rd Best rule	Bearish div. (MFI)		8945	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

Table 8
Experiment results using average profit per trade (EUR/USD 6 months data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	−3729	4602	1394
	1st–14th	5923	10,314	7666
	1st–27th	6576	11,986	13,363
	1st–28th	6531	8168	7308
Genetic algorithm	17, 16, 17[a]	6186	3982	9365
Best rule	Chaikin osc. cross.		−1499	
2nd Best rule	Bearish div. (RSI)		5824	
3rd Best rule	Bearish div. (MFI)		8945	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

Table 9
Resultant parameters of GA for parameter and rule selection (GBP/USD 6 months data).

	Total running time (in seconds)	Number of generations
Parameter selection	356	69,923
Rule selection	609	7509

The results in Table 8 show that the performance of the individual trading rules and the trading system shows a decrease compared with the net profit results. The negative result in greedy search heuristic method is a clear indication of the situation. Greedy search heuristic method shows a stable performance when the threshold is 25 and outperformed GA when the threshold is 50.

The trading system's performance outperformed the three best individual rules in 3 of the cases.

The results of the experiments using net profit and average profit per trade as the weight calculation metrics are given in Tables 7 and 8, respectively.

After comparing the results in Tables 7 and 8, it can be concluded that using net profit per trade as the weight calculation metric is more preferable to using average profit per trade. Also, GA results are more stable in both net profit and average profit per trade.

5.8. Experiments on GBP/USD: 6 months data

The data set covers 1-min time series data between 01.01.2014 and 30.06.2014. The total number of data points in the data set is 182,197. The training data set includes data between 01.01.2014 and 30.04.2014 with a total of 121,129 data points. The test data set includes data between 01.05.2014 and 30.06.2014 with a total of 61,068 data points.

In Table 9, GA resultant parameters for parameter and rule selection are given.

29 out of 38 trading rules are qualified for the second phase. The list of qualified and unqualified rules are given in Table 10.

The results of the experiments using net profit and average profit per trade as the weight calculation metrics are given in Tables 11 and 12, respectively.

The results in Table 10 indicate that 17 out of 22 crossover rules, 2 out of 6 Bollinger Bands and related indicators based rules and all the divergence rules successfully qualified for the second phase.

In Table 11, the best results are achieved for GA and greedy search heuristic methods when the threshold is 50. Greedy search heuristic method produced stable results when the threshold is 25. GA indicates more stable performance in all the thresholds. In two cases, the trading system outperformed the three best individual rules.

In Table 12, the best results for GA and greedy search heuristic method are obtained when the threshold is 25. The results show a decrease in Table 12 compared with the results in Table 11. 3 negative results in greedy search heuristic justify the situation. GA shows a good and stable performance outperforming the greedy

Table 10
Qualified and unqualified rules (GBP/USD 6 months data).

Rule type	Rule name	
	Qualified rules	Unqualified rules
Crossover	Double moving av., Triple moving av., Moving average env., RSI, Figurelli RSI, Chandelier exit, Psychological line, RVI, Stochastics osc., Ultimate osc., Rate of change, DeMarker, Rel. vigor index, MFI, Chaikin Osc., CMF, EMV	Moving av. price, TEMA, MACD, OBV, ADL
Bollinger Bands, %b and Bandwidth Ind. Based	W-type bottom pat., %b-MFI	M-type top pattern, Method III-reversals, %b(CMF) cross., The squeeze and expansion
Divergence	Bullish (RSI), Bearish (RSI), Bullish (Figurelli RSI), Bearish (Figurelli RSI), Bullish (Ultimate oscillator), Bearish (Ultimate oscillator), Bullish (DeMarker), Bearish (DeMarker), Bullish (MFI), Bearish (MFI)	–

Table 11
Experiment results using net profit (GBP/USD 6 months data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	2971	5167	910
	1st–15th	1328	6232	2743
	1st–28th	3310	316	5716
	1st–29th	3012	1994	1567
Genetic algorithm	13, 19, 11[a]	4856	5181	3715
Best rule	%b-MFI		5627	
2nd Best rule	Bearish div. (MFI)		4328	
3rd Best rule	MFI cross.		3432	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

Table 12
Experiment results using average profit per trade (GBP/USD 6 months data).

Rule selection approach	Selected rules	Threshold		
		25	50	75
Greedy search heuristic	1st–2nd	999	−3037	251
	1st–15th	4959	827	1936
	1st–28th	3896	−272	−672
	1st–29th	5446	343	−723
Genetic algorithm	17, 18, 19[a]	6232	3334	5312
Best rule	Bullish div. (RSI)		−947	
2nd Best rule	Bullish div. (Ultim. osc.)		−571	
3rd Best rule	Bearish div. (MFI)		4328	

[a] Number of selected rules for thresholds 25, 50, and 75, respectively.

search heuristic method in all the thresholds. The trading system's performance outperformed the three best individual rules in 4 of the cases.

The results in Tables 11 and 12 indicate that using net profit as the weight calculation metric and GA as the selection approach is more preferable to achieve stable profits.

6. Conclusion and future work

In this paper, a trading system on FX time series data is developed using trading rules based on technical indicators. A diverse set of technical indicators and various types of trading rules based on these indicators are implemented and used in the system. Among these, crossover rules use mathematical equations whereas divergence and Bollinger Bands and related indicators based rules detect special patterns in the price data.

The proposed trading system combines a potpourri of rules to act as a single rule and carry out buy/sell signals generated by the system. The implementation of the system is realized in three phases. In the first phase, each trading rule is tested for qualification. In this phase, GA module is used to test each rule for qualification in collaboration with trading simulation module. In the second phase, the rules qualified in the first phase are subjected to selection process firstly. The selection is carried out using two approaches: GA and greedy search heuristic. GA module in this phase is used to select the best combination of rules in collaboration with trading simulation and weighting modules. In greedy search heuristic module, the rules are sorted in decreasing order considering their performance followed by selecting the rule having the best performance in the first step and incrementally adding a rule into the selection in each step. Finally, a weighted majority voting method is used in the combination module which combines the selected rules. This module uses weights of the rules (which is provided by weighting module) to combine the decisions of the selected rules. In the final phase, the combined system is simulated on test data and the performance of the system is obtained.

The proposed trading system is tested on EUR/USD and GBP/USD currency pairs in 3 different time frames. In two out of three experiments, using net profit is more preferable to using average profit per trade as the weight calculation metric. Both greedy search heuristic and individual rules resulted negative profits (losses) while GA produced positive gains in all the experiments. This is an evidence of stable performance of GA as the rule selection approach. The trading system outperformed the compared individual rules especially in EUR/USD 1-year data set and showed competitive performance in other data sets. The results considering the threshold values are volatile but 25 as the threshold value produced more stable results in the experiments. One of the most prominent results in the experiments is the superiority of the percentage of profitable trades. When the complete results of the experiments in appendix are evaluated, it can be seen that an average of 60% of the trades are profitable. This is a promising achievement which allows room for increasing the profit of the trading system with a good cash management strategy.

The proposed trading system does not consider transaction costs such as spread, do not apply active order strategies such as take profit and stop loss (discussed in Section 3) and do not have an active cash management strategy. Improving the profitability and stability of the trading system by incorporating all these factors is a future direction of this study. Another future direction is the improvement of GA used in selection of the rules.

As a future work, other soft computing approaches may also be considered, such as simulated annealing, neural networks and support vector machines. Furthermore, rather than using technical indicator rules existing in the literature directly, fuzzified versions of these rules may be constructed and utilized. Since none of the existing technical indicator rules are claimed to be perfect, they are always subject to criticism. Thus, in real life financial market traders do not use any one of them alone to make their real decisions. In our work, when the rules are considered for their qualifications, their strengths are determined in terms of their profits on training data set. These values can be utilized in the fuzzification process. Therefore, we are considering fuzzification also as another possible extension as a future work.

Appendix A. Technical indicators

A.1. Moving average

The calculation of all 4 moving average types [33] are as follows:

$$SMA = SUM(Close, N)/N \quad (A.1)$$

$$EMA = (Close * P) + (EMA(prev) * (1 - P)) \quad (A.2)$$

$$SMMA = (SMMA(prev) * (N - 1) + Close)/N \quad (A.3)$$

$$LWMA = SUM(Close * i, N)/SUM(i, N) \quad (A.4)$$

In Eq. (A.1), *SMA* stands for simple moving average, *Close* is the close price, *N* is the number of periods and *SUM*(*Close*, *N*) is the sum of the close prices in *N* periods. In Eq. (A.2), *EMA* stands for exponential moving average, *Close* is the current period close price, *EMA*(*prev*) is the previous period's exponential moving average value and *P* is the percentage of using the close price value. In Eq. (A.3), *SMMA* is the smoothed moving average of the current period, *SMMA*(*prev*) is the smoothed moving average of the previous period, *Close* is the current period close price and *N* is the smoothing period. In Eq. (A.4), *LWMA* stands for linear weighted moving average, *N* is the smoothing period, *SUM*(*Close* * *i*, *N*) is the weighted sum of the close prices in *N* periods and *SUM*(*i*, *N*) is the total sum of weight coefficients in *N* periods.

A.2. Bollinger Bands

The standard calculation of Bollinger Bands is as follows:

$$MiddleBand = SMA(Close, 20) \quad (A.5)$$

$$UpperBand = SMA(Close, 20) + SD(Close, 20) * 2 \quad (A.6)$$

$$LowerBand = SMA(Close, 20) - SD(Close, 20) * 2 \quad (A.7)$$

where

$$SD = \sqrt{\frac{\sum (x_i - \mu)^2}{N}} \quad (A.8)$$

In Eq. (A.8), *SD* denotes the standard deviation, *x* is data point, μ is the average of data points and *N* is the number of points. In Eqs. (A.5)–(A.7), *SMA*(*Close*, 20) is the simple moving average of close prices in 20 periods, *SD*(*Close*, 20) is the standard deviation of close prices in 20 periods, *MiddleBand*, *UpperBand* and *LowerBand* denote the middle, upper and lower Bollinger Bands values, respectively.

The standard calculation is based on the recommended settings by John Bollinger. The type, period of moving average and standard deviation factor (i.e. SMA, 20, 2 in the above calculation, respectively) are subject to change.

A.3. %b

The calculation of the indicator is as follows:

$$\%b = \frac{Close - LowerBB}{UpperBB - LowerBB}$$

where *Close* is the close price value, *UpperBB* and *LowerBB* are upper and lower Bollinger Bands values, respectively.

A.4. Bandwidth

The calculation of the indicator is as follows:

$$Bandwidth = \frac{UpperBB - LowerBB}{MiddleBB}$$

where *UpperBB*, *MiddleBB* and *LowerBB* are upper, middle and lower Bollinger Bands values, respectively.

A.5. MACD

The calculations of MACD indicator with standard settings are as follows:

$$MACDLine = EMA(Close, 12) - EMA(Close, 26)$$

$$SignalLine = EMA(Close, 9)$$

$$MACDHistogram = MACDLine - SignalLine$$

where *EMA*(*Close*, 9), *EMA*(*Close*, 12) and *EMA*(*Close*, 26) are the exponential moving average of close prices in 9, 12 and 26 periods, respectively.

The period and type of moving averages are recommended by the author; they are subject to change.

A.6. ATR

The standard calculation of ATR indicator is as follows:

$$ATR = \frac{ATR(prev) \times (n - 1) + TR}{n}$$

where *ATR* and *TR* are the current period's ATR and TR value, *ATR*(*prev*) is the previous period's ATR value and *n* is the moving average period.

A.7. RSI

The calculation of RSI indicator with standard settings is as follows:

$$AverageGain = ((AverageGain(prev)) * 13 + Gain)/14 \quad (A.9)$$

$$AverageLoss = ((AverageLoss(prev) * 13 + Loss)/14 \quad (A.10)$$

$$RS = AverageGain/AverageLoss \quad (A.11)$$

$$RSI = 100 - \frac{100}{(1 + RS)} \quad (A.12)$$

In Eqs. (A.9) and (A.11), *AverageGain* and *AverageLoss* are the current average gain and loss in *N* periods (i.e. 14 in the above calculation), *AverageGain* (*prev*) and *AverageLoss*(*prev*) are the previous period's average gain and loss, *Gain* and *Loss* are the positive and negative difference (in absolute value) between the current and previous period's close price, respectively. In Eq. (A.12), *RS* is the relative strength and *RSI* denotes the RSI value, respectively. In the standard calculations, the period is 14 which is subject to change.

A.8. MFI

The calculation of the indicator with the standard settings is as follows:

$$TypicalPrice = (High + Low + Close)/3 \quad (A.13)$$

$$RawMoneyFlow = TypicalPrice * Volume \quad (A.14)$$

$$PositiveMoneyFlow = \begin{cases} RawMoney & Close- \\ Flow, & Close(prev) > 0 \\ 0, & otherwise \end{cases} \quad (A.15)$$

$$NegativeMoneyFlow = \begin{cases} RawMoney & Close- \\ Flow, & Close(prev) < 0 \\ 0, & otherwise \end{cases} \quad (A.16)$$

$$MoneyFlowRatio = \frac{SUM(PositiveMoneyFlow, 14)}{SUM(NegativeMoneyFlow, 14)} \quad (A.17)$$

$$MFI = 100 - \frac{100}{(1 + MoneyFlowRatio)} \tag{A.18}$$

In Eqs. (A.13) and (A.14), *TypicalPrice* and *RawMoneyFlow* denote the typical price and typical price used in the calculations, *Volume* is the volume in the current period, *Close*, *High* and *Low* are the close, highest and lowest price in the current period, respectively. In Eqs. (A.15) and (A.16), *Close* is the current period's close price, *Close*(*prev*) is the previous period's close price, *PositiveMoneyFlow* denotes the raw money flow if the close price increases compared to the previous period and *NegativeMoneyFlow* denotes the raw money flow if the close price decreases compared to the previous period. In Eqs. (A.17) and (A.18), *SUM*(*Posi- tiveMoneyFlow*, 14) and *SUM*(*NegativeMoneyFlow*, 14) denote the sum of positive and negative money flows in the specified period (i.e. 4 above), *MoneyFlowRatio* denotes the ratio of sum of positive money flow divided by sum of negative money flow and MFI denotes the Money Flow Index value. In the standard calculations, the period is 14 which is subject to change.

Appendix B. Trading rules

B.1. Crossover rules

B.1.1. Moving average price crossover

The moving average price crossover is a trading rule based on the moving average indicator [32]. Firstly, the moving average indicator values are calculated throughout the price data. In order to calculate the indicator values, two user defined parameters are needed: moving average period and type. Next, the indicator values are used to generate buy/sell signals using the following strategy:

- When the current price crosses above the current moving average indicator value, a buy signal is generated.
- Conversely, when the current price crosses below the current moving average indicator value, a sell signal is generated [32].

The rule is illustrated in Fig. B.12.

B.1.2. MACD crossover

MACD Crossover is a trading rule based on MACD (moving average convergence divergence) indicator [36]. Firstly, the MACD Line and Signal Line values of MACD indicator are calculated throughout the price data. In order to calculate the indicator values, three user defined parameters are needed: short, long and signal line moving average periods. Next, the indicator values are used to generate buy/sell signals using the following strategy:

- When the MACD Line value crosses above the Signal Line value of MACD indicator, a buy signal is generated.
- Conversely, when the MACD Line value crosses below the Signal Line value of MACD indicator, a sell signal is generated [32,36].

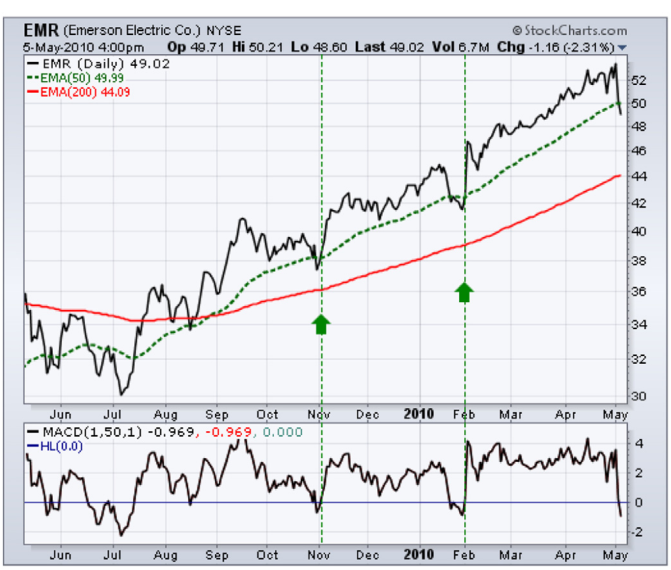

Fig. B.12. Moving average price crossover [32].

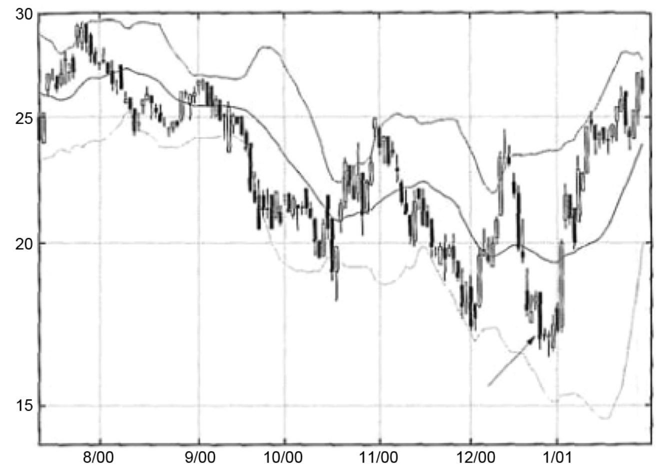

Fig. B.13. W-Type bottom pattern [35].

B.1.3. RSI crossover

RSI Crossover is a trading rule based on RSI (Relative strength index) indicator [37]. Firstly, the RSI indicator values are calculated throughout the price data. In order to calculate the indicator values, user defined period parameter (which is used to calculate average gain and loss) is needed. Next, user defined buy and sell limits are set. Using the indicator values and buy/sell limits, the strategy to generate buy/sell signals is as follows:

- When the RSI value crosses below the buy limit value, a buy signal is generated.
- When the RSI value crosses above the sell limit value, a sell signal is generated [32,37].

B.2. Rules based on Bollinger Bands, %b and bandwidth indicators

B.2.1. W-Type bottom pattern

W-Type Bottom Pattern is a trading rule based on capturing the 5-point W-shaped patterns categorized by Arthur Merrill [35,44]. Firstly, the Bollinger Bands (BB) and %b indicator values are calculated throughout the price data. In order to calculate those values, three user defined parameters are needed: moving average period, type and standard deviation factor. Next, user defined %b threshold is set. Using the BB indicator values, there are 4 successive steps to confirm a W-Type Bottom pattern:

- First, the price decreases below or touches the lower BB band, thereby setting the first low.
- Second, when the price increases above or touches the middle BB band, the highest price is set as the resistance point.
- Third, the price decreases below the middle BB band but hold above the lower BB band, thereby setting the second low.
- Finally, the price moves above the resistance point confirming the W-Type bottom pattern [32,35].

The pattern is illustrated in Fig. B.13. In the light of the above explanations, the buy/sell strategy of the rule is as follows:

- When a W-Type Bottom pattern is confirmed, a buy signal is generated.
- When the price increases above the user defined %b indicator threshold, a sell signal is generated.

"First buy, then sell" is preferred as the trading strategy.

B.2.2. %b-MFI

%b-MFI is a trading rule based on %b and MFI indicators. Firstly, the %b and MFI indicator values are calculated throughout the price data. In order to calculate those values, four user defined parameters are needed: moving average period (to calculate both of the indicators), moving average type and standard deviation factor (to calculate %b indicator) and volume type (to calculate MFI indicator). Next, user defined %b and MFI buy/sell limits are set. Using the indicators values and buy/sell limits, the strategy to generate buy/sell signals is as follows:

- When %b indicator value is less than user defined %b buy limit and MFI indicator value is less than user defined MFI buy limit, a buy signal is generated.
- When %b indicator value is greater than user defined %b sell limit and MFI indicator value is greater than user defined MFI sell limit, a sell signal is generated.

"always in the market" is preferred as the trading strategy.

B.2.3. The squeeze and expansion

The Squeeze and Expansion is a trading rule based on the squeeze and expansion of the upper and lower bands of BB indicator [35]. The Squeeze is based on the fact that Bollinger Bands squeeze (or narrow) when the volatility of the price decreases. This may be a signal of an uptrend/downtrend. On the other hand, the Expansion occurs when the Bollinger Bands expands where the high volatility comes into play. This may be and end of the current uptrend/downtrend which should be confirmed with some other indicator(s) [32,35].

The rule depends on Bandwidth and CMF indicators. Firstly, the Bandwidth and CMF indicator values are calculated throughout the price data. In order to calculate those values, four user defined parameters are needed: moving average period (to calculate both of the indicators), moving average type and standard deviation factor (to calculate Bandwidth indicator) and volume type (to calculate CMF indicator). Next, user defined squeeze and expansion thresholds are set. Using the indicators values and squeeze and expansion thresholds, the strategy to generate buy/sell signals is as follows:

- When the bandwidth indicator value is less than the squeeze threshold:
  - If the average of preceding and succeeding periods' CMF values are greater than 0, a buy signal is generated.
  - If the average of preceding and succeeding periods' CMF values are less than 0, a sell signal is generated.
- If a buy or sell signal is generated as a result of squeeze, the bandwidth indicator is checked. When the bandwidth indicator value is greater than the expansion threshold:
  - If the position is long (buy signal generated in squeeze), a sell signal is generated and the trade is closed.
  - If the position is short (sell signal generated in squeeze), a buy signal is generated and the trade is closed.

B.3. Divergence rules

B.3.1. Rules based on bullish divergences

A bullish divergence is the type of divergence which signals an uptrend in prices [45]. It occurs in case of a reversal in the current downtrend (regular bullish divergence) or as a confirmation of the beginning of an uptrend (hidden bullish divergence).

Several indicators can be used to detect a bullish divergence. However, in this thesis, five indicators are used: RSI, Figurelli RSI, MFI, DeMarker and Ultimate Oscillator. Firstly, the selected indicator (one of the RSI, Figurelli RSI, MFI, DeMarker or Ultimate Oscillator) values are calculated throughout the price data. The parameters of the selected indicator are defined by the user. Next, user defined sell limit is set. Using the price low data, indicator values and sell limit, the strategy to generate buy/sell signals is given in Algorithm 3.

B.3.2. Rules based on bearish divergences

A bearish divergence is the type of divergence which signals a downtrend in prices [45]. It occurs in case of a reversal in the current uptrend (regular bearish divergence) or as a confirmation of the beginning of a downtrend (hidden bearish divergence).

Several indicators can be used to detect a bearish divergence. Similar to bullish divergence, five indicators are used in this thesis: RSI, Figurelli RSI, MFI, DeMarker and Ultimate Oscillator. Firstly, the selected indicator (one of the RSI, Figurelli RSI, MFI, DeMarker or Ultimate Oscillator) values are calculated throughout the price data. The parameters of the selected indicator are defined by the user. Next, user defined buy limit is set. Using the price high data, indicator values and buy limit, the strategy to generate buy/sell signals is given in Algorithm 4.

Algorithm 3. Pseudo code of bullish divergence detection.

```
Require: ind:indicator data, low:price low data, sl:sell limit, idivl:indicator
limit to start checking divergence, i:current index
  sellFlag := False    // Firstly, check for divergence which generates a buy signal
  i := 0
  while i ≤ |ind| do
    if ind[i] < idivl ∧ sellFlag = False then
      Detect the first indicator low and mark as i1.
      Mark the low value as low1 which has the same index with i1.
      Detect the second indicator low and mark as i2.
      Mark the low value as low2 which has the same index with i2.
      if i1 < i2 ∧ low1 > low2 then
        A buy signal is generated.          // Regular Bullish Divergence
        sellFlag := True
      els if i1 > i2 ∧ low1 < low2 then
        A buy signal is generated.          // Hidden Bullish Divergence
        sellFlag := True
      end if
    end if
    if sellFlag = True then
      if ind[i] > sl then
        A sell signal is generated.   // Indicator value greater than the sell limit
        sellFlag = False
      end if
    end if
    i := i + 1
  end while
```

Algorithm 4. Pseudo code of bearish divergence detection.

```
Require: ind:indicator data, high:price high data, bl:buy limit,
idivl:indicator limit to start checking divergence, i:current index
  buyFlag := False    // Firstly, check for divergence which generates a sell signal
  i := 0
while i ≤ |ind| do
    if ind[i] > idivl ∧ buyFlag = False then
      Detect the first indicator high and mark as h1.
      Mark the high value as high1 which has the same index with h1.
      Detect the second indicator high and mark as h2.
```

Mark the *high* value as *high*2 which has the same index with *h*2.
if $h1 > h2 \wedge high1 < high2$ **then**
A sell signal is generated. // Regular Bearish Divergence
$buyFlag := True$
els if $h1 < h2 \wedge high1 > high2$ **then**
A sell signal is generated. // Hidden Bearish Divergence
$buyFlag := True$
end if
end if
if $buyFlag = True$ **then**
if $ind[i] < bl$ **then**
A buy signal is generated. // Indicator value less than the buy limit
$buyFlag = False$
end if
end if
$i := i + 1$
end while

References

[1] B.F.I.S.-M.E. Department, Foreign Exchange Turnover in April 2013: Preliminary Global Results, Tech. Rep., Bank For International Settlements, 2013 http://www.bis.org/publ/rpfx13fx.pdf.
[2] M.D. Archer, Getting Started in Currency Trading, Wiley, 2010.
[3] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.
[4] A. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, 2008.
[5] R.J. Sweeney, Beating the foreign exchange market, J. Financ. 41 (1) (1986) 163–182.
[6] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, J. Financ. Econ. 51 (1999) 245–271.
[7] S. Papadamoua, G. Stephanides, Improving technical trading systems by using a new MATLAB-based genetic algorithm procedure, in: Proceedings of the International Conference on Computational Methods in Sciences and Engineering 2004, Mathematical and Computer Modelling, 2004, pp. 189–197.
[8] A. Esfahanipour, S. Mousavi, A genetic programming model to generate risk-adjusted technical trading rules in stock markets, Expert Syst. Appl. 38 (7) (2011) 8438–8445.
[9] L. Wei, C. Cheng, A GA-weighted ANFIS model based on multiple stock market volatility causality for TAIEX forecasting, Appl. Soft Comput. 13 (2) (2013) 911–920.
[10] C. Schoreels, B. Logan, Agent based genetic algorithm employing financial technical analysis for making trading decisions using historical equity market data, in: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2004 (IAT 2004), IEEE, 2004, pp. 421–424.
[11] C. Neely, P. Weller, R. Dittmar, Is technical analysis in the foreign exchange market profitable? A genetic programming approach, J. Financ. Quant. Anal. 32 (4) (1997) 405–426.
[12] J. Korczak, P. Lipinski, Evolutionary building of stock trading experts in a real-time system, in: Proceedings, 2004 Congress on Evolutionary Computation, IEEE, 2004, pp. 940–947.
[13] N. Jin, E. Tsang, J. Li, A constraint-guided method with evolutionary algorithms for economic problems, Appl. Soft Comput. 9 (3) (2009) 924–935.
[14] A. Hryshko, T. Downs, System for foreign exchange trading using genetic algorithms and reinforcement learning, Int. J. Syst. Sci. 35 (13–14) (2004) 763–774.
[15] M. Dempster, C. Jones, A real-time adaptive trading system using genetic programming, Quant. Financ. 1 (4) (2001) 397–413.
[16] S. Deng, K. Yoshiyama, T. Mitsubuchi, A. Sakurai, Hybrid method of multiple kernel learning and genetic algorithm for forecasting short-term foreign exchange rates, Comput. Econ. 45 (1) (2015) 49–89.
[17] A. Hirabayashi, C. Aranha, H. Iba, Optimization of the trading rule in foreign exchange using genetic algorithm, in: GECCO '09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, ACM, 2009, pp. 1529–1536.
[18] H. Ni, H. Yin, Exchange rate prediction using hybrid neural networks and trading indicators, Neurocomputing 72 (13-15) (2009) 2815–2823.
[19] H. Thinyane, J. Millin, An investigation into the use of intelligent systems for currency trading, Comput. Econ. 37 (4) (2011) 363–374.
[20] L. Mendes, P. Godinho, J. Dias, A Forex trading system based on a genetic algorithm, J. Heuristics 18 (4) (2012) 327–656.
[21] R.F.B. de Brito, A.L.I. Oliveira, Comparative study of forex trading systems built with SVR+GHSOM and genetic algorithms optimization of technical indicators, in: Proceedings of the 2012 24th IEEE International Conference on Tools with Artificial Intelligence, IEEE, 2012, pp. 351–358.
[22] S. Deng, A. Sakurai, Integrated model of multiple kernel learning and differential evolution for EUR/USD trading, ScientificWorldJournal (2014).
[23] S. Deng, Y. Sun, A. Sakurai, Robustness test of genetic algorithm on generating rules for currency trading, in: Proceedings of the International Neural Network Society Winter Conference (INNS-WC2012), Procedia CS, 2012, pp. 86–98.
[24] A. Kayal, A neural networks filtering mechanism for foreign exchange trading signals, in: Proceedings of the 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), IEEE, 2010, pp. 159–167.
[25] S. Deng, A. Sakurai, Foreign exchange trading rules using a single technical indicator from multiple timeframes, in: Proceedings of the 2013 27th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2013, pp. 207–212.
[26] B. Dolan, Currency Trading For Dummies, Wiley, 2011.
[27] Learn Forex trading at the school of pipsology, http://www.babypips.com/school/ (accessed January 2015).
[28] E. Ponsi, Forex Patterns & Probabilities, Wiley, 2007.
[29] J.J. Murphy, Technical Analysis of The Financial Markets, New York Institute of Finance, 1999.
[30] A. Kritzer, Forex for Beginners: A Comprehensive Guide to Profiting from the Global Currency Markets, Apress, 2012.
[31] R.W. Colby, The Encyclopedia of Technical Market Indicators, McGraw-Hill, 2003.
[32] S. Inc., Technical indicators and overlays, http://stockcharts.com/school (accessed January 2015).
[33] M.S. Corp., http://www.metatrader5.com (accessed January 2015).
[34] F. Larsen, Automatic Stock Market Trading Based on Technical Analysis, Master's thesis, Norwegian University of Science and Technology, 2007.
[35] J. Bollinger, Bollinger on Bollinger Bands, McGraw-Hill, 2002.
[36] G. Appel, Technical Analysis: Power Tools for Active Investors, Financial Times Prentice Hall, 2005.
[37] J.W. Wilder, New Concepts in Technical Trading Systems, Trend Research, 1978.
[38] G. Quong, A. Soudack, The ultimate oscillator, Tech. Anal. Stocks Commod. 7 (3) (1989) 76–77.
[39] L. Sether, SFO Personal Investor Series: Forex Trading, W & A Publishing, 2009.
[40] M. Link, High Probability Trading, McGraw-Hill, 2003.
[41] S. Wagner, Heuristic Optimization Software Systems-Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment, Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria, 2009, Ph.D. thesis.
[42] T.D. Gwiazda, Genetic Algorithms Reference Volume I, 2006.
[43] http://dev.mysql.com/doc/refman/5.6/en/index.html (accessed January 2015).
[44] A.A. Merrill, M & W Patterns, Analysis Press, 1983.
[45] P.J. Kaufman, Trading Systems and Methods, Wiley, 2013.