



Trend following algorithms in automated derivatives market trading

Simon Fong^{*}, Yain-Whar Si, Jackie Tai

Department of Computer and Information Science, University of Macau, Macau

ARTICLE INFO

Keywords:

Trend following
Automated trading system
Futures contracts
Mechanical trading

ABSTRACT

Trend following (TF) is trading philosophy by which buying/selling decisions are made solely according to the observed market trend. For many years, many manifestations of TF such as a software program called Turtle Trader, for example, emerged in the industry. Surprisingly little has been studied in academic research about its algorithms and applications. Unlike financial forecasting, TF does not predict any market movement; instead it identifies a trend at early time of the day, and trades automatically afterwards by a pre-defined strategy regardless of the moving market directions during run time. Trend following trading has been popular among speculators. However it remains as a trading method where human judgment is applied in setting the rules (aka the strategy) manually. Subsequently the TF strategy is executed in pure objective operational manner. Finding the correct strategy at the beginning is crucial in TF. This usually involves human intervention in first identifying a trend, and configuring when to place an order and close it out, when certain conditions are met. In this paper, we evaluated and compared a collection of TF algorithms that can be programmed in a computer system for automated trading. In particular, a new version of TF called trend recalling model is presented. It works by partially matching the current market trend with one of the proven successful patterns from the past. Our experiments based on real stock market data show that this method has an edge over the other trend following methods in profitability. The results show that TF however is still limited by market fluctuation (volatility), and the ability to identify trend signal.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Trend following (TF) (Fong & Tai, 2009) is a reactive trading method in response to the real-time market situation; it does neither price forecasting nor predicting any market movement. Once a trend is identified, it activates the trading rules and adheres rigidly to the rules until the next prominent trend is identified. Trend following does not guarantee profit every time, but nonetheless in a long term period it may probably profit by obtaining more gains than losses. The nature of TF makes it as an ideal ingredient in implementing a decision-making component in automated trading software where human intervention is not required in automatic buying and selling. Automated trading here is referred to an 'auto-pilot' mode by which the software system decides when to buy or sell a stock (in a form of option, future contract, warrant, etc.) from the derivative market. Though the philosophy of trend following is simple – identify a current market trend and trade strictly according to the pre-defined rules (sometimes known as strategies), how the rules or strategies should be formulated has become an important research problem that deserves attention from researchers. Rules such as when to buy and sell or when to

open and close a position, as signaled from the market trend have a direct impact into the profitability of automated market trading. TF algorithms in this context are termed as automated trading methods that are guided by the current market trend (signals from the trend) and specified by the rules (reactions to the trend). For an example of a most basic TF algorithm, buying and selling are cued by the conditions when the market trend which is represented by its moving average rises over an up-threshold, and falls below a down-threshold respectively. The values of the thresholds are pre-defined as a part of the trading rules.

To the best of the authors' knowledge, surprisingly, TF algorithms have not been investigated in the computer science community. In contrast a lot of research papers on stock market forecast and prediction by soft computing can be found in literature. Fig. 1 classifies clearly that trading algorithms can be powered by two different domains of techniques, namely Predictive and Reactive. Predictive types of forecasting and trading in a stock market have a long history in academic research, which generally covers non-linear prediction by artificial neural works, decision trees and other regression models. These tools usually aim at predicting the future market movement ahead by analyzing over the historical data. TF algorithms however, belong to the latter category, which conduct trading decision solely by reacting to the current market trend. There are several variants of TF algorithms, depending on how the rules and the predefined thresholds are

^{*} Corresponding author. Tel.: +853 83974460; fax: +853 2883 8314.
E-mail address: ccfong@umac.mo (S. Fong).

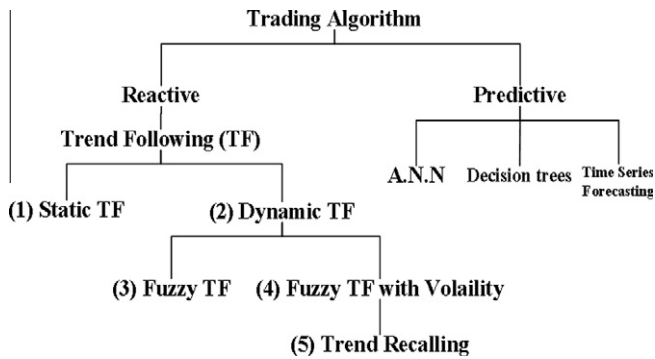


Fig. 1. Classification of trading algorithms.

specified. Without the need of training up a decision model, and subsequently updating it, like those under the predictive type, TF is readily implemented as a rule-based system in automated trading software.

The objective of this paper is to present a performance comparison of the five TF algorithms which have been recently proposed by the authors. The contribution of this paper is twofold: Theoretically insights are gained in terms of performance strength, weakness and characteristics of each TF algorithm under test, so that prospect of future research in improving from the existing TF algorithms is warranted. Secondly the TF algorithms presented in this paper serve as useful designs of the decision-making core for implementing an automated stock market trading system. The structure of this paper is as follow: Section 2 describes four possible implementations of TF algorithms, highlighting especially their shortcomings and possible improvement. Section 3 presents in details the latest addition to the family of TF algorithms called trend recalling algorithm. A simulation prototype of automated trading system is developed, for the purpose of evaluating their performances comparatively. The experiments carried out by the simulator are reported in Section 4. Section 5 concludes.

2. Trend following algorithms

In this paper, we discuss on the design of an automated trading system, which incorporates these trend following algorithms, which is a core element of the system, and the application of this type of system in financial market. We also investigate how the market fluctuation can affect the overall performance, and bring new perspective to handling the financial cycles.

One of our goals is to build up this automated trading system, which primarily operates on financial derivative market (Mayo, 2003), such as the Hang Sang Index Futures Contract that trade on HKFE (Hong Kong Future Exchange) or the Dow Jones Industrial Average Index Futures Contract that trade on CBOT (Chicago Board of Trade). In this thesis “Hang Sang Index Futures” is selected as the primary simulation market, although the system can be applied on any market theoretically.

By using only historical market data, the system is able to react according to real-time market state, and make trade decision on its own. To reduce the overnight risk and cost-of-carry, trade will be performed on a daily basic only, which means no contracts will be carried overnight. The P&L (profit and loss) on each trade will be recorded and accumulated as the total of ROI (return on investment), which is a common indication for the performance of an investment in financial world.

2.1. Static TF algorithm

The basic practice of trend following is to find the trend, identify trade signals and trade along with them. Based on a phenomenon

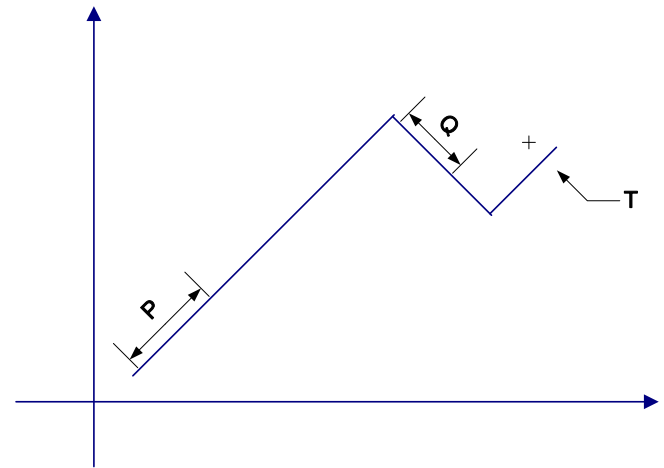


Fig. 2. Illustration of P and Q applied on trend T.

pointed out by Murphy (1999), which says “A trend in motion is more likely to continue than to reverse”, TF algorithm can possibly reap benefit by recognizing the direction of the trend at the start and bet on that direction for the remaining length of the trend, as long as the trend does not reverse. For confirming whether a trend is now in effect of development, it is necessary to measure how much it has progressed since the last reverse, and then ensure the current progress is greater than the minimum threshold that it has risen or fallen. This is to test by comparing the trend momentum to a specific level. If the amount of increase or decrease of a trend has already exceeded a certain level, a significant trend is assumed to emerge and trade signals can be safely generated accordingly. These marks or minimum levels will have to be chosen carefully in order to avoid false signals as a result of temporary fluctuation or short-term velocity of the trend. If they are too long, few positions will be opened or closed because it is not often that a large scale change on a trend would occur. If the marks are chosen too short, the trading is prone to be buying very frequently and selling too early, leaving little or no profit in between the trades.

In this Static TF algorithm, two constants are used to function as the two comparison marks, only over which a substantial change in the trend would trigger the trading system to open or close a position accordingly. These constants are defined as P and Q, where P is the amount of up-trend required for opening a position, and Q is the amount of opposite trend required to close this position.

Fig. 2 illustrates the basic concept of TF. T denotes as the market price trend. For example on the diagram, it will open a long position when the trend T advances over P, and will close out this position when the trend T declines for more than Q.

In reality market price does not move in a straight line. It is therefore impractical to apply the P and Q rules directly on the trend T, because the frequent fluctuation will alarm off too many signals of trading actions. An Exponential Moving Average (EMA) algorithm is used to smooth out this fluctuation, which equation is shown as follow:

$$EMA_{(t)} = \left(price_{(t)} - EMA_{(t-1)} \times \frac{2}{n+1} \right) + EMA_{(t-1)} \quad (1)$$

Fig. 3 demonstrates the effect of the smoothing.

This trading algorithm is represented by the following pseudo-code (Algorithm 1), in which P and Q are static constants. Their values are obtained by studying the historical data and deriving the optimal combination of P and Q. Different combinations of P and Q are tried by brute-force, testing out for the best values that generate the maximum profit in the simulation. $EMA(T)$ is the

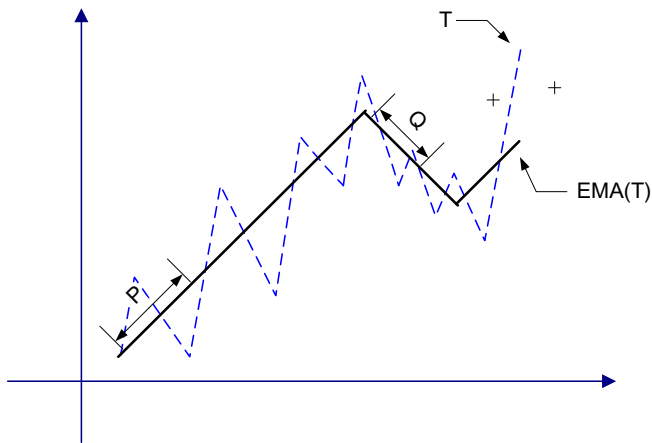


Fig. 3. P and Q applied on smoothed trend $EMA(T)$.

Exponential Moving Average of the real time market price trend, and $EMA(-T)$ is the reversion (opposite direction) of the trend counting from the highest(or lowest) point of this trend.

Algorithm 1. Pseudo-codes of the Static TF algorithm.

```

Repeat until end of market
  Compute  $EMA(T)$ 
  If no position opened
    If  $EMA(T) \geq P$ 
      If trend is going up
        Open a long position
      Else if trend is going down
        Open a short position
    Else if any position is opened
      If  $EMA(-T) \geq Q$ 
        Close position
  If end of market
    Close all opened position
  
```

The basic premise is that the most profit is gained when a trade is synchronized to an enduring trend. The rules of this algorithm are statically formulated as the trade parameters remain unchanged once they have been defined throughout the whole trading process. The experimental results from simulation with Static TF algorithm are detailed in Section 4.

2.2. Dynamic TF algorithm

In this dynamic TF algorithm, P and Q are variables instead of static constants, and their values change adaptively to the current market trends. Based upon this initial concept of trading algorithm, dynamic TF algorithm is introduced with incorporation of technical analysis concept. Technical analysis makes trade decision through technical indicators such as RSI (relative strength index), STC (stochastic oscillator), and EMA and these indicators are changing dynamically according to the market situation. By adopting one or more of these indicators and by studying how they react to the market, rules can be formed that are able to inherit this dynamic nature. By following these rules during trade session, we update the trade parameters P and Q with the latest dynamic values attribute.

There are hundreds of indicators in use today, but not all are tested to be reliable. Experiments have been conducted to try out many popular ones and RSI is found to be the best for TF. RSI compares the magnitude of underlying recent gains of an asset to the magnitude of its recent losses, and normalized to a number that ranges from 0 to 100

$$RSI_{(t)} = 100 - \frac{100}{1 + RS} \quad (2)$$

$$RS = \frac{AU(t)}{AD(t)} \quad (3)$$

$$AU(t) = \frac{Up(t) + Up(t-1) + \dots + Up(t-n+1)}{n} \quad (4)$$

$$AD(t) = \frac{Down(t) + Down(t-1) + \dots + Down(t-n+1)}{n} \quad (5)$$

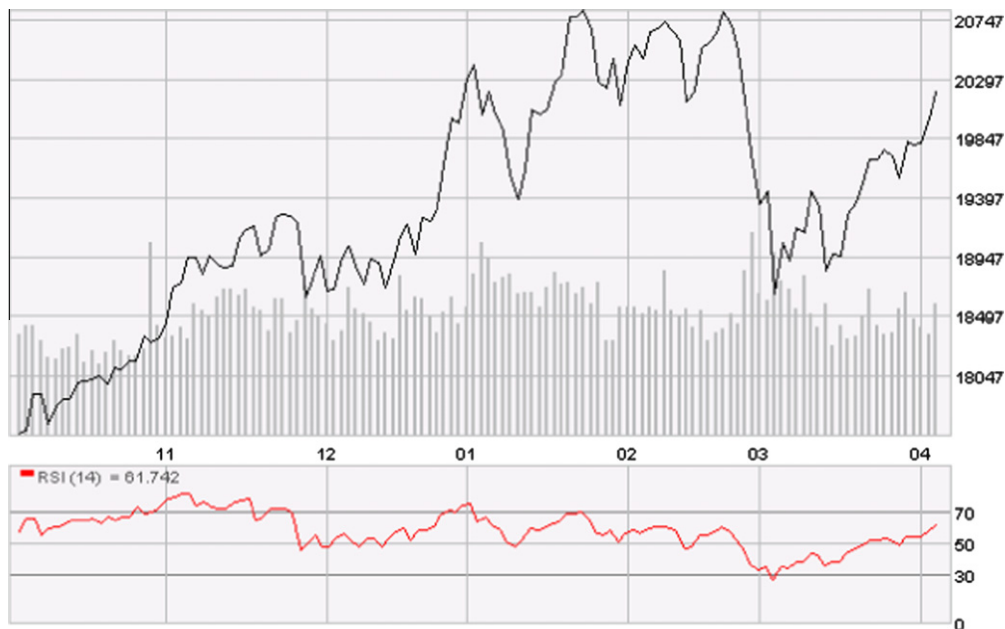


Fig. 4. Example of relative strength index.

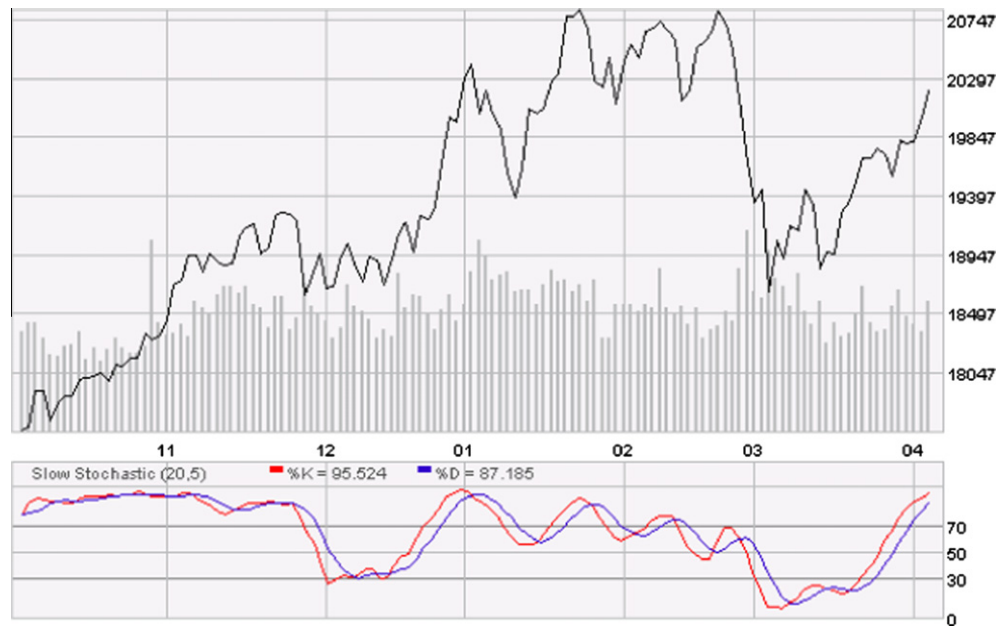


Fig. 5. Example of stochastic oscillator index.

where AU is average price upward in n periods, AD is average price downward in n periods, t is the time, n is the number of RSI periods usually 14.

Fig. 4 shows an example of this indicator, which is below the primary price trend.

Stochastic Oscillator (STC) is a momentum indicator that shows the location of the current close relative to the high/low range over a set number of periods.

Formula:

$$\%K(t) = 100 \times \frac{\text{Close}(t) - LL(n)}{HH(n) - LL(n)} \quad (6)$$

$$\%D(t) = \text{EMA}(\%K(t))(m) \quad (7)$$

where HH is highest high in n periods, LL is lowest low in n periods, n is number of STC periods, m is number of periods of EMA that applied on $\%K$.

Fig. 5 shows an example of this indicator, which is below the primary price trend.

By studying how this indicator reacts to the market, we can derive rules and criteria that describe the current market situation systematically. And this information could eventually lead to trade execution. The logic of rules is given below:

As for long position, the following criteria should be satisfied.

1. Price is advancing.
2. $RSI(t)$ is greater than $RSI(EMA(t))$.
3. $RSI(EMA(t))$ is less than 40 or greater than 60.

And for short position, the following criteria should be satisfied.

1. Price is declining.
2. $RSI(t)$ is less than $RSI(EMA(t))$.
3. $RSI(EMA(t))$ is less than 40 or greater than 60.

By following those rules and criteria, we can see the values of P and Q are now changing along the trend with reference to RSI. Since RSI is changing according to market condition dynamically, rules that are derived from it will also adopt to this dynamic situation. By applying these rules during trading session, we therefore

obtain the values of P and Q that give good results. When the criterion that is described above is satisfied, it triggers the amount of up (or down) trend and causes a position to be open (or close). Now we summarize this trading algorithm as the following pseudo-code (Algorithm 2)

Algorithm 2. Pseudo-codes of the dynamic TF algorithm.

```

Repeat until end of market
  Compute  $RSI(t)$  and  $RSI(EMA(t))$ 
  If price is advancing:
    If  $RSI(t) > EMA(t)$  and  $40 < EMA(t) > 60$ 
      If no position has been opened
        Open a long position
      Else if short position has been opened
        Close out short position
    Else if price is declining:
      If  $RSI(t) < EMA(t)$  and  $40 < EMA(t) > 60$ 
        If no position has been opened
          Open a short position
        Else if long position has been opened
          Close out long position
  If end of market
    Close all opened positions

```

The experimental results from simulation with dynamic TF algorithm are detailed in Section 4.

2.3. Fuzzy TF algorithm

The static and dynamic trend following algorithms described in previous section are designed to make trading decisions based on criteria which are formulated as crisp membership of classical binary logic. In this section, we extend the trending following algorithms based on Fuzzy logic (Zadeh, 1973). jFuzzyLogic (jFuzzyLogic, 2012) was used to develop a fuzzy inference system. Based on our experience with previous trend following algorithms, we define three membership functions for input and output variables.

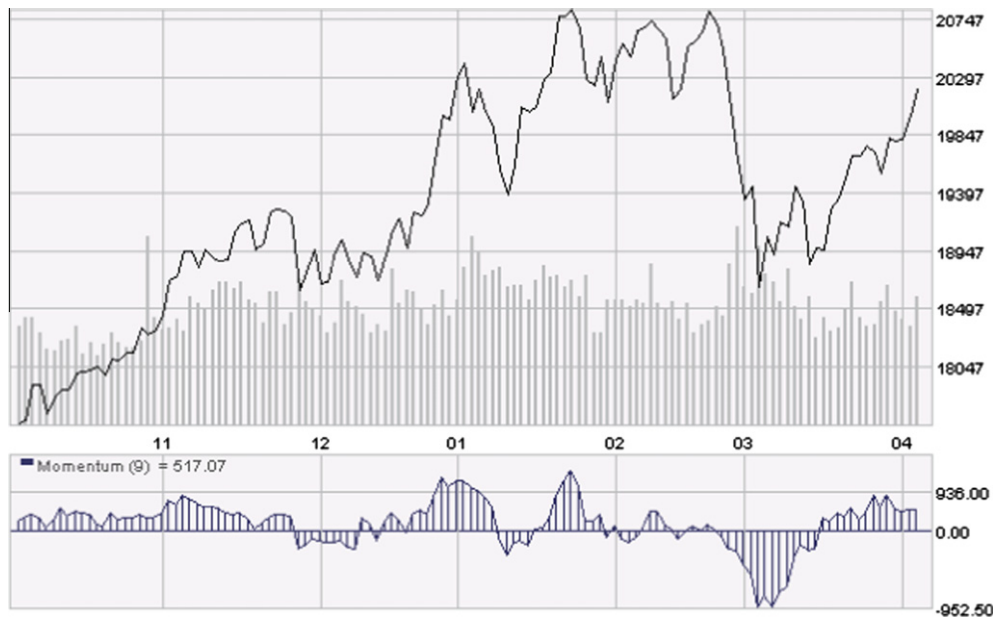


Fig. 6. Example of momentum index.

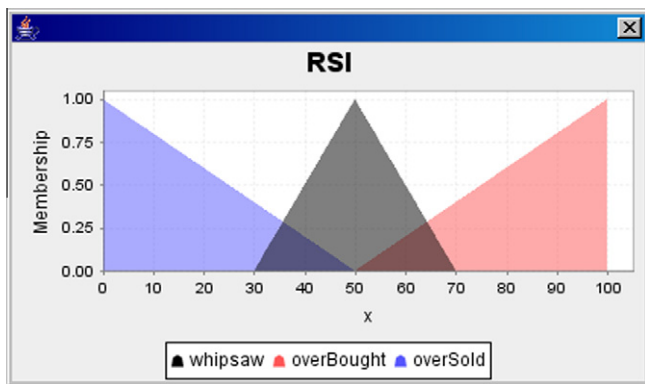


Fig. 7. Input membership function for relative strength index (RSI).

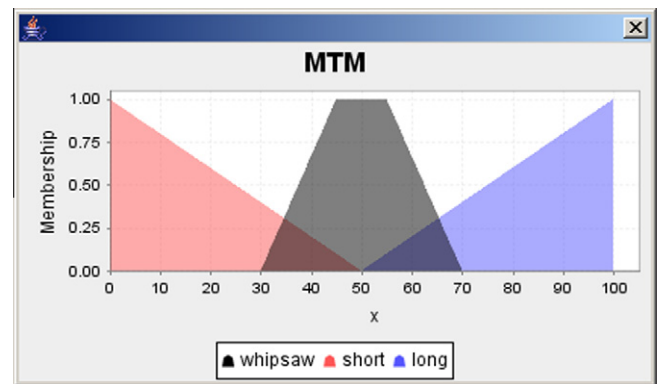


Fig. 8. Input membership function for momentum indicator (MTM).

These functions are depicted in Figs. 7–9. The fuzzy inference engine accepts relative strength index (RSI) and momentum indicator (MTM) as input and produces recommendations on whether or not to take a position (POS) as output.

Momentum (MTM) is an oscillator type indicator used to detect overbought and oversold conditions and to perform as a gauge indicating the strength of the current trend. Momentum calculations are either positive or negative and fluctuate around a zero-line

$$MTM(t) = C(t) - C(t - n) \quad (8)$$

where $C(t)$ is the closing price, n is the number of MTM periods.

Fig. 6 shows an example of this indicator, which is below the primary price trend.

From these membership functions, 27 inference rules can be implemented. Among these rules, five of them are selected for the experiments.

1. IF RSI IS whipsaw OR MTM IS whipsaw THEN POS IS doNothing.
2. IF RSI IS overSold AND MTM IS long THEN POS IS goLong.
3. IF RSI IS overBought AND MTM IS short THEN POS IS goShort.
4. IF RSI IS overSold AND MTM IS short THEN POS IS goShort.

5. IF RSI IS overBought AND MTM IS long THEN POS IS goLong.

Whipsaw is a condition where a security's price heads in one direction, but then is followed quickly by a movement in the opposite direction (Investopedia, 2012). Whipsaw pattern for relative strength index (RSI) can be considered as a neural signal in terms of the velocity and magnitude of directional price movements. The security is considered to be in overbought territory when RSI is above 70 and considered to be over sold when RSI is below 30. Momentum shows the difference between today's closing price and the closing price of n days ago (Wikipedia, Momentum (technical analysis), 2012). Momentum can be calculated as follows:

$$momentum = close_{today} - close_{n \text{ days ago}} \quad (9)$$

The first fuzzy inference rule directs the trend following system to hold the current position if relative strength index and momentum are neutral. The second fuzzy inference rule directs the system to buy when relative strength index is oversold and momentum is high. Remaining inference rules can be interpreted in a similar way. The experimental results from simulation with Fuzzy TF algorithm are detailed in Section 4. In the next section,

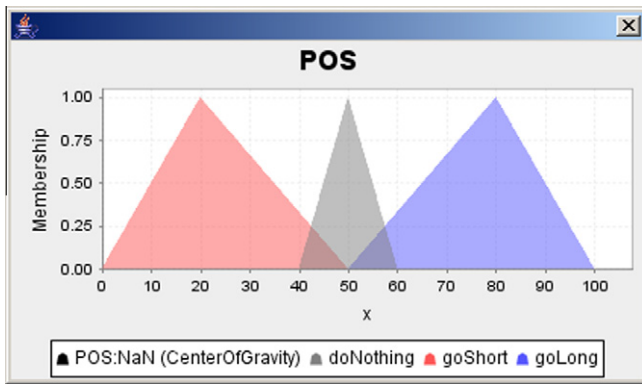


Fig. 9. Output membership function on taking a position (POS).

we further extend the algorithm to take into account the volatility in price.

2.4. Fuzzy TF with Volatility algorithm

Fluctuation or volatility is one of the important aspects considered by professional traders before making any trading decision. Volatility can be described as the intensity of the ups and downs in price. Volatility is often driven by factors such as emotion, greed, fear, and political situation. Optimism and greed can also drive the prices up when the majority of investors are making profit in up-trend market. During downtrend, fear and panic among investors can drive the prices into lower levels. There are two type of volatility, Historical and Implied (Fontanills & Gentile, 2002). Historical volatility is a measure of price fluctuation over a time period in the past. Historical volatility is usually derived from standard deviation or from the average of price difference. Implied Volatility is often used on option trading. Implied Volatility is a measure of market expectations to an asset's future volatility, which is implied

by the underlying asset's option price. It is based on an option pricing model such as the famous Black-Scholes (Hull, 2000). Implied Volatility is often used on option trading.

To find out how market fluctuation can affect our algorithm, we attempt to simulate the market with a controllable fluctuation. We then apply our trading algorithms on it, and observe how it performs under certain degree of fluctuation. Although it is almost impossible to model a real-world market fluctuation exactly in lab, we therefore turn to simulate the fluctuation scenario by numeric patterns. To do that we need to generate a series of simulated market data; these data must contain some degree of randomness, so it will behave similar to the market movement. Meanwhile we still need to control it, so it will move and fluctuate as expected. The following formula is designed to accomplish these:

$$\text{MarketData}(t) = (\text{COS}(e) \times C \times R(t)) + B \quad (10)$$

where COS is the geometry cosine which creates a basic oscillating wave structure, e is the angle that controls the fluctuation frequency, C is a constant that controls the fluctuation depth, R is a random number that creates the Whipsaw sharp, and B is a fixed base price that the generated prices will fluctuate along with.

By controlling the parameter in the formula (10), we can generate the market data that gradually increases the level of fluctuation, with a degree starting from 0 that indicates zero fluctuation, up to 100 that indicates maximum fluctuation. Fig. 10 shows some of the simulated market data sets in an intra-day scale chart.

As the fluctuating data are generated, they are fed into the trading simulation engine. The result is summarized in Fig. 11. It shows when the fluctuation is over 50%, it starts to make loss. On each time when the simulation data set is regenerated, though there might be some differences on the P&L numbers, but in general the observation of making loss gradually stay true, no matter how many times the simulated data set regenerate.

Table 1 summarizes all fluctuation test result data.

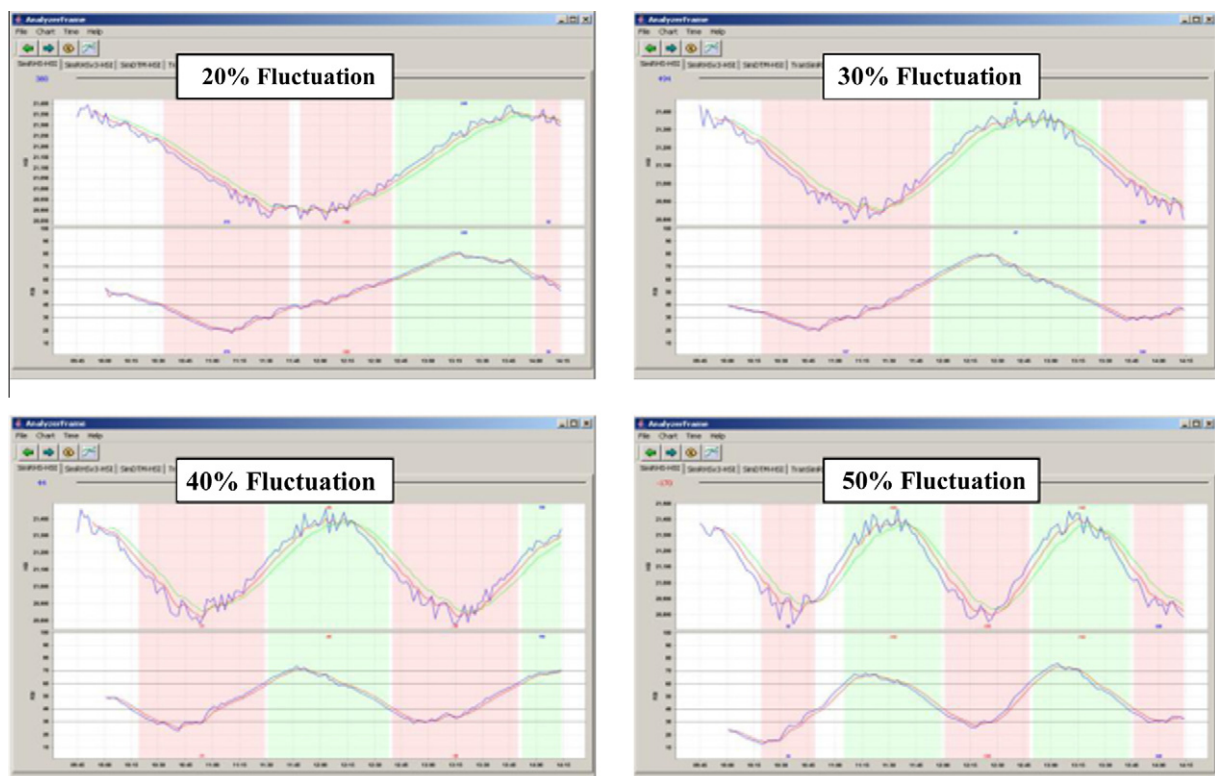


Fig. 10. Example of simulated market trend with fluctuation control.

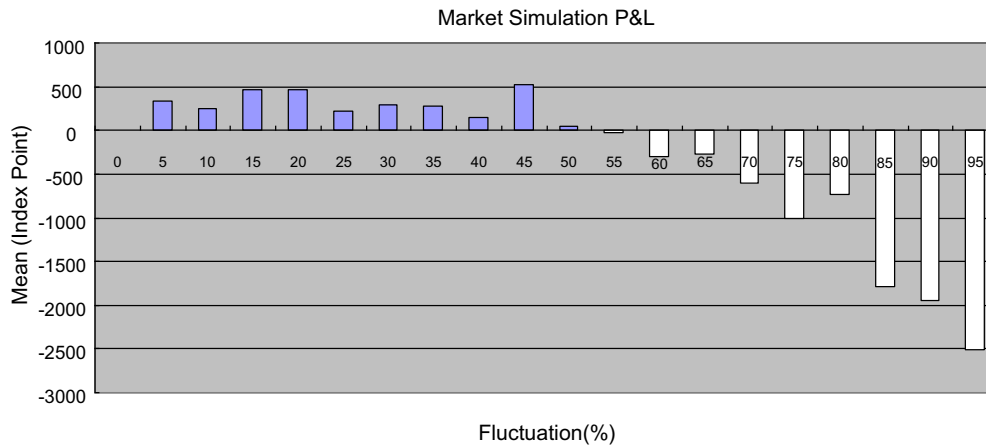


Fig. 11. Fluctuation test result.

Table 1
Fluctuation test result data.

Fluctuation (%)	Mean	Standard deviation	Average trade
0	0	0	0
10	245	23	1
20	471	189	2
30	296	405	3
40	150	433	4
50	41	560	4
60	-297	642	5
70	-606	662	5
80	-729	707	5
90	-1953	474	6
95	-2515	446	7

In previous section, we have discussed how the market fluctuation (or volatility) can affect our trading algorithms. Given this volatility information, we can formalize and integrate it into our trading algorithm and it should help to improve our trading algorithm profitably. Based on this assumption, market fluctuation should be taken into account during decision making. There are many ways to calculate volatility. The most common one is standard deviation of asset closing prices over a year. However, since we only concentrate on intra-day trades, it may not be very suitable here. The central concept of volatility is finding the amount of change in the price of the asset over a period of time. Therefore we can measure it simply by the following formula:

$$\text{Volatility}(t) = \text{SMA}_n((\ln(\text{price}(t)) - \ln(\text{price}(t-1))) \times C) \quad (11)$$

where \ln is the natural logarithm, n is the number of periods, t is current time, C is a constant that enlarge the digit to significant figure.

SMA (Simple Moving Average) is formed by computing the average (mean) price of a security over a specified number of periods. It is possible to create moving averages from the Open, the High, and the Low data points, most moving averages are created using the closing price. For example, a 5-day simple moving average is calculated by adding the closing prices for the last 5 days and dividing the total by 5

$$\text{SMA}(t) = \frac{\text{Close}(t) + \text{Close}(t-1) + \dots + \text{Close}(t-n+1)}{n} \quad (12)$$

where n is the number of periods, t is the time.

From these formulas, we can see that the volatility is actually a period of price difference in average, and there is no upper or lower boundary. When a price is advancing while increasing volatility, the formula result in positive, in contrary when price is falling

while increasing in volatility, the formula results in negative; and when there is no volatility it tends to be zero. By observation, we find the highest reading of the equation is around “±18”. Now if we follow on previous fluctuation results, when it is over 50% we should consider taking action, such as getting out of a position; and when it is under 10% we may not want to open any position, as there is a chance that we will make a loss.

To include volatility measurement during decision-making, we add an additional factor into our fuzzy inference system. Fig. 12 shows the fuzzy membership function for volatility.

Based on this additional input factor, we revise the fuzzy membership rules as follows:

1. IF VIX IS *nagHigh* OR VIX IS *posHigh* THEN POS IS *doNothing*.
2. IF RSI IS *whipsaw* OR MTM IS *whipsaw* or VIX IS *low* THEN POS IS *doNothing*.
3. IF RSI IS *overSold* AND MTM IS *long* AND VIX IS *pMiddle* THEN POS IS *goLong*.
4. IF RSI IS *overSold* AND MTM IS *short* AND VIX IS *nMiddle* THEN POS IS *goShort*.
5. IF RSI IS *overBought* AND MTM IS *short* AND VIX IS *nMiddle* THEN POS IS *goShort*.
6. IF RSI IS *overBought* AND MTM IS *long* AND VIX IS *pMiddle* THEN POS IS *goLong*.

The experimental results from simulation with Fuzzy TF with Volatility Algorithm are detailed in Section 4.

3. Trend recalling algorithm

Since TF is an objective mechanism that is totally free from human judgment and technical forecasting, the trends and patterns of the underlying data play an absolutely influential role in deciding its ultimate performance. It was already shown in Fong, Tai, and Si (2011) that market fluctuation adversely affects the performance of TF. Financial cycles are inevitable phenomena and it is a controversy whether cycles can be predicted or past values cannot forecast future values because they are random in nature. Nonetheless, we observed that cycles could not be easily predicted, but the abstract patterns of such cycles can be practically recalled by simple pattern matching. The formal interpretation of financial cycle (or better known as economic cycle) refers to economy-wide fluctuations in production or economic activity over several months or years. Here we consider it as the cycle that run continuously between bull market and bear market, some people refer this as market cycle (although they are highly correlated), In gen-

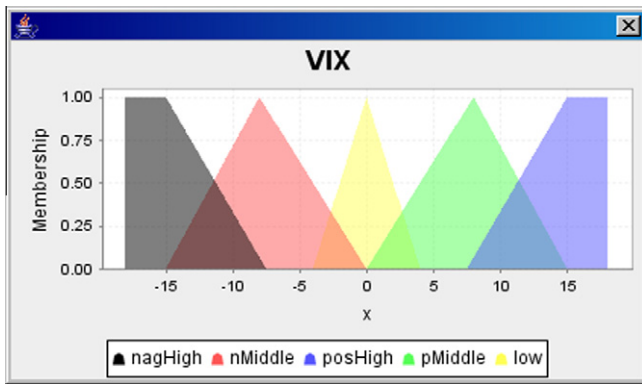


Fig. 12. Volatility membership function.

eral a cycle is made of four stages, and these four stages are: “(1) consolidation (2) upward advancement (3) culmination (4) decline” (Weistein, 1988). Despite being termed as cycles, they do not follow a mechanical or predictable periodic pattern. But similar patterns are observed to always repeat themselves in the future, just as a question of when, though in approximate shapes. We can anticipate that some exceptional peak (or other particular pattern) of the market trend that happen today, will one day happen again, just like how it did happen in history. For instance, in the “1997 Asian Financial Crisis” (Wikipedia, 1997 Asian Financial Crisis, the Hang Seng Index, 2012) plunged from the top to bottom (in stages 3 to 4); then about 10 years later, the scenario repeats itself in the “2008 Financial Crisis” (Wikipedia, Financial Crisis of 2007–2010, 2012).

Dow theory (Schannep, 2008) describes the market trend (part of the cycle) as three types of movement. (1) The “primary movement”, main movement or primary trend, which can last from a few months to several years. It can be either a bullish or bearish market trend. (2) The “secondary movement”, medium trend or intermediate reaction may last from 10 days to three months. (3) The “minor movement” or daily swing varies from hours to a

day. Primary trend is a part of the cycle, which consist of one or several intermediate reaction and the daily swings are the minor movements that consist of all the detailed movements. Now if we project the previous assumption that the cycle is ever continuously rolling, into the minor daily movement, can we assume the trend that happens today, may also appear some days later in the future?

Here is an example for this assumption; Fig. 13 shows two intra-day 2009-12-07 (top) and 2008-01-31 (bottom) trend graphs of Hang Seng Index Futures. Although they are not exactly the same, in terms of major upwards and downwards trends the two graphs do look alike.

This observation provides a foundation for recalling trading strategies that are based on searching for similar patterns from the past. This concept is valid for TF because TF works by smoothing out the averages of the time series. Minor fluctuations or jitters along the trend are averaged out. This is important because TF is known to work well on major trending cycles aka major outlines of the market trend.

Based on this underpinning, an improved version of Trend Following algorithm is proposed in this section, which looks back to the past for reference of selecting the best trading strategy. The design of a TF system is grounded on the rules that are summarized by Covell, into the following five questions (Covell, 2004):

1. How does the system determine what market to buy or sell at any time?
2. How does the system determine how much of a market to buy or sell at any time?
3. How does the system determine when you buy or sell a market?
4. How does the system determine when you get out of a losing position?
5. How does the system determine when you get out of a winning position?

The first and second questions are already answered in our previous work (Fong & Tai, 2009). The third question is rather chal-



Fig. 13. Intra-day of 2009-12-07 and 2008-01-31 day trend graphs.

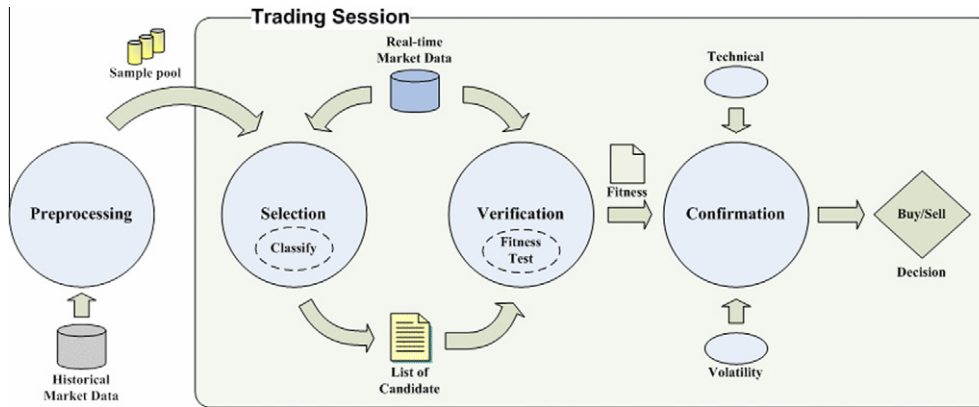


Fig. 14. Improved TF process with recalling function.

lenging, that is actually the core decision maker in the TF system and where the key factor in making profit is; questions 4 and 5 are related to it. Suppose that we have found a way to identify trend signal (to buy or sell), and we have a position opened. Now if the system along the way identifies another trend signal, which complies with the current opened position direction, then we should keep it open, since it suggested that the trend is not yet over. However, if it is counter to the current position, we should probably get a close out, regardless whether you are currently winning or losing, as it indicates a trend reversion.

Our improved TF algorithm is designed to answer this question: when to buy or sell. It is a fact that financial cycles do exist, and it is hypothesized that a trend on a particular day from the past could happen again some days later. This assumption supports the recalling trading mechanism, which is the basic driving force that our improved trend following algorithm relies on. The idea is expressed in Fig. 14. As it can be seen in the diagram there's four major processes for the decision making. Namely they are *Pre-processing*, *Selection*, *Verification* and *Analysis*. Fig. 14 also shows the process of which our improved TF model works by recalling a trading strategy that used to perform well in the past by matching the current shape of the pattern to that of the old time. A handful of such patterns and corresponding trading strategies are short-listed; one strategy is picked from the list after thorough verification and analysis.

3.1. Pre-processing

In this step, raw historical data that are collected from the past market are archived into a pool of samples. A sample is a day trend from the past with the corresponding trading strategy attached. The trend is like an index pattern for locating the winning trading strategy that is in the format of a sequence of buy and sell decisions. Good trading strategy is one that used to maximize profit in the past given the specific market trend pattern. This past pattern which is deemed to be similar to the current market trend, is now serving as a guidance to locate the strategy to be applied for decision making during the current market trade session.

Since the past day trend that yielded a great profit before, reusing it almost can guarantee a perfect trading strategy that is superior to human judgment or a complex time series forecasting algorithm. The past samples are referenced by best trading strategies on an indicator that we name it as "EDM" (exponential divergence in movement). EDM is a crisp value indicator that is based on two moving average differences

$$EDM(t) = f(EMAs(t) - EMAl(t)) \quad (13)$$

$$EMA(t) = \left(price(t) - EMA(t-1) \times \frac{2}{n+1} \right) + EMA(t-1) \quad (14)$$

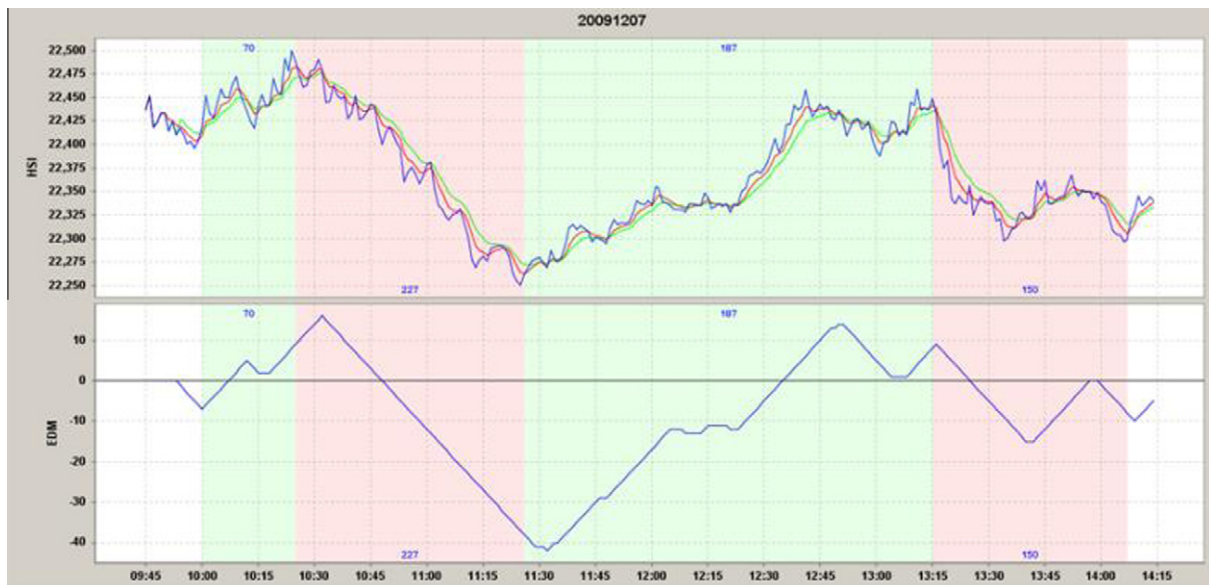


Fig. 15. Example of EDM and preprocessed trend of 2009-12-17.

where $price(t)$ is the current price at any given time t , n is the number of periods, s denotes a shorter period of $EMA(t)$ at time t , l represents a longer period $EMA(t)$, $f(\cdot)$ is a function for generating the crisp result. The indicator sculpts the trend; and based on this information, a TF program finds a list of best trading strategies, which can potentially generate high profit. An example of pre-processing a 2009-12-07 day trend that shows the EDM is depicted in Fig. 15. As indicated from the diagram, the program first found a long position at 10:00 followed by a short position at around 10:25, then a long position at 11:25, finally a short position around 13:51 and closes it out at the end of the day, which reaps a total of 634 index points. In Hong Kong stock market, there is a two hours break between morning and afternoon sessions. To avoid this discontinuation on the chart, we shift the time backward, and joined these two sessions into one, so 13:15 is equivalent to 15:15.

3.2. Selection

Once a pool of samples reached a substantial size, the improved TF with recalling mechanism is ready to use. The stored past samples are searched and the matching ones are selected. The goal of this selection process is to find the most similar samples from the pool, which will be used as a guideline in the forthcoming trading session. A foremost technical challenge is that no two trends are exactly the same, as they do differ day by day as the market fluctuates in all different manners. Secondly, even two sample day trends look similar but their price ranges can usually be quite different. With consideration of these challenges, it implies that the sample cannot be compared directly value to value for a simple match. Some normalization is necessary for enabling some rough matches. Furthermore the comparison should allow certain level

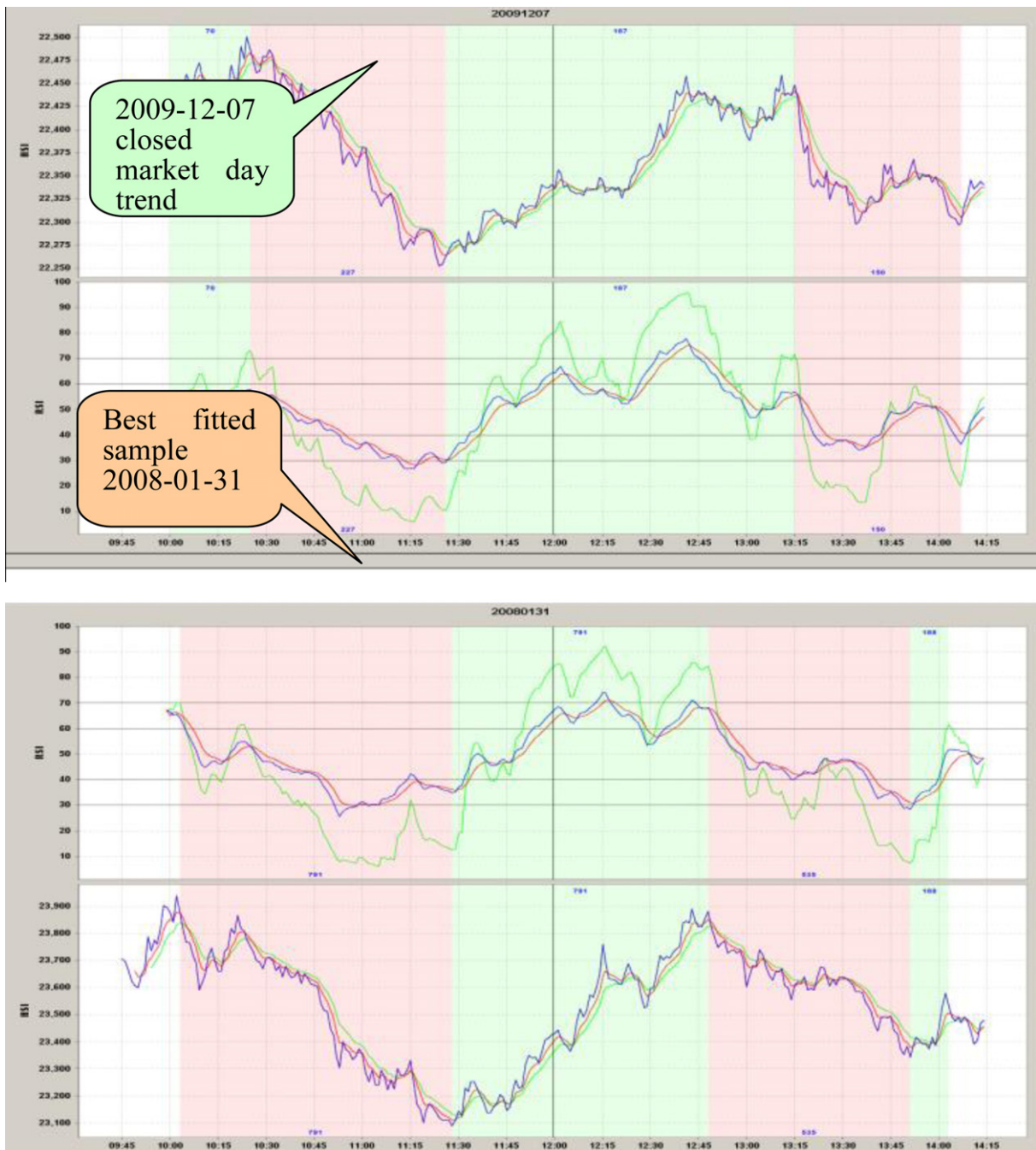


Fig. 16. Example of 2009-12-07 sample with best fitness (2008-01-31) day trend and RSI graph.

Table 2

Fitness test applied on 2009-12-07 at the time 14:47.

List of Candidate		List After Fitness Test		
Rank	Sample	Sample	Fitness	Rank
0	2008-01-25	2009-03-17	373	15
1	2008-03-07	2007-09-13	357	7
2	2009-08-18	2008-12-15	341	8
3	2008-01-23	2008-02-01	339	9
4	2008-11-10	2008-01-31	326	11
5	2007-08-29	2007-12-05	305	14
6	2008-04-16	2008-01-25	300	0
7	2007-09-13	2008-02-21	296	12
8	2008-12-15	2009-08-18	294	2
9	2008-02-01	2008-01-23	273	3
10	2009-11-25	2008-11-10	270	4
11	2008-01-31	2007-08-29	179	5
12	2008-02-21	2009-11-25	165	10
14	2007-12-05	2008-03-07	143	1
15	2009-03-17	2008-04-16	3	6

of fuzziness. Hence each sample trend should be converted into a normalized graph, and by comparing their rough edges and measure the difference, it is possible to quantitatively derive a numeric list of similarities. In pattern recognition, the shape of an image can be converted to an outline like a wire-frame by using some image processing algorithm. The same type of algorithm is used here for extracting features from the trend line samples for quick comparison during a TF trading process.

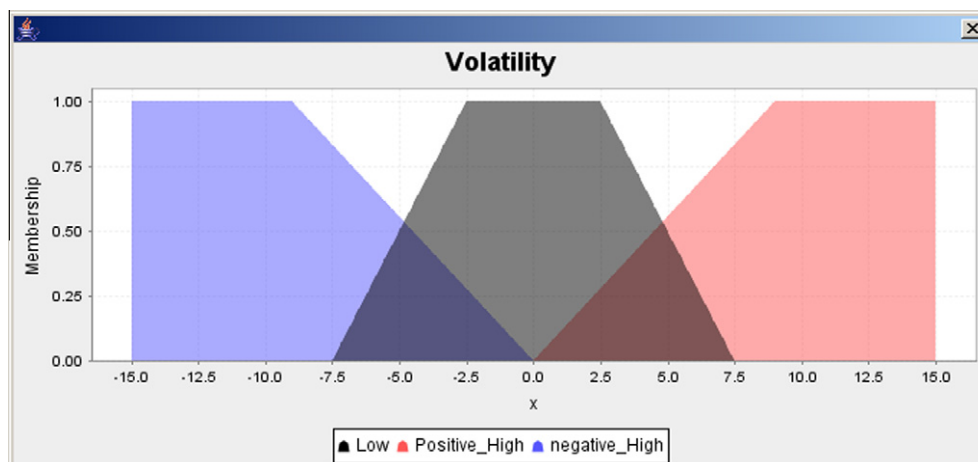
In our methodology, each sample is first converted into a normalized graph, by calculating their technical indicators data. Indicators such as RSI and STC have a limited value range (from 1 to 100) which is suitable for fast comparison, and they are sufficient to reflect the shape of a trend. In other words, these indicators help to normalize each trend sample into a simple 2D line graph. We can then simply compare each of their graphical difference by superimposing these line graphs on top of each other for estimat-

Table 3

Results of the overall simulation runs.

Simulation	Static	Dynamic	Fuzzy	FuzzyVix	Recalling
Total Index Point	1115	1142	1476	2981	7201
Net worth (HKD)	55,750	57,100	73,800	149,050	360,050
Total trade	1088	987	1578	1149	1186
Cost	41,997	38,098	60,911	44,351	45,780
P&L	13,753	19,002	12,889	104,699	314,270
Total ROI (%)	19	26	17	141	425

ing the differences. This approach produces a hierarchical similarity list, such that we can get around with the inexact matching problem and allows a certain level of fuzziness without losing their similarity attributes. Fig. 16 shows an example of two similar samples trend graphs with the RSI displayed.

**Fig. 17.** Volatility membership definition for trend recalling.

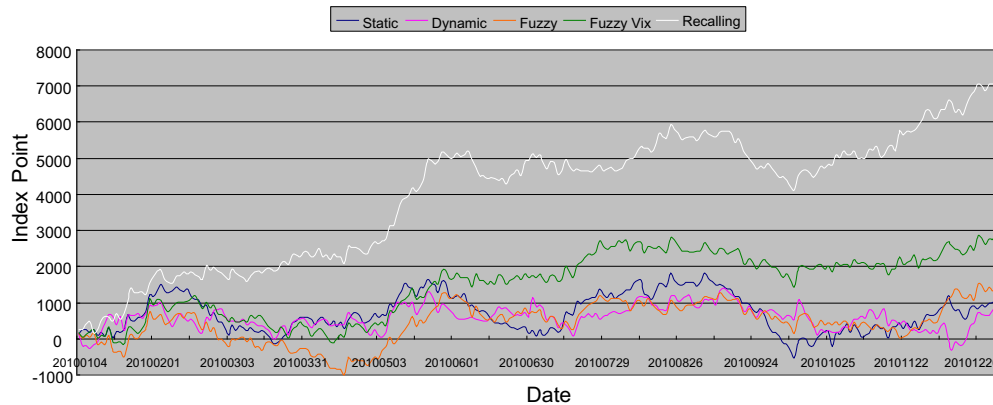


Fig. 18. Performance comparison of five TF algorithms.

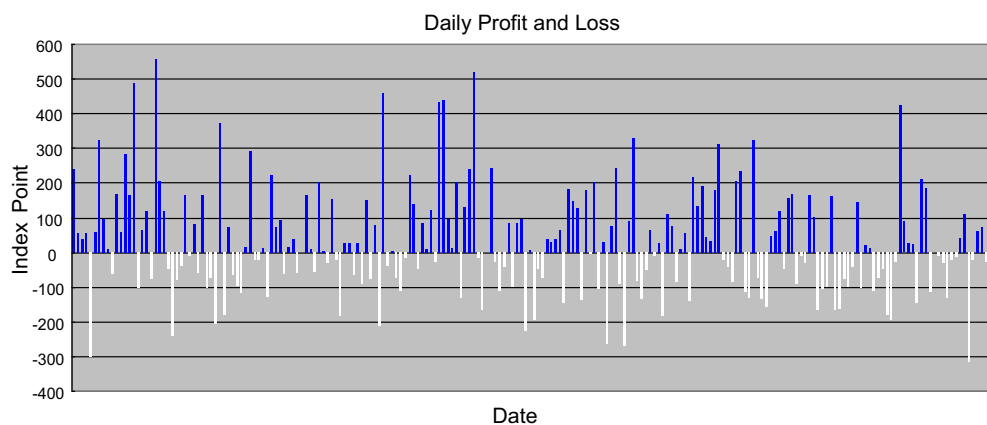


Fig. 19. Daily profit and loss diagram of TF algorithm with recalling function.

3.3. Verification

During this process, each candidate from the list is tested against the current market state. Each trend sample corresponds to a specific trading strategy (that was already established in the pre-processing step). Each trading strategy is extracted and evaluated against historical to determine how well it performed as a trial. Each of their performances is then recorded and the trial performance is used as a criterion to rearrange the list. An example of the sample list before and after the fitness test is shown in Table 2. The fitness test was run on the 2009-12-07 during the middle of simulated trade session.

Verification is needed because the selection of these candidates is by a best effort approach. That is because the market situations now and the past may still differ to certain extent.

3.4. Confirmation

After the verification process is finished, the candidate list is re-sorted according to the fitness test results. The fittest one will be used as the reference of subsequent trading strategy during the TF decision making. In order to further improve the performance on top of the referencing to the past best strategy, some technical analysis is also referenced as well. By the advice of Weissman from his book (Weissman, 2004), the two-moving average crossover system is used as a signal confirmation. (The two-moving average crossover system entails the rise of a second, shorter-term moving average.) But instead of using simple moving average, EMA – exponential moving average with RSI should be used, that is a short-

term RSI EMA and a long-term RSI EMA crossover system. When a trend is identified and it appears as a good trading signal, the crossover system must also be referenced and check if it gives a consistent signal. Otherwise the potential trend is considered as a false signal or noise. For example in our case the trading strategy from the recalled sample hints a long position trade. We check if RSI crossover system shows a short-term EMA crossing over its long-term EMA or not.

In addition to validating the hinted trading signals from past strategy, market volatility should be considered during decision making. By observing how the equation responds to historical data, we can find the maximum volatility as ± 15 . Based on the previous fluctuation test result, we define the volatility fuzzy membership function for trend recalling in Fig. 17.

During the trading session, volatility will be constantly referenced while the following rules apply at the TF system:

1. IF volatility is too positive high and long position is opened THEN close it out.
2. IF volatility is too positive high and no position is opened THEN open short position.
3. IF volatility is too low THEN do nothing.
4. IF volatility is too negative high and short position is opened THEN close it out.
5. IF volatility is too negative high and no position is opened THEN open long position.

These rules have a higher priority over the trade strategies, such that when the condition has met any of these rules, it will take over

the control regardless of what decision that the trade strategies has made. The improved TF algorithm with recalling function is programmed as an automated trading simulator, written in JAVA.

4. Experimental results

In this section, we perform simulation experiments on the five trading algorithms previously proposed based on The Hong Kong Index Futures Contracts. All trials of simulations are run on historical market data within 2.5 years prior to the year of 2010. A price that spread between bid and ask price is considered on each trade.

In the simulation, each trade is calculated in the unit of index point. Each index point is equivalent to 50 HKD, which is subject to overhead cost as defined by Interactive Broker unbundled commission scheme at 19.3 HKD per trade. Six attributes are used to compare the performance of the algorithms including net worth, cost, and profit & lost.

$$\text{Net Worth} = \text{Index Point} \times \text{CM} \quad (15)$$

$$\text{Cost} = \text{Trade} \times C \quad (16)$$

$$P\&L = \text{Net Worth} - \text{Cost} \quad (17)$$

CM is the contract multiplier that is defined by the Hong Kong Futures Exchange, which currently set as HKD 50 per index point. C is the commission that the broker charges. According to the IB (Group, 2012), bundled commission scheme during the year of 2007–2009, each trade cost 60 HKD commission. ROI is the prime performance index that is based on Hong Kong Exchange current Initial margin requirement (each contract 7400 HKD in year of 2010).

From the simulation result from Table 3, we find that trend recalling algorithm achieves the best performance at the end of the experimental run. This is a significant result as it implies the proposed TF algorithm can reap more than three folds of whatever the initial investment is over just 2.5 years of trading. In Fig. 18, we summarize the performance comparison of five TF algorithms. Fig. 19 shows a horizon of trading results and we can observe that in overall, there are more profits than losses.

5. Conclusion and future work

Trend following has been known as a rational stock trading technique that just rides on the market trends with some preset rules for deciding when to buy or sell. TF has been widely used in industries, but none of it was studied academically in computer science communities. In this paper, we pioneer in formulating TF into algorithms and evaluating their performance.

First, we describe statistic and dynamic TF algorithms. Next, to improve the performance, we introduce fuzzy logic into our trade system, which forms our third and fourth versions of trading algorithms. The properties of these trading algorithms are generally built upon the experiences of previous trading algorithms, such as the membership definition and fuzzy sets generation. All these trading algorithms are later verified on Hang Sang Index futures market in a simulated environment, and the result is encouraging.

These trading algorithms show an outstanding performance in the wild bullish and bearish markets between the years of 2007–2009. However, during the year of 2010, they seem to be under performing, as they are no longer generating great profits. This is due to the large flip-flop changes of the market. We observe that frequent market trend fluctuations in large extents deter trend following algorithms and cause the degradation in performance.

To alleviate this problem, we extend the original TF algorithm by adding a market trend recalling function in our last trading

algorithm. Trading strategy that used to make profit from the past is recalled for serving as a reference for the current trading. The trading strategy is recalled by matching the current market trend with the most similar past market trend which is known to produce good profit. Matching market trend patterns is a complex task since patterns can be quite different in details. The concept of financial cycle was also taken into account of in the trend recalling algorithm so that the algorithm is adaptable to market behavioral changes. This trading algorithm is also verified on Hang Sang Index futures contracts in simulated environment. Our simulation results show that the improved TF model with trend recalling is able to generate profit from stock market at more than three times of ROI.

In summary, contributions of this research are summarized as follows:

- Developed an automated trading system prototype, which provides a cornerstone for future development, and experimental platform for evaluating trading algorithms and programmed.
- Provided an alternative view and comparative of two kinds of trading algorithms (predictive model vs. reactive model).
- Proposed five innovative trading algorithms based on trend following concepts.

As for the future work, we are planning to test the algorithms on other stock market data. We are also planning to further improve the calculation of P and Q in dynamic TF algorithms by applying time series segmentation methods.

Acknowledgement

Credits go to Tai Kam Fong, Jackie, who is the Master student of the University of Macau in year 2010, for his theoretic contributions and for programming the simulator.

References

- Covel, M. W. (2004). *Trend following: How great traders make millions in up or down markets*. Financial Times Prentice Hall.
- Fong, S., & Tai, J. (2009). *The application of trend following strategies in stock market trading* (pp. 1971–1976). <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5331484>>.
- Fong, S., Tai, J., & Si, Y. W. (2011). Trend following algorithms for technical trading in stock market. *Journal of Emerging Technologies in Web Intelligence (JETWI)*, 3(2), 136–145.
- Fontanills, G. A., & Gentile, T. (2002). *The volatility course*. Wiley.
- Group, I. B. (2012). About the interactive brokers group. <<http://www.interactivebrokers.com/en/general/about/about.php>> Retrieved 18.02.12.
- Hull, J. C. (2000). *Options, futures, and other derivatives* (4th ed.). Prentice Hall.
- Investopedia (2012). Whipsaw. <<http://www.investopedia.com/terms/w/whipsaw.asp>> Retrieved 18.02.12.
- jFuzzyLogic (2012). An open source fuzzy logic library and FCL language implementation. <<http://jfuzzylogic.sourceforge.net/html/index.html>> Retrieved 18.02.12.
- Mayo, H. B. (2003). *Financial institutions, investments, and management: An introduction* (8th ed.). South-Western College Pub..
- Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance.
- Schannep, J. (2008). *Dow theory for the 21st century: Technical indicators for improving your investment results*. Wiley.
- Weissman, R. L. (2004). *Mechanical trading systems: Pairing trader psychology with technical analysis* (1st ed.). Wiley.
- Weinstein, S. (1988). *Stan Weinstein's secrets for profiting in bull and bear markets* (1st ed.). McGraw-Hill.
- Wikipedia (2012a). 1997 Asian Financial Crisis. <http://en.wikipedia.org/wiki/1997_Asian_Financial_Crisis> Retrieved 18.02.12.
- Wikipedia (2012b). Financial Crisis of 2007–2010. <http://en.wikipedia.org/wiki/Financial_crisis_of_2007> Retrieved 18.02.12.
- Wikipedia (2012c). Momentum (technical analysis). <http://en.wikipedia.org/wiki/Momentum_%28technical_analysis%29> Retrieved 18.02.12.
- Zadeh, L. (1973). Outline of a new approach to the analysis of complex system and decision process. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1), 28–44.