

Stock Trading using Linear Genetic Programming with Multiple Time Frames

Garnett Wilson
Afinin Labs Inc.
St. John's, Canada
gwilson@afinin.com

Derek Leblanc
Afinin Labs Inc.
St. John's, Canada
dleblanc@afinin.com

Wolfgang Banzhaf
Afinin Labs Inc.
St. John's, Canada
banzhaf@afinin.com

ABSTRACT

A number of researchers have attempted to take successful GP trading systems and make them even better through the use of filters. We investigate the use of a linear genetic programming (LGP) system that combines GP signals provided over multiple intraday time frames to produce one trading action. Four combinations of time frames stretching further into the past are examined. Two different decision mechanisms for evaluating the overall signal given the GP signals over all time frames are also examined, one based on majority vote and another based on temporal proximity to the buying decision. Results indicated that majority vote outperformed emphasis on proximity of time frames to the current trading decision. Analyses also indicated that longer time frame combinations were more conservative and outperformed shorter combinations for both overall upward and downward price trends.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms, Performance, Experimentation

Keywords

computational finance, linear genetic programming, algorithmic trading

1. INTRODUCTION

Researchers who apply genetic programming (GP) or evolutionary computation methods for analysis of financial markets have a number of different approaches to discover profitable opportunities in the price time series they analyze. Some researchers train a system on an extended period of time, and then allow the evolved solution of static rules to

operate on a subsequent extended test period [3], [4]. Other researchers have found it beneficial to continually train on a moving window, and only act in anticipation of the immediate future of the price changes based on the immediate past [1], [8]. Others go further than this, combining either of these systems with filters that are used to improve the confidence that a GP signal is actually being evolved on an inherently trending time series [3], [4], [7]. In this work, we combine the notions of filtering GP signals and moving windows of varying length in a linear genetic programming system (LGP). In particular, we use the LGP system to determine whether price series in partially overlapping time frames are collectively consistent in producing a particular algorithm signal.

The next section describes existing literature on the use of filters with GP and the notion of predictable windows with price series. Section 3 describes the linear GP system applied to stock trading using multiple time frames. Section 4 describes trading performance characteristics of the system and its profitability. Conclusions follow in Section 6.

2. PREVIOUS WORK

To the authors' knowledge there are no GP systems in the literature that run over multiple time frames to create more accurate signals. However, there are a few examples of the application of pretests to market data used in conjunction with evolutionary computing to determine if the price series is predictable (and thus appropriate for GP analysis). The first instance of a GP system with filtering applied to a financial domain was Kaboudan [4]. Kaboudan proposed a metric called the " η " statistic that indicates predictability. To establish η , a GP used the unaltered price series and a shuffled version of the price series. The sum of squared errors (SSE) between the results predicted by GP and the unaltered price series is calculated, and compared to the SSE of the GP and the randomly shuffled series over numerous GP trials. Kaboudan examined the price series of eight Dow Jones stocks and found that predictability was inversely related to time elapsed between price samplings. Chen et al. [3] expand on Kaboudan's work by comparing GP results to different search methods rather than running GP on shuffled time series. The authors compare GP to both intensive random search and a search method called "lottery" trading based on a random variable. The comparison of GP to these random search methods accurately indicated predictability of the time series in some of the nine markets tested. Neely and Weller [6] used GP to attempt to forecast the volatility of future currency prices in a time series rather than future

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '11, July 12–16, 2011, Dublin, Ireland.

Copyright 2011 ACM 978-1-4503-0557-0/11/07 ...\$10.00.

prices themselves, but they found that the technique was not more useful than recognized technical volatility measures.

Rather than testing predictability of time series using GP, other researchers have attempted to restrict GP signals to improve their accuracy. Li and Tsang [5] use a GP hybrid called “FGP” that includes decision trees of rule sets. The GP was given a minimum and maximum of signals that were permitted when predicting prices based on the training data. The authors reported that the failure rate was reduced, but they also found that there were more missed opportunities by the system.

In this work we do not train GP on shuffled versions of actual price series or compare results of GP to randomized search techniques to increase confidence in underlying predictability of a price series. In contrast, we use GP multiple times, each time on a separate unaltered time series. The time series are progressively longer in order to increase the confidence that the GP is making a signal based on a more sustained trend.

3. LGP MULTIPLE TIME FRAME TRADING SYSTEM

3.1 LGP Algorithm and Parameters

The algorithm applied to stock trading in this work is linear genetic programming (LGP). LGP evolves individuals composed of binary strings and associated registers, in contrast to the tree-based individuals of traditional GP. The bits that make up one individual can be further divided into separate instructions. These instructions can be executed sequentially, with the subresult of each instruction stored in one of the individual’s registers. Each instruction performs an operation corresponding to a member of the function set that it references, and uses subresults in registers or external data as operands. Thus, each LGP individual represents a small machine-language type program that is evaluated by a fitness function upon termination, often by examining a final solution in one of the individual’s registers. LGP is now considered an established hallmark form of GP, and further details of general LGP implementation can be found in [2].

In this work, we apply a LGP automated trading implementation to the intraday trading of four stocks: BRK-B, RIMM, RY, and AAPL. An initial period was reserved to establish values of technical indicators, following which the GP fitness was evaluated on data corresponding to a moving window of n minutes. Individuals represented sets of trading rules, based on functions in the function set (to be described). For each window of trading minutes m to n , each of m to $n - 1$ minutes were used for calculation of a trading decision, with $m + 1$ to n left for the evaluation of the signal based on the preceding minutes. Data used for the determination of a trading decision were normalized using two-phase preprocessing similar to treatment of stock data in [1]: All daily values were transformed by division using a lagged moving average, and then normalized using linear scaling into the range $[0, 1]$ using

$$v_{scaled} = \frac{v_t - l_n}{h_n - l_n} \quad (1)$$

where v_{scaled} is the normalized trading value, v_t is the transformed trading value at time step t , h_n is highest transformed value in the last n time steps, l_n is the lowest trans-

formed value in the last n time steps, and n is the length of the time lag chosen for the initial transformation.

In addition to an instruction set, each LGP individual possesses four registers, a flag for storing the current value of logical operators, and an output (trade) register for storing the value corresponding to a trade signal following execution of the instruction set. If the value of the trade register is 0, no trade is conducted. If the value in the trade register corresponds to a value in the range $+/-[0, 1]$, it indicates the strength of a buy or sell signal based on positive or negative value, respectively. The fitness function is the profitability of the GP individual after evolution over a series of prices when acting on a next unknown price (described in detail below). For each trade conducted, there is a moderate \$10 trading commission fee (with a round trip commission of \$20).

The LGP function set includes standard mathematical operators and logical operators. In addition, established technical analysis metrics such as moving average, momentum, and channel breakout are used. Moving average is the mean of the previous n share prices. The momentum is the ratio of a time-lagged price to the current price. Channel breakout typically uses Bollinger bands around a n -minute moving average of the price at $+/- 2$ standard deviations of the price movement over the last n minutes to alert the trader of significant movements in rates. Each LGP tournament consisted of 1000 rounds. XOR mutation on individual instructions was used with a probability of 0.5, and crossover occurred with a probability of 0.9.

3.2 Multiple Time Frames

The best individual, consisting of the best trading rule set, is used by a “live” trading algorithm. The live trading system provides known information to the LGP for minutes m to n . The LGP algorithm returns a signal for the live trading system, which is used as the basis of its trade on the following minute, $n + 1$. The net value of the best evolved individual (trading rules) given the signal of the trade register over all the fitness cases is the buy or sell signal to the “live” trading system. The best LGP individual can thus recommend a buy, sell, or hold to the live system. Based on the signal, the live system purchases as much stock as possible using its current cash assets, sells all shares, or does nothing, respectively. The sell signal of the GP system is generally issued as a sell to stop losses in a downtrend, so an additional sell mechanism to take profit based on momentum is used that closes the trade when a future higher price is met following a buy signal. The transactions of the live trading system are actually based on unknown data and determine the success of the automatic trader.

For a time frame of m to n minutes, there are $n-1$ fitness cases. In the smallest time frame, that of 5 minutes, there are 4 fitness cases for $m = 1$ to $n = 5$ in the 5 minute window and a signal is made for the unknown (6th) minute following the time frame. For example, in the smallest (5 minute) time frame, the LGP will train on minute 1 with minute 2 unknown, then minute 1, 2 with 3 unknown, then 1, 2, 3 with 4 unknown, and finally 1, 2, 3, 4 with 5 unknown (giving 4 fitness cases in total). Based on the mean value over all the trade registers of the best individuals after each fitness case, the overall signal is determined. The automated system then trades in anticipation of the next minute (the unknown).

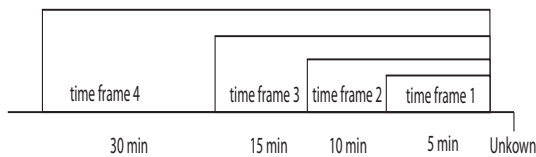


Figure 1: The LGP trains repeatedly on progressively longer time scales to generate the signal for the “live” trading system to act on unknown values.

This smallest time frame, given minute-to-minute trading decisions, provides an algorithm that is reactive to change. In this paper we examine tempering the reactivity of the smaller time scale with additional information from GP run on longer time frames into the past. For each time frame, the LGP algorithm evaluates known (provided) values of m to n , with minute $n + 1$ being unknown and GP being run on $n - 1$ fitness cases. Now there are multiple signals, which may or may not be the same, since the LGP trains over longer time frames consisting of more fitness cases as n increases. As the number of time frames increases, the recommended action corresponds to trading strategies for longer trends. Combining the signals for shorter and longer time frames creates a more conservative (safer) signal for buying and selling actions. In this work, we found it useful to examine the time frames of 5, 10, 15, and 30 minutes. The process is shown in Figure 1.

Once the GP algorithm has been run on each of the progressively increasing time frames, an overall trading signal must be determined from the signals generated by the GP for each time frame. Two methods for deciding this action are examined. The first decision method is called “Majority” and uses a buy or sell signal if that signal accounts for half (or more) of all the signals for each time scale. The second method is called “Temporal Proximity” and uses a buy or sell signal if that signal is present for (at least) all of the time frames back from the most recent time frame f to time frame $f - \lfloor f/2 \rfloor$ for f time frames. Hold signals are not counted toward an action decision, but in the case of both majority and temporal decision methods holds are issued if no decision can be made. The majority signal decision is designed to base a decision to buy or sell on an overall trend over all the time frames, irrespective of the latest price movement in the most current time frame. For instance, if the price trend has been increasing for the past 30 minutes overall given four time frames and a 30 minute and 15 minute time frame recommend buys but the very latest time frame recommends a sell, the majority decision recommends a buy. There is a slight aggressiveness of the system toward buying: If there is a tie between buy and sell, buy takes preference. In contrast, the temporal proximity decision emphasizes the trend closest to the unknown price point. For example, temporal proximity will recommend a sell if the time frame at 10 minutes and 5 minutes both recommend a sell given four time frames.

4. STOCK TRADING RESULTS

The live trading system is evaluated using minute interval price values for each of 373 minutes of the arbitrarily chosen trading day, October 18, 2010. Results are shown for one trading day since we are most interested in examining behavior in

high volatility, intraday situations within the space of this paper. We use data for stocks selected for their variations in trend, and thus stocks are examined on an individual basis. Two minutes and an initial 15 minutes of values are withheld for data feed verification and seeding of technical indicators, respectively. The stock prices examined were BRK-B (Berkshire Hathaway Class B), RIMM (Research in Motion Limited), RY (Royal Bank of Canada), and AAPL (Apple Inc.). The stocks were chosen in order to provide a number of different trends to evaluate the robustness of the algorithm and for the difficulty these trends present for GP: all are volatile and only one presents an upward price movement throughout the day that is not abrupt. Variation across trials is practically non-existent since the LGP is consistent enough that it will recommend the same action (one of buy, sell, or hold) given the same price values in a time frame window, so a single run is shown. Performance of the LGP over time is examined, followed by examination of trading behavior and overall profitability.

4.1 Performance Trends

The ability of the algorithm to trade over time is examined in this section. The general success of the LGP trading system used in this work was previously established in [8], and in this work we are interested in focusing on the effect of multiple time frames on trading. The system starts the trading day with \$1,000,000 in cash assets. The total worth (total value of cash and shares) in dollars of the live trading system using all time frames as described in the previous section are provided in Figure 2 for each minute. As a baseline, the value of total assets when as much of the initial cash as possible is invested in shares at the start of trading is indicated as *Buy and Hold*. For fairness, trading can start for all time frame combinations only after the number of minutes required for the longest time frame have elapsed.

The most striking aspect of Figure 2 is that when the Majority and Temporal Proximity decision methods are compared for each stock, the majority decision process yields better results in general across time frame combinations. The less restrictive decision process of Majority allowed more time frames to rise above the profitability of the buy-and-hold plot (solid line) compared to Temporal Proximity. To make the comparison with buy-and-hold clearer, the ratio of each time frame combination’s current total worth to buy-and-hold is plotted in Figure 3 for each minute. These graphs indicate that in the case of almost every stock, the ratios achieved by temporal proximity decision do not reach the levels of majority decision made by one or more of the time frames. Based on the overall better performance of the majority decision, we now focus on its analysis.

By examining the buy-and-hold behavior of each stock’s prices, we can note from Figure 2 the behavior of each stock’s price trend throughout the day. We see that BRK-B exhibits an overall sideways trend for the day, with considerable phases of volatility during certain periods. For this type of price trend, we see that the {5, 10, 15} minute time frame combination performs best. RIMM exhibits a price series that (overall) climbs throughout the day, but includes periods of decline, abrupt and gradual gains, and sideways movement. For this type of trend we see that the most conservative time frame combination works best, namely {5, 10, 15, 30}. RY exhibits a fairly consistent gain in price throughout the day, and we again find that {5, 10, 15, 30}

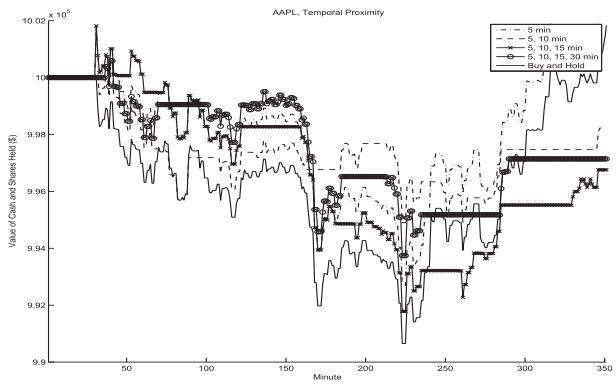
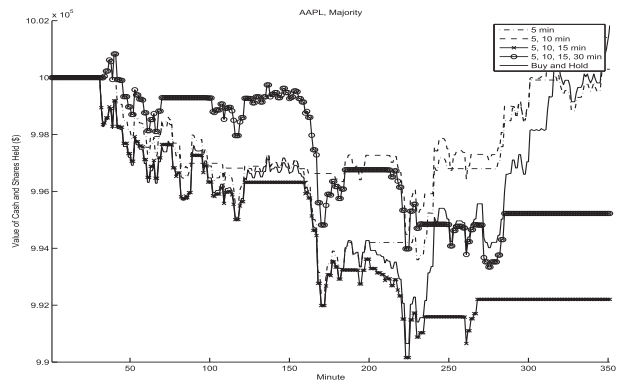
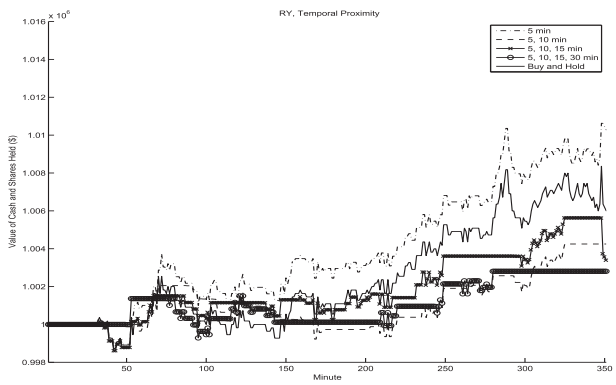
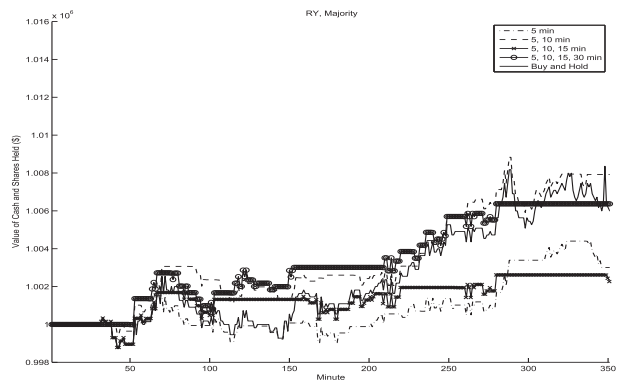
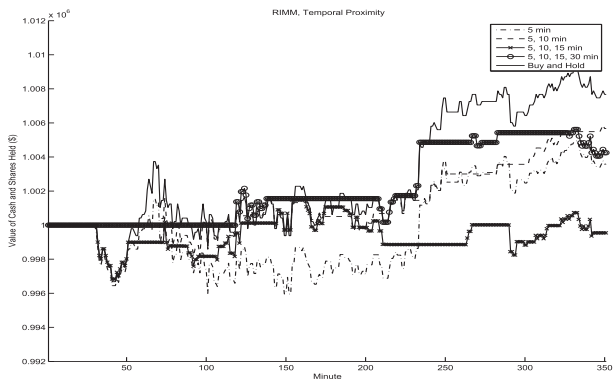
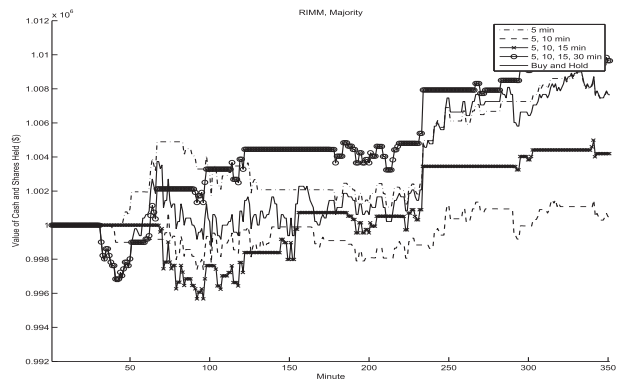
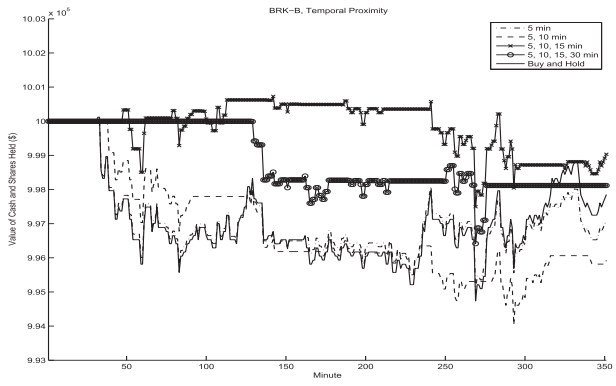
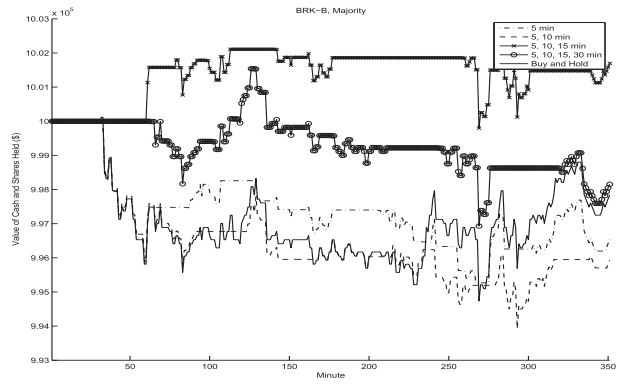


Figure 2: Total value of assets (cash and shares held) given initial \$1,000,000.

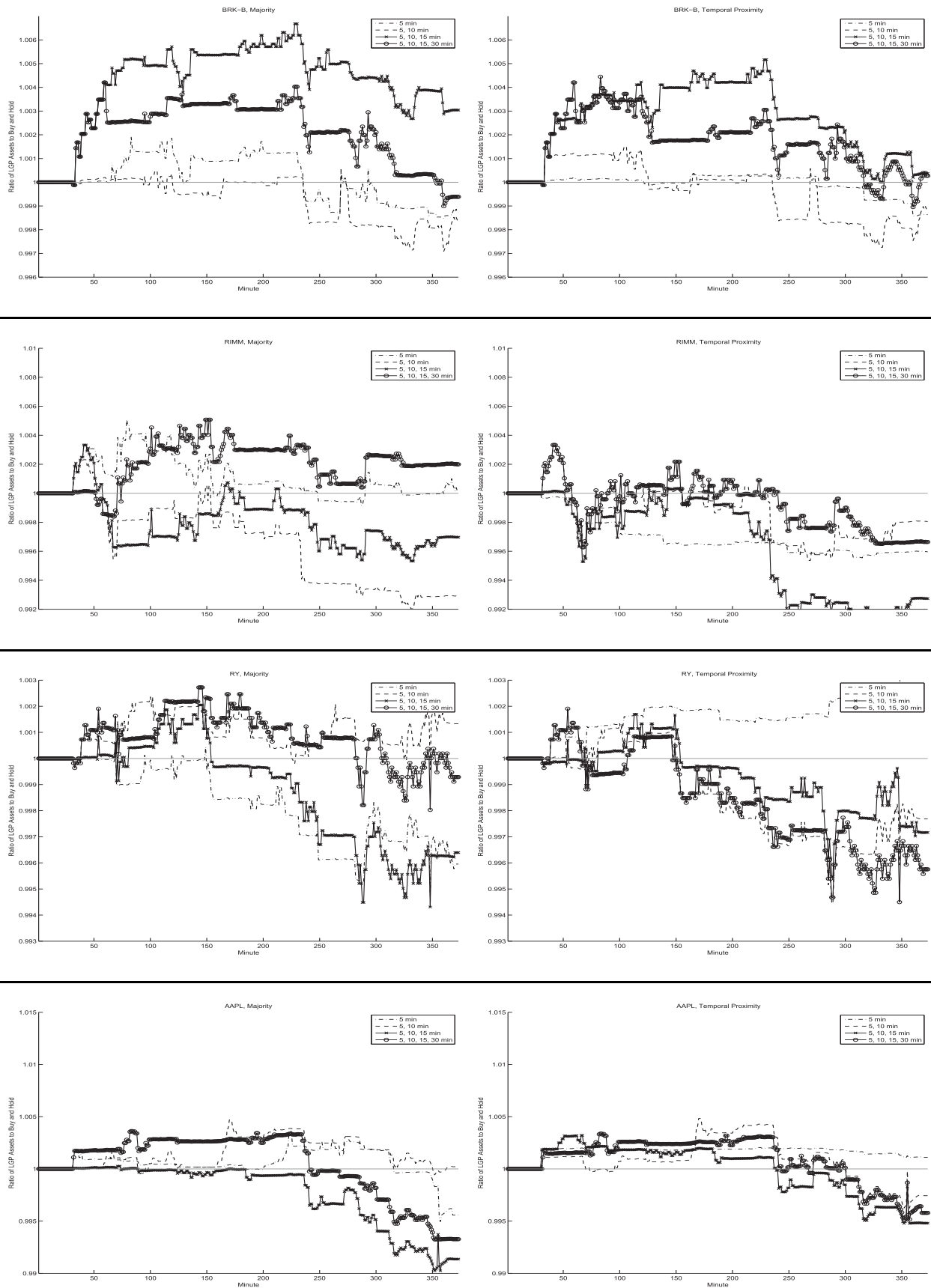


Figure 3: Ratio of total assets to buy and hold given initial \$1,000,000.

(and {5, 10}) perform best. AAPL has a largely declining price trend, followed by gains in the later part of the day (prior to minute 300 until close). In order to avoid losses, the {5, 10, 15, 30} time frame combination dominates for most of the day but it is not able to realize gains at the end of the day because it was out of market during a large price increase that the more reactive, shorter time frame combinations ({5} and {5, 10}) could quickly identify. One cannot see the buy-and-hold trend as a reference point, but Figure 3 shows clearly the ranking of time frame combinations against buy-and-hold throughout the day for each stock. All results just discussed are clearly presented, and it is evident that GP using multiple time frames outperforms buy-and-hold where in many instances GP using only the 5 minute time frame may not. However, the less reactive nature of the longer time frame combinations does not allow them to react to fast moving price changes toward profit and then 5 outperforms others (see end of price series for RY and AAPL in Figure 3). Examining all stocks using Majority decision, the longer {5, 10, 15, 20} time frame combination dominates the shorter time frame combinations other than for the case of RIMM. Overall profitability of the different time frame combinations is discussed in greater detail in Section 4.3. The reactivity of trading behavior related to number of time frames in a combination, which we now examine in greater detail, directly impacts profitability.

4.2 Trading Analysis

The profitability of the different fitness metrics are a result of underlying length and frequency of trades conducted. Figure 4 shows a chart of the trading activity for each GP and time frame combination; both majority and time proximity decision types are shown to determine the effect of decision type on trading activity. Time is graphed on the abscissa, with a trade that ends in profit (after commissions) graphed above the abscissa and a trade that ends in a loss (after commissions) graphed below the abscissa. These graphs allow the reader to examine in a simple way both the total number of trades made by each time scale combination, and how many of those trades were successful. All trades shown in Figure 4 are the result of the GP algorithm issuing a signal to buy, whether or not it resulted in profit at the sell signal of the trade. The sell signal, as discussed in Section 3.2, is the result of a sell signal to take profit or a sell signal to stop further losses.

It is evident from Figure 4 that for each stock, as the number of time frames in a combination increases, the number of trades decreases. Additionally, the overall time spent in the market for each trade increases as the number of time frames in a combination increases. The temporal proximity decision (right side) did not generate any noteworthy behavior differences from the majority decision trading patterns (left side) with respect to trade length or overall trading accuracy. The temporal proximity decision, however, did result in fewer trades overall, which is to be expected due to its more stringent requirement for trade execution. Seeing no overall benefit to the temporal proximity decision method, as mentioned in the last section, we discuss the majority decision results (left side) for the remainder of this work: For BRK-B, where the overall trend of the stock was sideways with initial decline and end of day gains, the number of successful trades for the longest time frame combination is the lowest of all time frame combinations. Similarly, for AAPL

Table 1: Final Profit (%)

	BRK-B	RIMM	RY	AAPL
5 Min.	-0.34	0.70	0.30	-0.30
5, 10 Min.	-0.37	0.11	0.79	0.10
5, 10, 15 Min.	-0.18	0.88	0.64	-0.48
5, 10, 15, 30 Min.	0.20	0.42	0.34	-0.78

which showed a decline over the price series, the two longest time frame combinations showed the lowest number of successful trades. RIMM and RY, for which the daily trend showed a climb in price over the time series, show the highest proportion of profitable trades in the longest time frame combination. We now examine whether the higher accuracy of the longer time frame combinations correspond to higher overall profits.

4.3 Profitability Analysis

There are two analyses of overall profit provided in this section. The final profit at the end of the time period is somewhat arbitrary but is often stated in studies, and is shown in Table 1. A preferable measure is the cumulative profit average over all minutes compared to buy-and-hold throughout the time period, shown in Figure 5. Bottom, middle, and top of boxes indicate lower quartile, median, and upper quartile values, respectively. If notches of boxes do not overlap, medians of the two sets of data differ at the 0.95 confidence interval. Points are outliers to whiskers of 1.5 times the interquartile range. The symbol ‘+’ denotes points from 1.5 to 3 times the interquartile range, and ‘o’ represents points outside 3 times the interquartile range.

From Figure 5, it is evident that the longest time frame combination {5, 10, 15, 20} outperformed (or performed as well as) the buy-and-hold baseline for three of four stocks. For RIMM and AAPL, the longest time frame combination outperformed all others with statistical significance. For RY, it outperformed all others, but did not differ at 95% statistical significance around the median in comparison to {5, 10}. For BRK-B, the {5, 10, 15} time frame combination outperformed the others with statistical significance. The underlying price trend for BRK-B was sideways throughout the day with sporadic and short-lived price climbs and declines. The unique aspect of the {5, 10, 15} combination is that due to the majority voting rule, two out of three of the time frame GP runs must produce a buy/sell signal. That is, the time frame combination must produce a definitive buy majority to execute the buy/sell. For the other time frames, only half of the members of the time frame combination need to be used to create a majority. For the single {5} minute time frame, no voting occurs. It is likely the tendency for {5, 10, 15} to require a proper majority that leads it to stay out of the market for sporadic gains or losses of which the other time frames may try to take advantage. On a volatile sideways trend such as that for BRK-B, not trying to take advantage of these short changes was advantageous. For the other trend types of RIMM, RY, and AAPL where most of the day was sustained climb or decline, a more reactive voting mechanism was most beneficial.

Final profit at the end of the time frame combination for each stock is shown in Table 1. In terms of profits generated over the selected time period, final profit was largely depen-

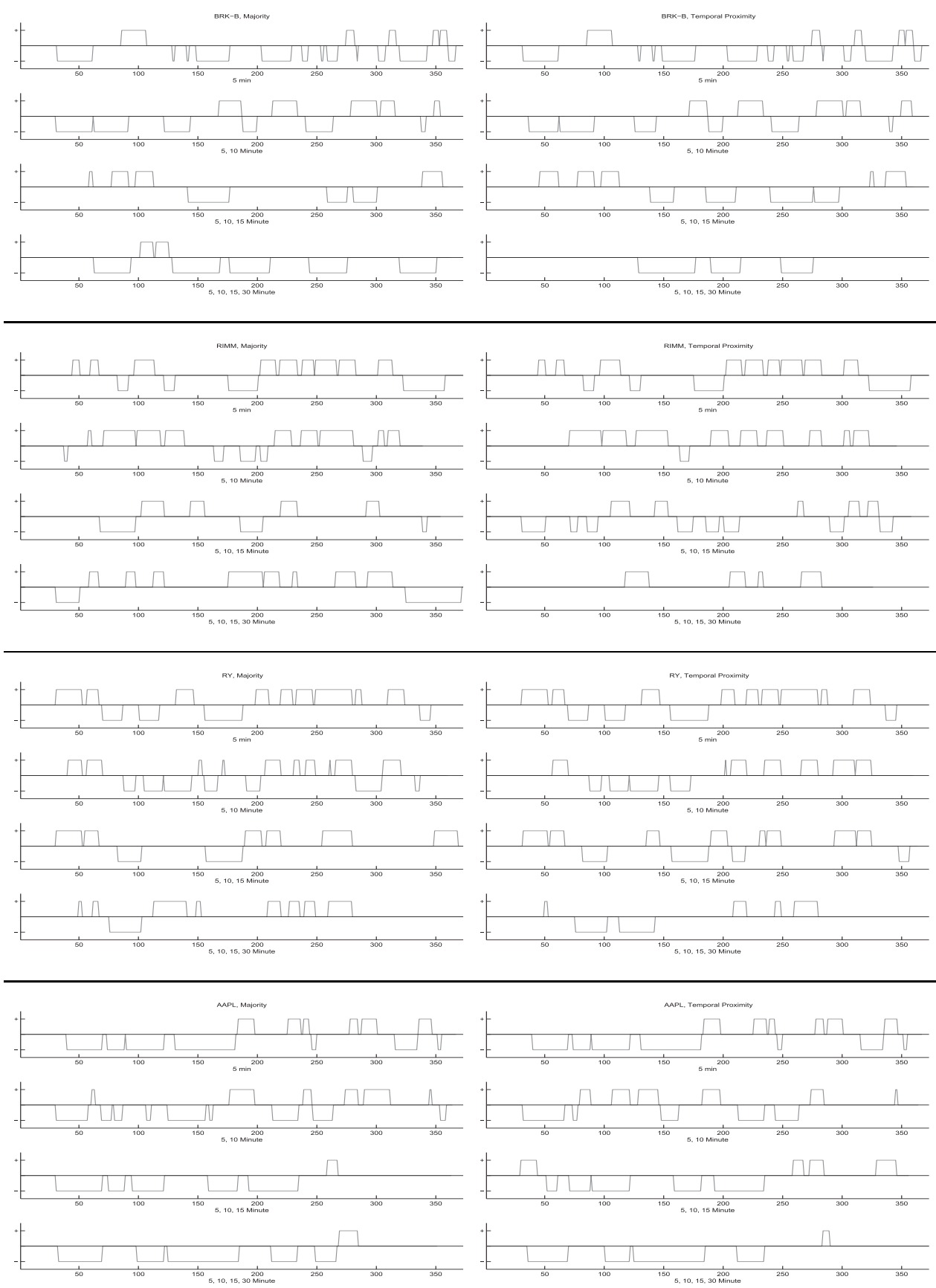


Figure 4: Time spent in and out of market for each time frame combination.

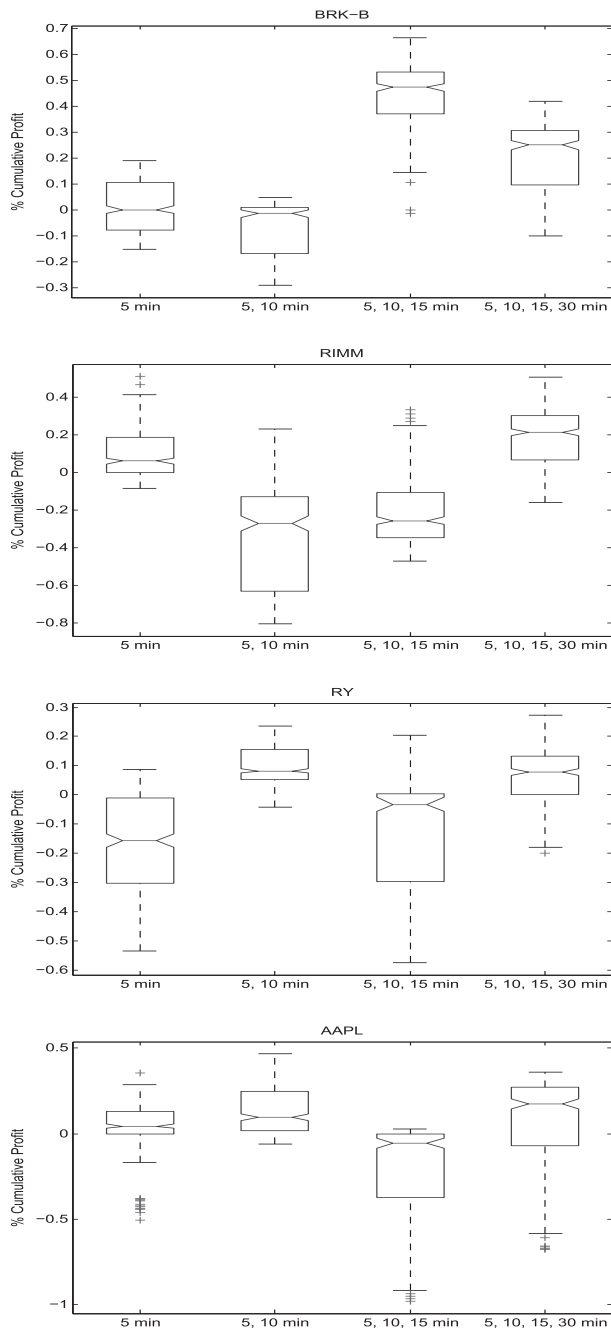


Figure 5: Cumulative profit (%) greater than buy and hold.

dent on the net gain of the stock for the day. Comparing the cumulative results to the final profits indicates that the time frames with highest final profit do not necessarily correspond to those with highest cumulative profits. It should be noted that final profit depends heavily on the arbitrary stopping point.

5. CONCLUSIONS

A linear genetic programming (LGP) system is applied to intraday stock trading using multiple time frame combinations: 5 minutes, {5, 10} minutes, {5, 10, 15} minutes, and

{5, 10, 15, 20} minutes. Two types of decision technique were used to determine whether or not a buy, sell, or hold signal would be issued by the system: majority and temporal proximity. The temporal proximity decision mechanism was more restrictive and traded slightly less often than the majority decision, and it was found not to work as well during performance analysis. Focusing on the majority decision mechanism, it was evident that time frame combinations involving more time frames (stretching into the past) were more conservative and less reactive to changes in price trend. Combinations of more time frames also traded less often and stayed in the market longer during trades. Such behavior was found to be more beneficial than shorter time frame combinations that did not stretch as far into the past for price trends that moved down or upwards overall throughout the day. However, for price trends that involved larger volatile movements, the three-member time frame worked best, likely due to a more restrictive demand (2 out of 3) to achieve majority for a trading action. Planned future work includes different means of gathering time frame data, such as sampling points at the time frame interval, and different voting mechanisms for determining majority decision.

6. REFERENCES

- [1] A. Brabazon and M. O'Neill. *Biologically Inspired Algorithms for Financial Modeling*. Springer Verlag, Berlin, 2006.
- [2] M. Brameier and W. Banzhaf. *Linear Genetic Programming*. Springer, New York, 2007.
- [3] S.-H. Chen and N. Navet. Failure of genetic-programming induced trading strategies: Distinguishing between efficient markets and inefficient algorithms. In S.-H. Chen, P. P. Wang, and T.-W. Kuo, editors, *Computational Intelligence in Economics and Finance*, pages 169–182. Springer Berlin Heidelberg, 2007.
- [4] M. A. Kaboudan. A measure of time series' predictability using genetic programming applied to stock returns. *Journal of Forecasting*, 18(5):345–357, 1999.
- [5] J. Li and E. P. K. Tsang. Reducing failures in investment recommendations using genetic programming. *Computing in Economics and Finance 2000* 332, Society for Computational Economics, 2000.
- [6] C. J. Neely and P. A. Weller. Predicting exchange rate volatility: Genetic programming vs. GARCH and RiskMetrics™. *Federal Reserve Bank of St. Louis Review*, 84(3):43–54, May/June 2002.
- [7] G. Wilson and W. Banzhaf. Fast and effective predictability filters for stock price series using linear genetic programming. In *2010 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.
- [8] G. Wilson and W. Banzhaf. Interday and intraday stock trading using probabilistic adaptive mapping developmental genetic programming and linear genetic programming. In A. Brabazon, M. O'Neill, and D. Maringer, editors, *Natural Computing in Computational Finance*, volume 293 of *Studies in Computational Intelligence*, pages 191–212. Springer Berlin / Heidelberg, 2010.