

Interday and Intraday Stock Trading Using Probabilistic Adaptive Mapping Developmental Genetic Programming and Linear Genetic Programming

Garnett Wilson and Wolfgang Banzhaf

Memorial University of Newfoundland, St. John's, NL, Canada
gwilson@cs.mun.ca, banzhaf@cs.mun.ca

Summary. A developmental co-evolutionary genetic programming approach (PAM DGP) is compared to a standard linear genetic programming (LGP) implementation for trading of stocks in the technology sector. Both interday and intraday data for these stocks were analyzed, where both implementations were found to be impressively robust to market fluctuations while reacting efficiently to opportunities for profit. PAM DGP proved slightly more reactive to market changes compared to LGP for intraday data, where the converse held true for interday data. Both implementations had very impressive accuracy in choosing both profitable buy trades and sells that prevented losses for both interday and intraday stock data. These successful trades occurred in the context of moderately active trading for interday prices and lower levels of trading for intraday prices.

11.1 Introduction

Technical analysis of the stock market involves attempts to examine the past effects of market movements in order to anticipate what traders will do next to affect the market. Such analysis involves the use of technical indicators to examine price trends and trading volume in order to identify the likely future trading activity and change in price of an asset [1]. In recent years, a number of Evolutionary Computation-inspired algorithms, including genetic programming (GP), have been applied to the analysis of financial markets with a reassuring degree of success. This paper explores the use of a developmental GP system, Probabilistic Adaptive Mapping Developmental Genetic Programming (PAM DGP), that uses co-operative co-evolution of genotype solutions and genotype-phenotype mappings, as well as Linear Genetic Programming (LGP), for both interday and intraday stock trading. While the encoding of functions is static for LGP, PAM DGP allows emphasis of particular functions over others.

The following section describes previous related approaches to stock analysis. Section 11.3 describes the LGP and PAM DGP implementations and their application to trading of individual stocks, as well as the function set for this domain and the related interpretation of an individual's genotype. Results are provided in Section 11.4 for the interday analysis of three technology sector stocks, with intraday analysis following in Section 11.5. Conclusions and future work follow in Section 11.6.

11.2 Related Approaches to Stock Prediction

Genetic programming approaches have met with considerable success when applied to stock analysis. Yan et al. have shown standard GP to outperform other machine learning techniques such as support vector machines for application to portfolio optimization in highly volatile markets, where this success is attributed to adaptation to optimize profits rather than simply predict returns [7]. Furthermore, the authors found that GP was superior in its balance of Return On Investment (ROI) and robustness to volatility. LGP has been applied to market analysis previously by Grosnan et al. [4], where Nasdaq and Nifty indices were examined. Multi-expression programming (MEP), LGP, and an MEP/LGP ensemble were found to surpass the predictive performance of the particular neural network or neuro-fuzzy implementations chosen by the authors for next day prediction of stock prices. The PAM DGP algorithm that is used in this study relies on a co-evolutionary mechanism. A co-evolutionary process has also been applied to the creation of trading rules by Drezewski and Sepielak [3] where one species represented market entry strategies and one species represented exit strategies. In addition, a multi-agent version of the co-evolutionary algorithm and evolutionary algorithm were tried. For the particular data set used by the authors, the multi-agent co-evolutionary approach generated the most profit. To the authors' knowledge, developmental GP had not been applied to stock market analysis until the original study by the authors on which this chapter is partially based [6].

In terms of the application of the GP algorithm to interday trading rule generation, a technique somewhat similar to the grammatical evolution (GE) approach of Brabazon and O'Neill [1] was adopted: After a period of initial training, the best evolved rules in the population were used to trade live for a window of n days. The window is then shifted ahead and the current population is retrained on the data within the window on which it was previously trading live in order to trade live on the following n days, and so on. The authors compare two versions of the GE system, one that maintains its population across window-based training periods and one that re-initializes the population with each window shift / training period. The authors found that maintaining the populations, rather than re-initializing them with each window, provided better trading rules that yielded greater profits. As detailed in the following section, our technique uses a shifting window of length 5 (increments of one day for interday analysis, minute interval ticks for intraday), but shifts only in increments of 1 day/tick. Following the findings and recommendations of [1], populations are not re-evolved with the shifting of each window.

11.3 Stock Analysis Using Developmental and Linear GP

Genetic programming is one of a family of algorithms in machine learning classified as evolutionary methods. In such methods, a population of candidate solutions (called "individuals") are ranked in their ability to perform the objective of solving an optimization problem based on some measure of error or success, called a "fitness" function. The individuals in the population are ranked using the fitness function, and the fittest individuals are biased for selection as parents and used to create a new population of

solutions. The genetic material composing the parents is then manipulated, after being copied, to create children who typically replace some individuals originally in the population. Population size thus does not typically change. The manipulation of genetic material is accomplished through the use of the two operators of crossover and mutation. Crossover allows swapping of genetic material between two individuals, while mutation causes the creation of new genetic material by altering the genotype that is already present.

11.3.1 Description of PAM DGP and LGP Algorithms

This work uses a well-established variant of genetic programming called “linear genetic programming” or “LGP” [2]. In this variant, the genetic material of the individuals (called “genotype”) has the form of a linear list of instructions. Program execution is that of a simple register machine, and instructions are made up of opcodes and operands. The opcodes correspond to functions in a functional set, with arguments to the functions being considered a terminal set. As the program executes, it alters the contents of internal registers and perhaps a separate solution register. The structure of a linear GP individual is depicted below in Figure 11.1. When the bit strings are interpreted, they correspond to members of the functional sets to produce a solution that makes semantic sense in terms of the original problem, also called the “phenotype.” For instance, the binary sequence “011” in the individual’s genotype could be interpreted as the functional set member “addition” in the phenotype. All the instructions in the phenotype are then evaluated to determine its fitness.

A modern trend in genetic programming, called “developmental genetic programming (DGP)” has been to develop models that more closely mirror the higher-level developmental processes of nature. One type of developmental GP is accomplished by using a genotype-phenotype mapping as an intermediary between an individual’s genotype and phenotype. Specifically, a genotype-phenotype mapping is the encoding of a phenotypic symbol by one or more codons, where a codon is a non-zero contiguous bit sequence from a binary genotype. The biological analogue of the evolution of a genotype-phenotype mapping in a developmental system, as is described in this work,

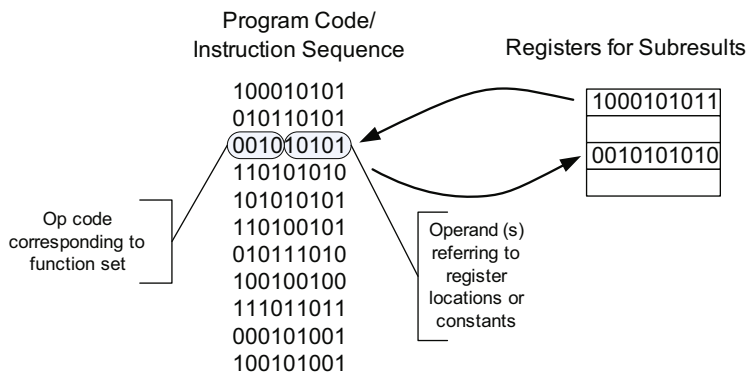


Fig. 11.1. Linear Genetic Programming (LGP) genotype

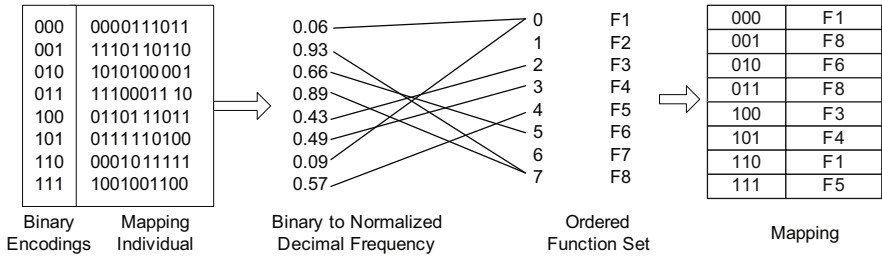


Fig. 11.2. PAM DGP mapping process

is evolution of individuals and a shared “genetic code” that maps codons to amino acids in actual biological organisms.

The developmental GP variant used in this work for stock analysis is Probabilistic Adaptive Mapping Developmental Genetic Programming (PAM DGP), introduced in [5]. Genotypes in PAM DGP are binary strings, with interpretation of sections of the binary string being instruction-dependent (see next Section 11.4). Mappings in this work are redundant such that individuals are composed of $b \geq s$ 10-bit binary strings, where b is the minimum number of binary sequences required to represent a function set of s symbols. Each 10 bit mapping section is interpreted as its decimal equivalent, normalized to the range $[0, 1]$, and mapped to an ordered function set index by multiplying by $s - 1$ and truncating to an integer value to yield an index value from 0 to $s - 1$ (allowing redundant encoding of symbols). For example, in the mapping expressed in Figure 11.2, the first 10-bit binary sequence of the mapping individual “0000111011” corresponds to an encoding for the genotype binary sequence “000.” By dividing the decimal equivalent of the mapping individual’s 10-bit sequence (59) by the maximum value (“0000000000”, 1023 in decimal), a normalized value of 0.06 results. Truncating 0.06 to the integer value 0 means that the genotype binary sequence (op code) corresponds to the member of the ordered function set at index position 0, which is function (F1). Thus, genotype binary sequence “000” corresponds to the function set member F1.

Using this mapping mechanism with co-evolutionary selection, PAM DGP will emphasize the most useful members of the function set, ignore members of the function set which are not pertinent, and simultaneously evolve an appropriate genotype solution. PAM DGP is compared to the standard LGP implementation as described by Brameier and Banzhaf in [2]. LGP individuals are also bit strings, and there is naturally only a genotype population. The interpretation of instructions in PAM DGP can be considered the same for LGP, only LGP uses a static mapping and constant function set. Thus, PAM DGP extends LGP such that members of a function set are adaptively emphasized.

In PAM DGP there is a population of genotypes that cooperatively coevolves with a separate population of mappings. A probability table is updated throughout algorithm execution with entries corresponding to each pair of individual genotype and mapping from both populations. The table entries represent frequencies that dictate the probability that roulette selection in a steady state tournament will choose the genotype-phenotype pairing of individuals determined by the indices of the table. The genotype

and mapping individual that are members of the current best genotype-mapping pairing are immune to mutation and crossover to maintain the current best solution discovered. Each tournament round involves the selection of four unique genotype-mapping pairings. Following fitness evaluation and ranking, the probability table columns associated with the winning combinations have the winning combination in that column updated using Equation 11.1 and the remaining combinations in that column updated using Equation 11.2

$$P(g, m)_{new} = P(g, m)_{old} + \alpha(1 - P(g, m)_{old}) \tag{11.1}$$

$$P(g, m)_{new} = P(g, m)_{old} - \alpha(P(g, m)_{old}) \tag{11.2}$$

where g is the genotype individual / index, m is the mapping individual / index, α is the learning rate (corresponding to how much emphasis is placed on current values versus previous search), and $P(g, m)$ is the probability in table element $[g, m]$. To prevent premature convergence, the algorithm uses a noise threshold. If an element in the table exceeds the noise threshold following a tournament round, a standard Gaussian probability in the interval $[0, 1]$ is placed in that element and all values in its column are re-normalized so the column elements sum to unity. The PAM DGP algorithm and selection mechanism are summarized in Figure 11.3. Additional details of PAM DGP and its improvements over other, related systems are described in [5].

Each steady state tournament consists of 1000 rounds (4 individuals per round). PAM DGP uses a genotype population of size 10 (as does LGP) and mapping population of size 10. Each genotype consists of 320 bits and 4 subresult registers, and each mapping consists of 160 bits (10 bits for each of 16 required encodings for a function set of size 16). XOR mutation on a (uniform) randomly chosen instruction was used on genotypes, with low threshold point mutation used on mappings to provide a more stable context against which the genotype could evolve. The genotype population used a mutation rate of 0.5 and a crossover rate of 0.9. The mapping population uses a lower crossover and mutation rate, both set at 0.1. PAM DGP used a conservative learning rate of 0.1 and noise threshold of 0.95 to prevent premature convergence.

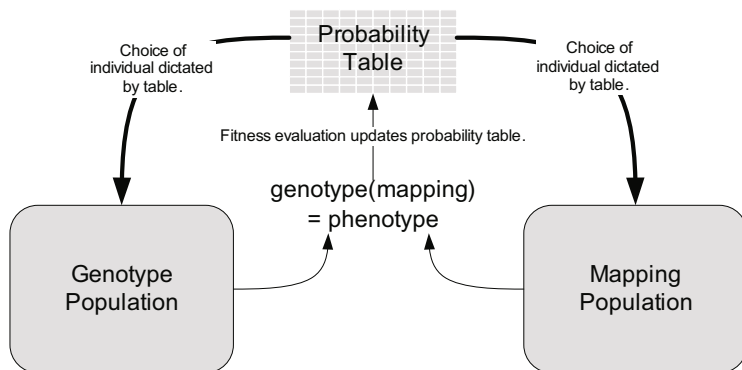


Fig. 11.3. Probabilistic Adaptive Mapping Developmental Genetic Programming (PAM DGP)

11.3.2 Stock Analysis with PAM DGP and LGP Algorithms

The PAM DGP and LGP implementations are applied to three stocks in the technology sector: Google Inc., ticker symbol “NASDAQ:GOOG,” Apple Inc., ticker symbol “NASDAQ:AAP,” and Microsoft Corporation, ticker symbol “NASDAQ:MSFT.” The initial exchange portion of the ticker symbols will be removed for brevity in the remainder of the paper. High, low, open, and close data was provided as input for the 200 day period in 2007 for interday data analysis, which we use here to explain the operation of the algorithm. (Particular differences in implementation for intraday data are discussed in Section 11.5.) The first 16 days of the 200 day intraday data set were reserved as a basis on which to draw technical indicator data. After the first 16 days, the GP fitness was evaluated on data corresponding to a moving window of 5 days. Individuals represent sets of trading rules, based on functions in the function set (to be described). For each window of 5 days corresponding to trading days m to n , each of m to $n - 1$ days were used for calculation of a trading decision given the individual’s rule set, with $m + 1$ to n being used to evaluate the recommendation based on the immediately preceding day. Days used for the calculation of a trading decision were normalized using two-phase preprocessing similar to [1]: All daily values were transformed by division by a lagged moving average, and then normalized using linear scaling into the range $[0, 1]$ using

$$v_{scaled} = \frac{v_t - l_n}{h_n - l_n} \quad (11.3)$$

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

In addition to an instruction set, each individual consists of a set of four registers, a flag for storing the current value of logical operations, and a separate output (trade) register for storing a final value corresponding to a trade recommendation. Following the execution of the trading rules of a GP individual, if the value of the trade register is 0, no action is recommended. Otherwise, the final value in the trade register corresponds to a value in the range $[0, 1]$. This value was multiplied by a maximum dollar amount to be bought or sold per trade (\$10,000 was used here based on an initial account balance of \$100,000 with which to trade) to give some portion of \$10,000 to be traded. For each trade conducted, there is a \$10 commission penalty. The trading system is permitted to run a small deficit \geq \$10 to either handle a sell recommendation when maximally invested (where the deficit would be immediately recouped) or, similarly, to allow a buy in order to be maximally invested. Fitness of an individual is the value of the cash and shares held.

The best individual consisting of the best trading rule set is used by a “live” trading algorithm. That is, the live trader provides known information to the GP for days m to n . The GP algorithm returns a recommendation on which the live trading system bases its decision to trade on the following day, $n + 1$. In particular, the net number of shares bought and sold by the best evolved individual (trading rules) given the recommendation of the trade register over all the fitness cases (4 cases given a 5 day window) is the buy or sell recommendation to the “live” trading system. The best GP individual can thus

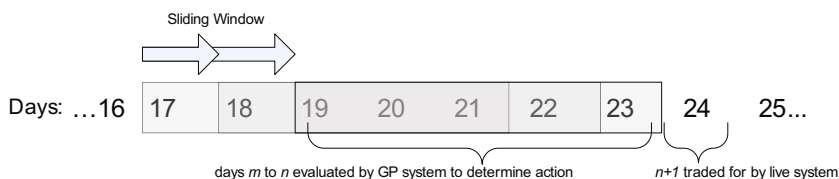


Fig. 11.4. Relationship between “live” trading system and GP tournament execution

recommend up to \$40,000 worth of share selling or buying per actual trading day to the live system. With the next window shift, the current cash and shares of stock held by the “live” trading system are the new initial amounts for the GP individuals in the next tournament on the new window content. The transactions of the live trading system are what are actually based on unknown data, and determine the success of the algorithms. The process is summarized in Figure 11.4.

While PAM DGP uses co-evolution to refine function set composition, the appropriate initial function set members must be provided as a basis upon which the algorithm can select its optimum function set. In the case of standard GP, this initial function set remains constant throughout execution. The function set includes standard mathematical operators (+, -, *) and instructions to trade based on logical operators (<, >, =) applied to the four internal registers. In addition, there are four established financial analysis metrics of moving average, momentum, channel breakout, and current day high, low, open, or close price. The financial technical indicator moving average is the mean of the previous n share prices. The momentum indicator provides the rate of change indicator, and is the ratio of a particular time-lagged price to the current price. Momentum is used to measure the strength of the trend of a stock price, and is often used to predict price peaks [1]. Channel breakout establishes a trading range for a stock, and reflects its volatility. The most popular solution places Bollinger bands around a n -day moving average of the price at ± 2 standard deviations of the price movement over the last n days. A trader is typically alerted when the stock price passes the upper or lower bound of the Bollinger bands.

11.4 Interday Trading Results

The worth of the assets held by the live trading system for each of 184 days of trading is initially analyzed (200 fitness cases were used overall, with the initial 16 being reserved so initial technical financial indicators had values). Fifty such trials over 184 days of trading were conducted for each of the four stocks using an Apple iMac Intel Core 2 Duo 2.8 GHz CPU and 4GB RAM using OS X Leopard v10.5.4. Starting trading with \$100,000, the mean worth (with standard error) of the live trading system for PAM DGP, LGP, and naïve buy-and-hold strategies is provided in Figures 11.5 to 11.7.

Given Figures 11.5 to 11.7, the prevalent observation is that PAM DGP and LGP are both impressively robust to share price fluctuations (as indicated by the buy-and-hold trend line). The evolved solutions seem to take advantage of the upward trends, although the solutions reflect a conservative strategy overall, adept at anticipating and buffering against sharp share price declines and volatility in general. In the instance

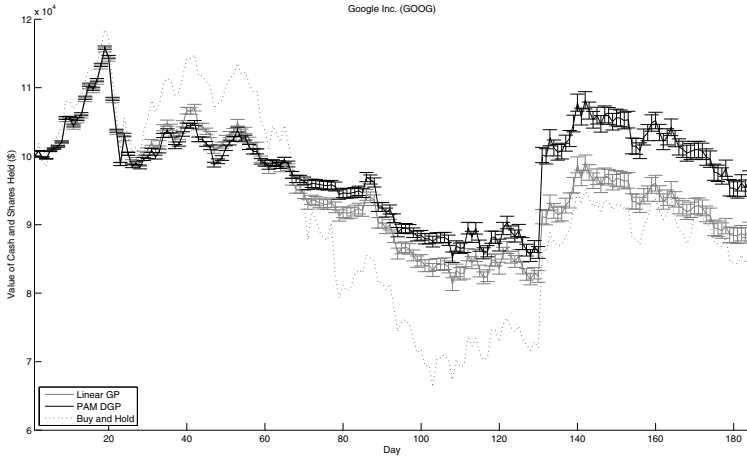


Fig. 11.5. Mean total worth for GOOG interday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies over 50 trials with standard error given initial \$100,000 cash value

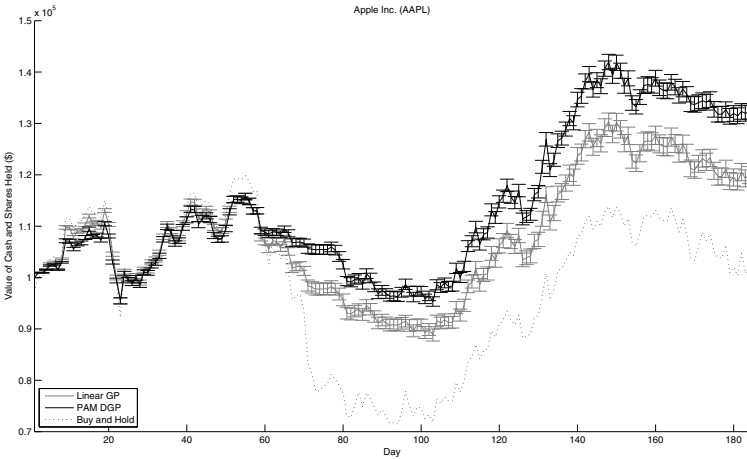


Fig. 11.6. Mean total worth for AAPL interday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies over 50 trials with standard error given initial \$100,000 cash value

of MSFT, there are no declines sharp enough or for long enough duration to cause significant withdrawal of investments. The MSFT example shows that a moderately volatile, gradual downward trend will have the algorithm gradually lose money as there is no consistent period of gain or loss. However, in this degenerate case buy-and-hold results in a similar loss of money—that is, one does not lose much even given such a deceptive scenario. Both algorithms achieve final profits better than buy-and-hold for

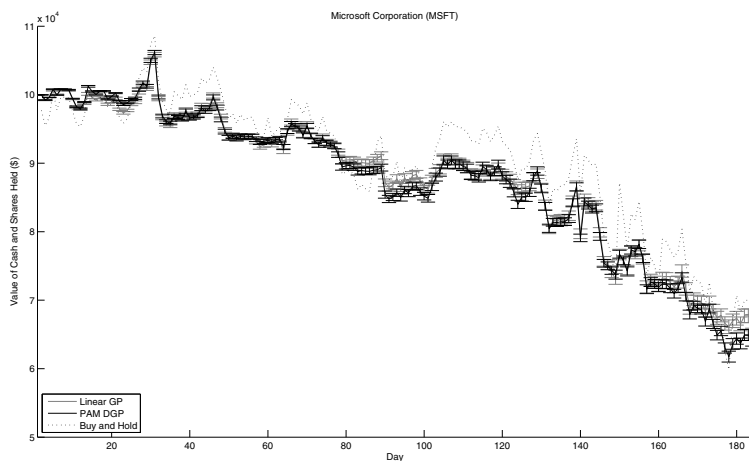


Fig. 11.7. Mean total worth for MSFT interday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies over 50 trials with standard error given initial \$100,000 cash value

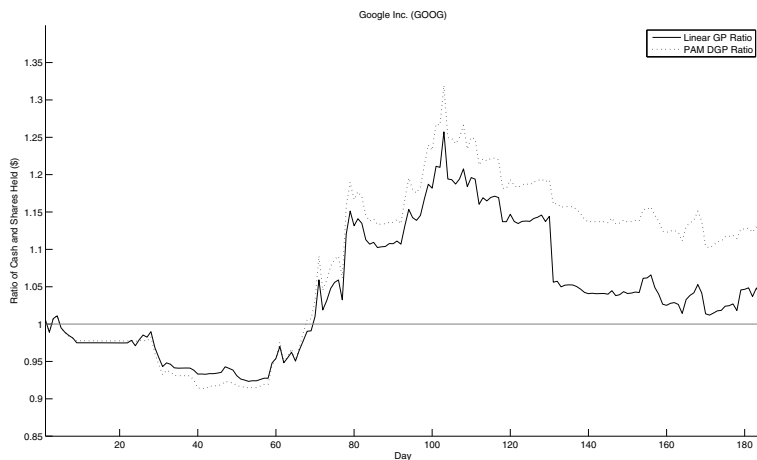


Fig. 11.8. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold over 50 trials for GOOG interday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 *vice versa*

the remaining two stocks (GOOG and AAPL). Figures 11.8 to 11.10 provide a ratio of PAM DGP and LGP to buy-and-hold total worth for a finer comparison, with profit (final and cumulative measures) shown in Figure 11.11.

Comparing the ratio of PAM DGP and LGP worth across stocks in Figures 11.5 to 11.8, PAM DGP maintains higher worth than LGP for the large majority of trading days in the instances of GOOG and AAPL, with LGP dominating PAM DGP almost

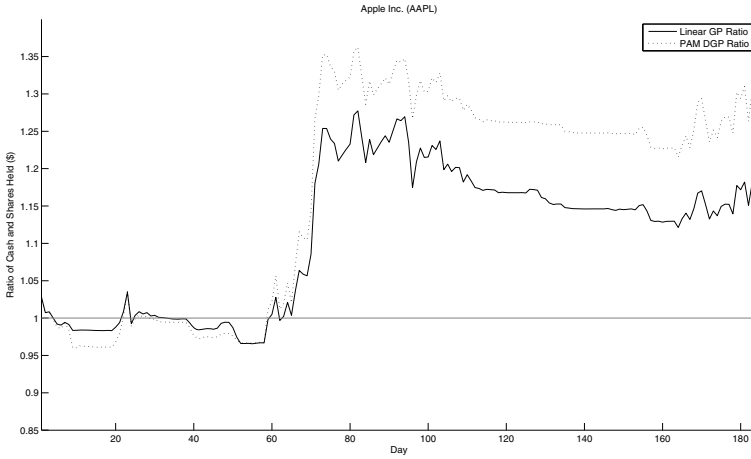


Fig. 11.9. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold over 50 trials for AAPL interday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 *vice versa*

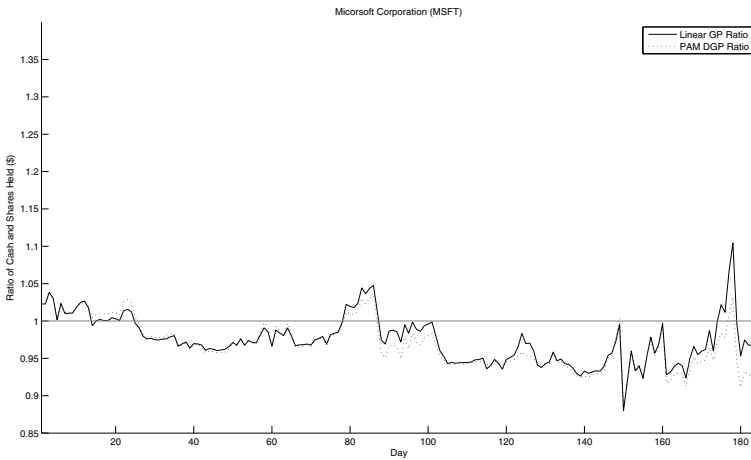


Fig. 11.10. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold over 50 trials for MSFT interday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 *vice versa*

the entire period for MSFT (but not by a significant margin). PAM DGP outperforms buy-and-hold by almost 35% at times for GOOG and AAPL, with LGP outperforming buy-and-hold by over 25% at times on those stocks. LGP and PAM DGP perform very closely throughout MSFT (Figure 11.10). Both PAM DGP and LGP outperform buy-and-hold for the majority of the time period for GOOG and AAPL. When buy-and-hold outperforms the GP algorithms throughout MSFT it is by a lower margin (typically

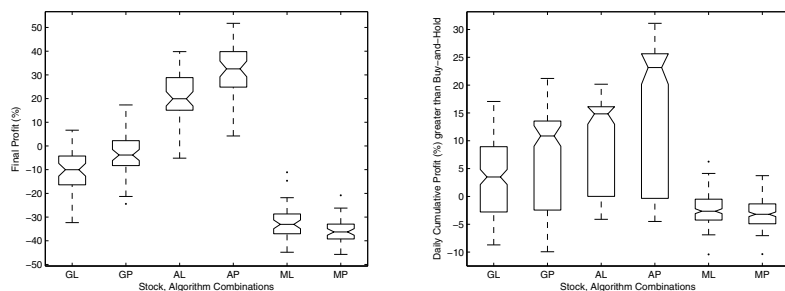


Fig. 11.11. Boxplot of mean final profit (%) and mean daily cumulative profit (%) greater than buy-and-hold for PAM DGP and LGP over 50 trials for interday prices. First letter of label indicates stock, second letter indicates algorithm. Value of 0 indicates the break even point

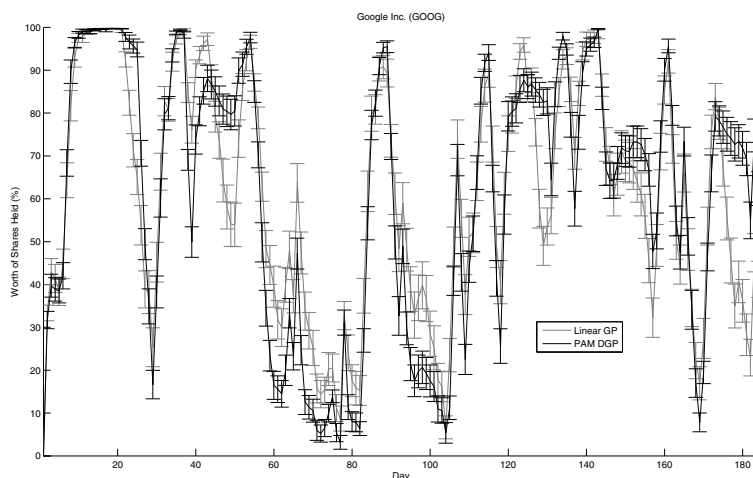


Fig. 11.12. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for GOOG interday prices as a percentage of total worth over 50 trials with standard error

5% or less). Comparing Figures 11.5 to 11.7 and 11.8 to 11.10, respectively, it is evident that PAM DGP provides increased robustness to market downturns and quickly capitalizes growth opportunities later in evolution.

In the boxplots of Figure 11.11, each box indicates the lower quartile, median, and upper quartile values. If the notches of two boxes do not overlap, the medians of the two groups differ at the 0.95 confidence interval. Points represent outliers to whiskers of 1.5 times the interquartile range. PAM DGP outperforms LGP at the end of the time period (Figure 11.11, left) for GOOG and AAPL, with no statistically significant difference in final profits for MSFT (all at the 95% confidence interval). Figure 11.11 (left) shows impressive final profit for AAPL, where the algorithm took advantage of market gains and losses (Figure 11.6). There was a general loss for both PAM DGP and

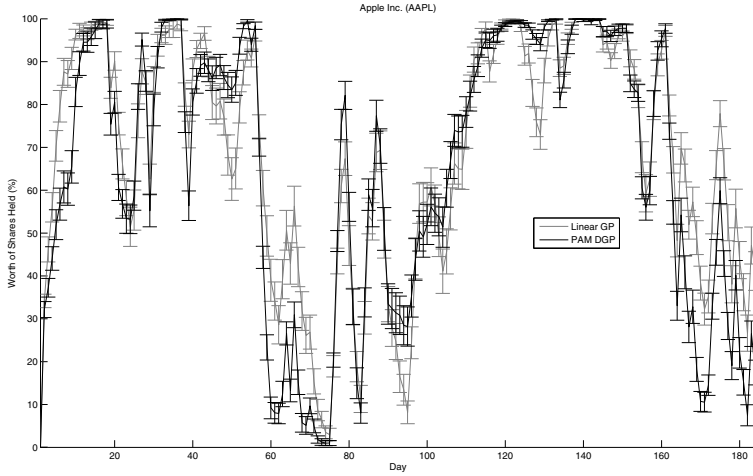


Fig. 11.13. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for AAPL interday prices as a percentage of total worth over 50 trials with standard error

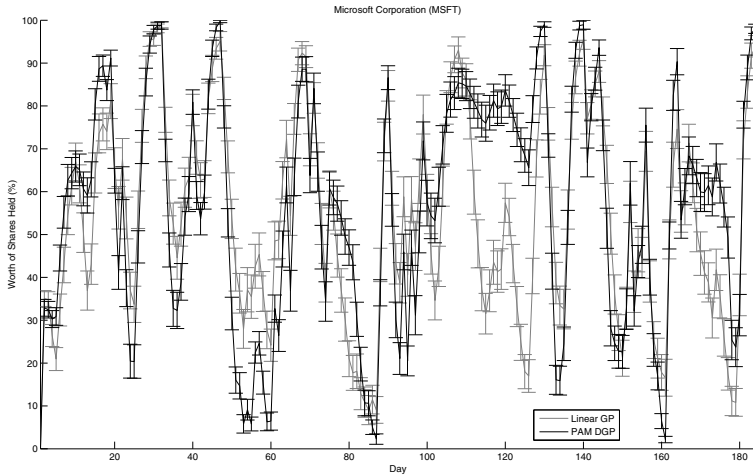


Fig. 11.14. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for MSFT interday prices as a percentage of total worth over 50 trials with standard error

LGP considering final profit for GOOG. GOOG incurred losses during most of the time period and was thus not profitable overall. Note that time period end is arbitrary and profits are a direct reflection of underlying market trend. Figure 11.11 (right) shows the mean daily cumulative profit (%) greater than buy-and-hold for the LGP and PAM DGP live trading systems over all trading days. Figure 11.11 (right) indicates that both PAM DGP and LGP were generally more profitable than buy-and-hold at any given time for GOOG and AAPL, but not for the degenerate case of MSFT where all GP algorithms

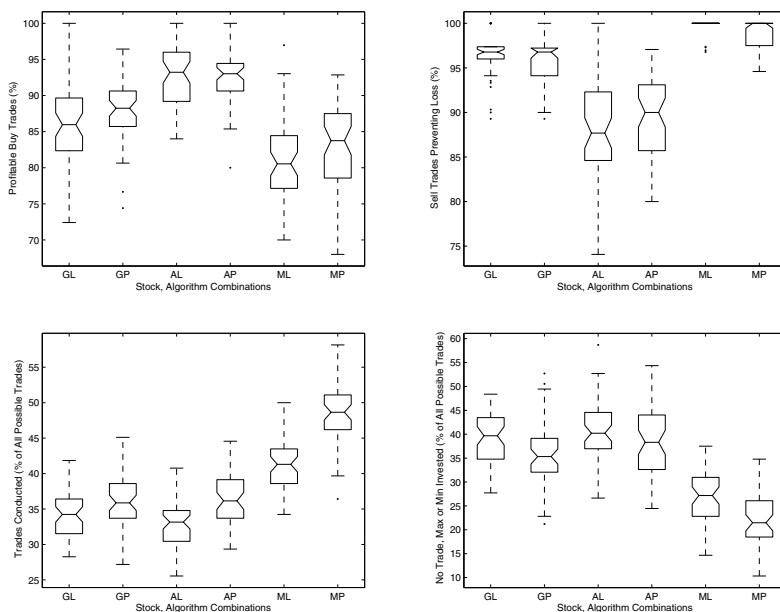


Fig. 11.15. Percentage of profitable buy trades, sell trades preventing losses, percentage of trades executed overall for each stock, and percentage of trades not conducted while maximally or minimally invested for each algorithm combination over 50 trials for interday prices. First letter of label indicates stock, second letter indicates algorithm

and naïve buy-and-sell incur similar losses. PAM DGP was more profitable than LGP at any given time by a large margin for GOOG and AAPL, and there was no statistically significant difference (at the 95% confidence interval) between LGP and PAM DGP for MSFT. Number of shares retained daily as a percentage of live trading total worth is shown in Figures 11.12 to 11.14.

Comparing Figures 11.5 to 11.7 and 11.12 to 11.14, it is evident that both PAM DGP and LGP are impressively reactive in that they will sell stock if a market downturn starts and buy when the market appears to be experiencing gains. Figures 11.5 to 11.6 and 11.12 to 11.13 also indicate that both algorithms are effective at staying maximally invested during profitable periods. In the instance of MSFT (Figure 11.7), however, where frequent gains and losses of non-substantial amounts occur, Figure 11.14 indicates that the GP never fully invests or is out of the market. Indeed, given the trend of Figure 11.7, it is not prudent to fully invest or sell all shares at any particular point. Proportion of profitable trades is a common metric for evaluation of trading activity, although it is deceptive: it does not even reflect the overall ability of an algorithm in terms of actual profit generated [1]. Many trades, although not profitable, are beneficial in preventing loss during market downturns. Thus, rather than percentage of profitable trades, the percentage of profitable buy trades and percentage of sell trades preventing loss for each algorithm are shown in the top left and right boxplot of Figure 11.15, respectively. A profitable buy is defined as a buy where the total value of shares and cash held at a

time prior to the next sell exceeds the total value at the time of purchase (less transaction cost). Similarly, a sell preventing further losses is defined as a sell where the total value of shares and cash held at a time prior to the next buy is less than the total value at the time of sale (less transaction cost). The percentage of trading opportunities where action was taken is shown in Figure 11.15 (bottom left). Out of all possible trades, the number of trades not conducted when the system was maximally or minimally invested is shown in Figure 11.15 (bottom right).

Figure 11.15 reveals that both algorithms are extremely accurate at buying to gain profit, with medians of approximately 80% - 95% profitable buys. There is no statistical difference (at the 95% confidence interval) in the ability of PAM DGP or LGP to buy for profit for any of the stocks examined. In terms of protecting investment through selling to prevent loss, the median for PAM DGP and LGP was very good (typically 95% to 100%) for GOOG and AAPL. For MSFT, the percentage of sells preventing loss was lower (87 to 90%), where the trend in price was volatile with few sudden, extreme changes (Figure 11.7). LGP had better, or no statistical difference in, performance compared to PAM DGP when selling to prevent losses. Any outliers in either buying for profit or selling to prevent loss were acceptably high percentages. These beneficial transactions are also the result of trading levels with medians of 30% to 40% of possible trades for GOOG and AAPL, with higher medians for MSFT (more trades were conducted due to the volatility of MSFT). PAM DGP generally conducted more trades (based on spread of data) than LGP for all stocks. Figure 11.15 (right, bottom) indicates medians of approximately 35-40% of trades where the system wished to maintain a maximally or minimally invested position for GOOG and AAPL. Compared with Figures 11.12 and 11.13, it is evident that most of these positions were maximal investment to generate profit. Neither algorithm maximally or minimally invested for a high percentage of trades for MSFT in Figure 11.15 (right, bottom), also seen previously in Figure 11.14, due to the volatility of MSFT. Overall, Figure 11.15 indicates that the percentage of beneficial trades that were made to generate profit or protect from losses were impressively high, where this occurred in the context of moderate levels of trading.

11.5 Intraday Trading Results

Both PAM DGP and LGP systems were applied to intraday data on the same three technology stocks (GOOG, AAPL, and MSFT) in a different time period. The worth of the assets held by the live trading system for each of 9 consecutive days of trading is analyzed. For each of nine days, 370 stock ticks from the beginning of the trading day are used where the value of the stock is requested from a server every minute (giving 3330 ticks total). As in [1], we opted not to trade during the final minutes of the trading day due to extreme values (daily high/low points) and high volatility in early and late trading. The last 20 minutes of trading each day are simply not used, as there are 390 minutes of trading on the NASDAQ (where all three stock prices are traded) exchange per day. To avoid early trading but not lose too much trading time, the first 8 minutes of trading are used to generate preprocessed values and then seed technical indicators. It is also worth noting that the data was taken from a period during which the markets were not doing well, namely September to October 2008, so the challenge was to trade for

Table 11.1. Intraday stock trading final profit (%)

	GOOG	AAPL	MSFT
PAM DGP	-11.52	-14.76	-7.84
LGP	7.07	-16.64	-3.97

profit on generally declining stocks. After each trading day, all shares were cashed out; thus no overnight positions were held (a common practice for many intraday traders to avoid the pitfalls of volatility in late trading).

At the start of the next trading day, the cash value left after the previous trading day was used to invest. The nature of intraday data trends compared to interday trends also necessitated some changes to the parametrization of the algorithm, since the intraday data was much more volatile. Thus, a smaller window for technical indicators was adopted (5 ticks), but the size of the moving window used by the GP was kept the same (also 5 ticks). Since there were considerably more opportunities to trade during each day, with increased volatility of prices, the maximum trade allowed was reduced to \$1000. An Apple iMac Intel Core 2 Duo 2.8 GHz CPU and 4GB RAM using OS X Leopard v10.5.4 was again used to conduct intraday trading, as for interday data. In real time, the analysis took approximately 4 hours to analyze 3330 ticks for one stock, giving a speed of 4.32 seconds per tick. Starting trading with \$100,000, the mean worth (with standard error) of the live trading system for PAM DGP, LGP, and naive buy-and-hold strategies is given in Figures 11.16 to 11.18. Ratio of PAM DGP and LGP performance to buy-and-hold is shown in Figures 11.19 to 11.21. Final profit is shown in Table 11.1, with cumulative profit (%) greater than buy-and-hold in Figure 11.22.

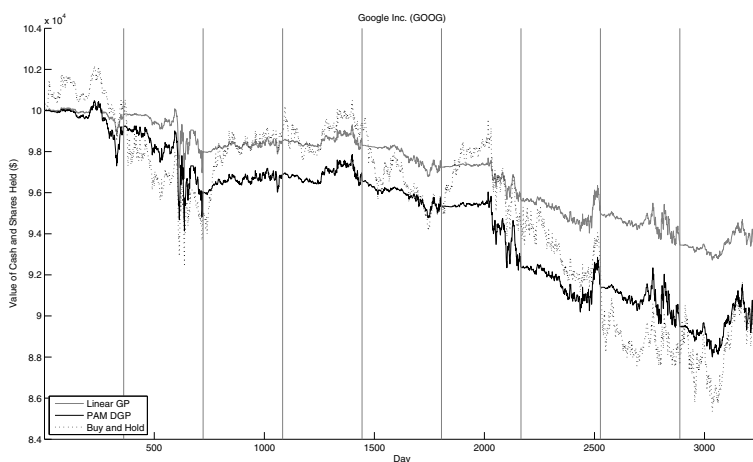


Fig. 11.16. Mean total worth for GOOG intraday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies given initial \$100,000 cash value. Vertical lines separate trading days

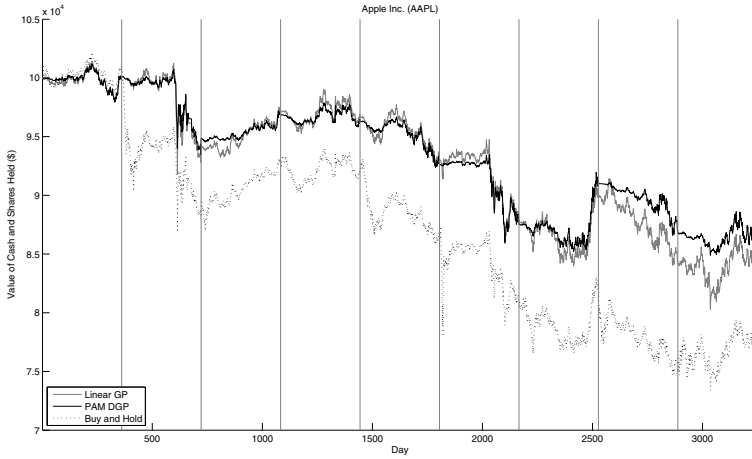


Fig. 11.17. Mean total worth for AAPL intraday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies given initial \$100,000 cash value. Vertical lines separate trading days

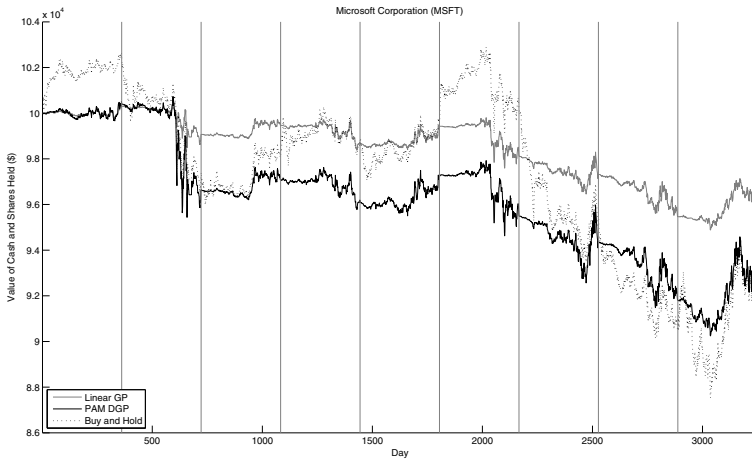


Fig. 11.18. Mean total worth for MSFT intraday prices (value of cash and shares) of PAM DGP, LGP, and buy-and-hold strategies given initial \$100,000 cash value. Vertical lines separate trading days

Figures 11.16 to 11.18 show Linear GP outperforming PAM DGP and buy-and-hold for most portions of the time period for GOOG (Figure 11.16) and MSFT (Figure 11.18). Moreover, for both GOOG and MSFT, LGP outperforms buy-and-hold for much of the time period, whereas PAM DGP does not. LGP also seems to be better able to sell to prevent loss during market downturns for GOOG and MSFT (Figure 11.16 and 11.18). For AAPL (Figure 11.17), however, the performance of LGP and PAM DGP are closely

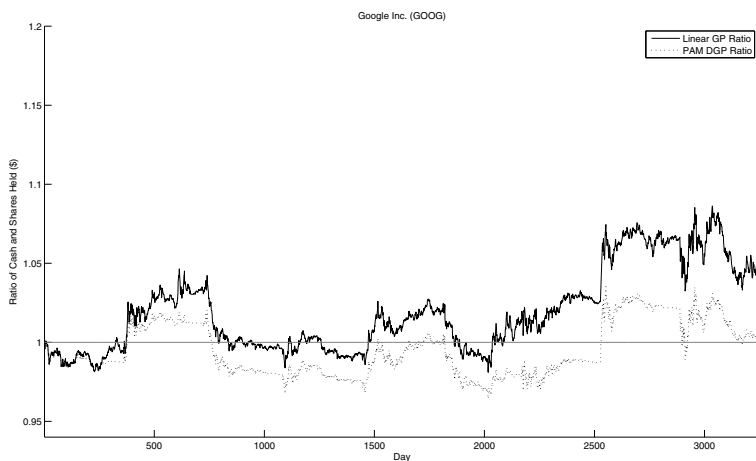


Fig. 11.19. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold for GOOG intraday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 vice versa

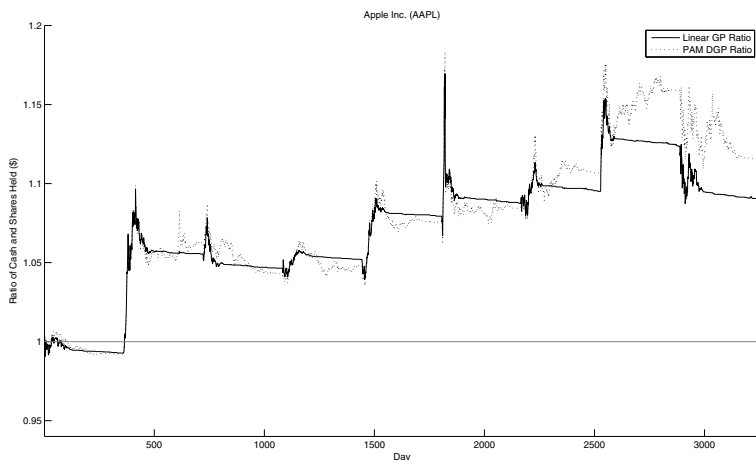


Fig. 11.20. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold for AAPL intraday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 vice versa

matched, with PAM DGP outperforming LGP (albeit by a small margin) during a significant portion of the time period. The intraday trend of AAPL differs from GOOG and MSFT in that it presents a downward slope with less severe downturns and climbs, with the exception of start of trading on the second day. The intraday AAPL trend is similar to, but slightly more volatile than, MSFT interday data in Figure 11.7 where LGP and PAM DGP performed closely throughout the time period. Figures 11.19 to 11.21 echo

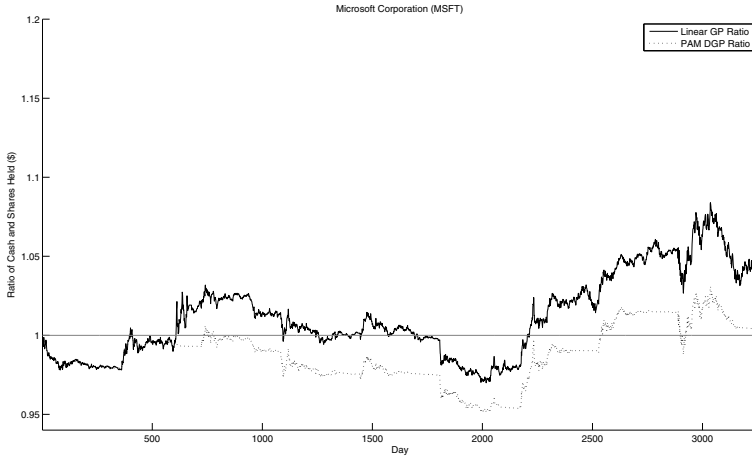


Fig. 11.21. Mean ratio of PAM DGP and LGP live trading system total worth to buy-and-hold for MSFT intraday prices. Values greater than 1 indicate greater GP worth than buy-and-hold, values less than 1 vice versa

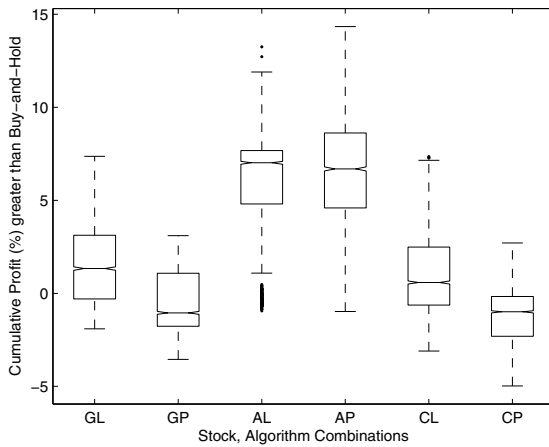


Fig. 11.22. Intraday cumulative profit (%) greater than buy-and-hold for PAM DGP and LGP over 50 trials. First letter of label indicates stock, second letter indicates algorithm. Value of 0 indicates the break even point

the observations of Figures 11.16 to 11.18: For GOOG and MSFT (Figures 11.19 and 11.21), LGP outperforms buy-and-hold the majority of the time whereas PAM DGP lags behind simple buy-and-hold. In the case of AAPL, Figure 11.20, LGP and PAM DGP perform closely throughout the data set. However, both more significantly outperform buy-and-hold than was the case for either GOOG or MSFT.

Table 11.1 reflects that trading occurred during a difficult time for the markets. That is, no final profit was made due to the general downward trend of all stocks. Greater

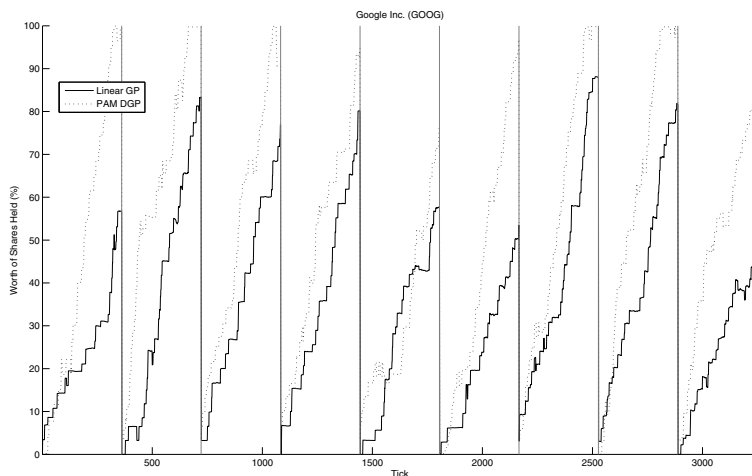


Fig. 11.23. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for GOOG intraday prices as a percentage of total worth

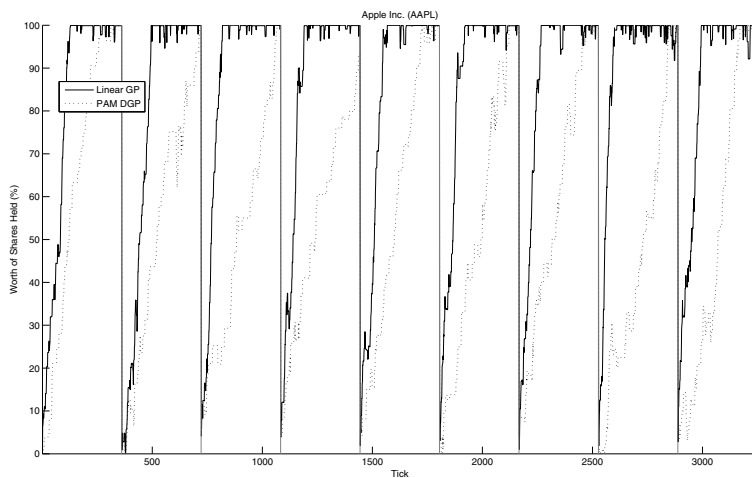


Fig. 11.24. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for AAPL intraday prices as a percentage of total worth

losses were incurred by PAM DGP than for LGP for GOOG and MSFT, but AAPL incurred greater (but similar) losses using LGP as when using PAM DGP. Considering cumulative profit over buy-and-hold (Figure 11.22) for all intraday data, LGP outperformed buy-and-hold for every stock. LGP outperformed PAM DGP for GOOG and MSFT, but performed on par with PAM DGP for AAPL (as seen also in Figures 11.16 to 11.18). The shares held by the algorithm as a percentage of total worth is shown in Figures 11.23 to 11.25.

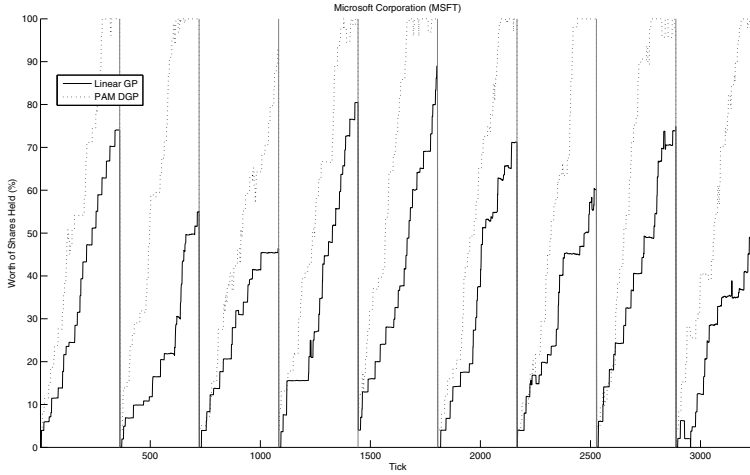


Fig. 11.25. Mean shares held by PAM DGP (black) and LGP (grey) live trading systems for MSFT intraday prices as a percentage of total worth

Figures 11.23 and 11.25 show that for GOOG and MSFT, respectively, on any given trading day LGP will not maximally invest. Note that at the end of each trading day, the shares are all sold so no overnight positions are held: the abrupt selling of all shares is not a result of GP solutions. However, PAM DGP will maximally invest. Given the overall performance of both algorithms compared to buy-and-hold (see Figure 11.22), the LGP tendency not to maximally invest yields better performance. In contrast, for AAPL in Figure 11.24, LGP shows a tendency to maximally invest prior to PAM DGP taking a similar market position. In this instance, though, neither strategy significantly

Table 11.2. Profitable buy trades (% of all buys), protective sell trades that prevented loss (% of all sells), trades conducted (% of all possible trades), and no trade with maximum or minimum invested (% of all possible trades)

		GOOG	AAPL	MSFT
Profitable Buys(%)	PAM DGP	95.64	96.63	98.10
	LGP	97.12	97.76	96.19
Protective Sells(%)	PAM DGP	100.00	98.46	98.36
	LGP	100.00	100.00	100.00
Conducted Trades(%)	PAM DGP	12.22	15.88	11.60
	LGP	8.09	20.62	7.02
Max/Min No Trade(%)	PAM DGP	7.02	8.77	10.40
	LGP	1.42	51.77	1.79

outperforms the other (see Figure 11.22). Further analysis of the success of the trades is provided in Table 2 including: profitable buy trades, protective sell trades, trades conducted, and number of trades not made while maximally or minimally invested. Profitable buys and protective sells were determined in the same way as interday data (see previous Section).

Table 11.2 shows that both algorithms were very successful at choosing trades that led to profitable buys and sells to prevent further losses across every stock. In this respect, the interday and intraday performance of the algorithms was very similar. However, compared to interday data (Figure 11.15), the algorithms traded much less frequently when operating on intraday data. As there were many more opportunities to trade, but the price per trade was kept constant, the algorithms naturally had to be more selective regarding the trades to be conducted. Regarding the number of trades not conducted to allow maximum or minimum investment, it is clearly evident that PAM DGP had a much greater tendency than LGP to stay entirely in or out of the market for GOOG and MSFT, with the converse being true for AAPL.

11.6 Conclusions and Future Work

This work examined the trading performance of a co-evolutionary developmental GP model (PAM DGP) using a genotype-phenotype mapping and a more traditional LGP on four stocks. For interday data, PAM DGP was found to better adapt to guard investments during market downturns and readily take advantage of market gains than LGP. However, for the more volatile intraday data where trading had to be more selective to attempt to be profitable, LGP was found to perform better by not investing as reactively as PAM DGP. For both interday and intraday data, the algorithm was found to not perform optimally in the degenerate case of a moderately volatile, gradual downward slope. However, even in such a deceptive scenario, the GP algorithms performed comparably with buy-and-hold. In both interday and intraday data sets, both algorithms exhibited impressive accuracy in choosing beneficial trades, both for profitable buys and selling to protect investments. The accurate trading ability occurred with moderate levels of trading for interday and lower levels for intraday (indicating more selective trading). Future work will examine options for risk adjusted fitness and portfolio management.

References

1. Brabazon, A., O'Neill, M.: *Biologically Inspired Algorithms for Financial Modelling*. Springer, Berlin (2006)
2. Brameier, M., Banzhaf, W.: *Linear Genetic Programming*. Springer, New York (2007)
3. Drezewski, R., Sepielak, J.: *Evolutionary System for Generating Investment Strategies, Applications of Evolutionary Computing*. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O'Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) *EvoWorkshops 2008*. LNCS, vol. 4974, pp. 83–92. Springer, Heidelberg (2008)
4. Grosnan, C., Abraham, A.: *Stock Market Modeling Using Genetic Programming Ensembles*. *Studies in Computational Intelligence* 13, 131–146 (2006)

5. Wilson, G., Heywood, M.: Introducing Probabilistic Adaptive Developmental Genetic Programming with Redundant Mappings. *Genetic Programming and Evolvable Machines* 8, 187–220 (2007)
6. Wilson, G., Banzhaf, W.: Prediction of Interday Stock Prices Using Developmental and Linear Genetic Programming, Applications of Evolutionary Computing. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., Machado, P. (eds.) *EvoWorkshops 2009*. LNCS, vol. 5484, pp. 172–181. Springer, Heidelberg (2009)
7. Yan, W., Sewell, M., Clack, C.: Learning to Optimize Profits Beats Predicting Returns—Comparing Techniques for Financial Portfolio Optimisation. In: *Proceedings of the 2008 Genetic and Evolutionary Computation Conference (GECCO 2008)*, pp. 1681–1688. ACM Press, New York (2008)