

# Prediction of Interday Stock Prices Using Developmental and Linear Genetic Programming

Garnett Wilson and Wolfgang Banzhaf

Memorial University of Newfoundland, St. John's, NL, Canada  
{gwi1son, banzhaf}@cs.mun.ca

**Abstract.** A developmental co-evolutionary genetic programming approach (PAM DGP) is compared to a standard linear genetic programming (LGP) implementation for trading of stocks across market sectors. Both implementations were found to be impressively robust to market fluctuations while reacting efficiently to opportunities for profit, where PAM DGP proved slightly more reactive to market changes than LGP. PAM DGP outperformed, or was competitive with, LGP for all stocks tested. Both implementations had very impressive accuracy in choosing both profitable buy trades and sells that prevented losses, where this occurred in the context of moderately active trading for all stocks. The algorithms also appropriately maintained maximal investment in order to profit from sustained market upswings.

**Keywords:** Developmental Genetic Programming, Linear Genetic Programming, Computational Finance.

## 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 [1]. 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 interday 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 market analysis. Section 3 describes the LGP and PAM DGP implementations and their application to interday 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 4 for the analysis of four individual stocks in three market sectors. The conclusions and future work follow in Section 5.

## 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 [2]. 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. [3], 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 neural networks or neuro-fuzzy implementations 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 [4] where one species represented 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 has not been applied to market analysis until this work.

In terms of the application of the GP algorithm to inter-day 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 days, but shifts only in increments of 1 day. Following the findings and recommendations of [1], populations are not re-evolved with the shifting of each window.

## 3 PAM DGP and LGP Algorithms for Stock Analysis

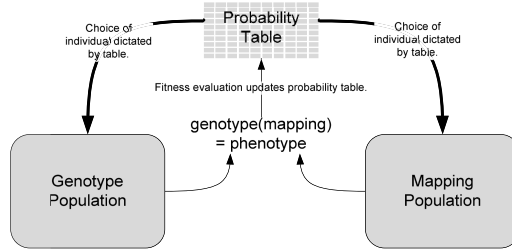
In PAM DGP [5], there is a population of genotypes that cooperatively coevolves with a separate population of mappings. A probability table is updated throughout algorithm execution with entries corresponding to each pair of individual genotype and mapping from both populations. The table entries represent 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 1 and the remaining combinations in that column updated using Equation 2

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

$$P(g, m)_{new} = P(g, m)_{old} - \alpha(P(g, m)_{old}) \quad (2)$$

where  $g$  is the genotype individual / index,  $m$  is the mapping individual / index,  $\alpha$  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 1.



**Fig. 1.** Probabilistic Adaptive Mapping Developmental Genetic Programming (PAM DGP)

Genotypes in PAM DGP are binary strings, with interpretation of sections of the binary string being instruction-dependent (see next Section 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  $s$  and truncating to an integer value (allowing redundant encoding of symbols). 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 a standard LGP implementation [6]. LGP individuals are also bit strings, and there is naturally only a genotype population. The interpretation of instructions is the same for LGP, using a static mapping and constant function set. PAM DGP extends LGP such that members of a function set are adaptively emphasized. Additional details of PAM DGP are available 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.

The PAM DGP and LGP implementations are applied to four stocks, two in the technology sector (Google Inc., ticker symbol “NASDAQ:GOOG”, and Apple Inc., ticker symbol “NASDAQ:AAPL”), one in the energy sector (Chevron Co., ticker symbol “NYSE:CVX”), and one in the consumer/non-cyclical sector (PepsiCo Inc., ticker symbol “NYSE:PEP”). 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 200 day periods from 2007 to 2008, with periods chosen to test against performance against particular stock price trends. The first 16 days of the 200 days 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). Daily data 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

$$v_{scaled} = \frac{v_t - l_n}{h_n - l_n} \quad (3)$$

where  $v_{scaled}$  is the normalized daily trading value,  $v_t$  is the transformed daily 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 ( $n = 16$  in this study).

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  $n$  in the moving window. 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 given the

recommendation of the trade register over all cases in the sliding window is the buy or sell recommendation to the “live” trading 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 2.

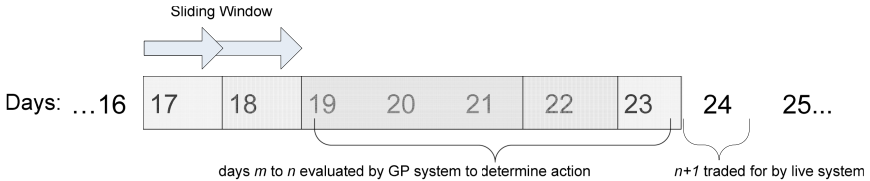


Fig. 2. Relationship between “live” trading system and GP tournament execution

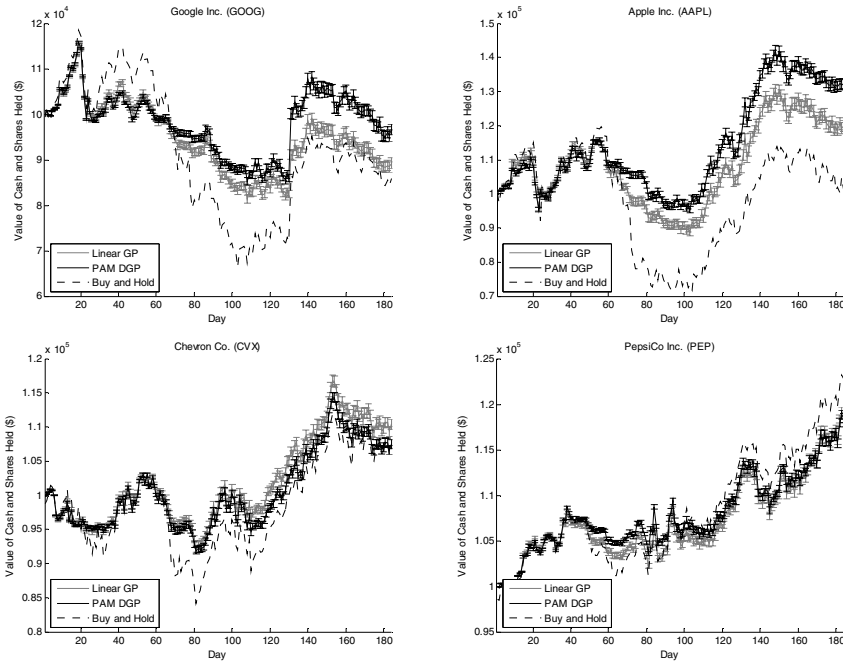
While PAM DGP uses co-evolution to refine function set composition, the appropriate initial function set members must be provided as a basis upon which the algorithm can select its optimum function set. In the case of standard GP, this initial function set remains constant throughout execution. The function set includes standard mathematical operators (+, -, \*) and instructions to trade based on logical operators (<, >, =) applied to the four internal registers. In addition, there are 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  $\pm 2$  standard deviations of the price movement over the last  $n$  days (used here with  $n = 20$ ). A trader is typically alerted when the stock price passes the upper or lower bound of the Bollinger bands.

## 4 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 given in Figure 3.

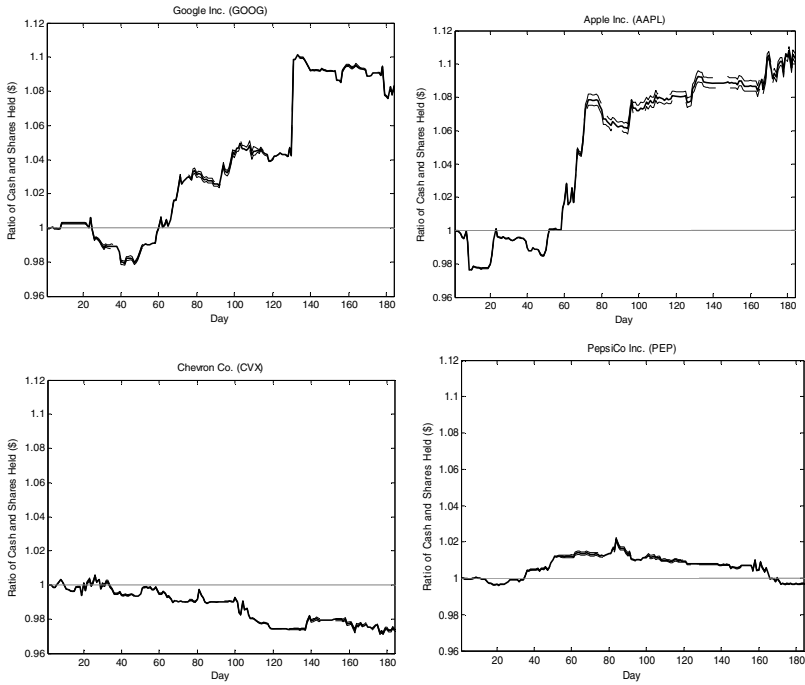
Given Figure 3, the prevalent observation is that PAM DGP and LGP are both impressively robust to share price fluctuations (as indicated by the buy and hold trend

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 of PEP, both algorithms are naturally not maximally invested prior to the large market upswing, and thus achieve less final profit (but are still competitive). Both algorithms achieve final profits better than buy-and-hold for the remaining three stocks (GOOG, AAPL, CVX). Figure 4 provides a ratio of PAM DGP to LGP total worth for a finer comparison, with profit (final and cumulative measures) shown in Figure 5.

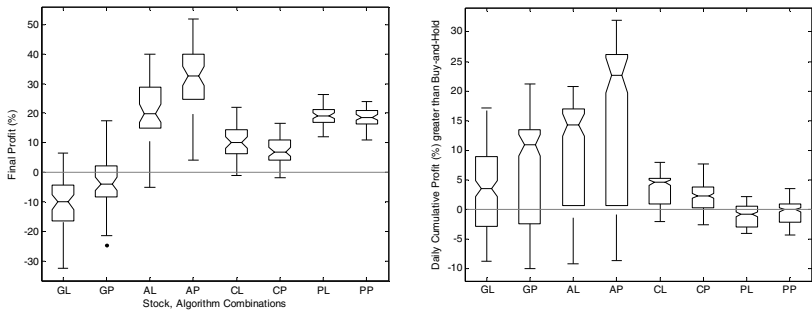


**Fig. 3.** Mean total worth (value of cash and shares) for PAM DGP, LGP, and buy-and-hold strategies over 50 trials with standard error given initial \$100,000 cash value

Comparing the ratio of PAM DGP and LGP worth across stocks in Figure 4, PAM DGP maintains higher worth than LGP for the large majority of trading days in the instances of GOOG, AAPL, and PEP, with LGP dominating PAM DGP almost the entire period for CVX. PAM DGP outperforms LGP by over 10% at times for GOOG and AAPL, but when LGP outperforms PAM DGP throughout CVX it is by a lower margin (just over 5%). Comparing Figures 3 and 4, it is evident that PAM DGP provides increased robustness to market downturns and quickly takes advantage of growth opportunities later in evolution. Also, Figure 3 and 4 indicate that LGP slightly outperforms PAM DGP for CVX by not selling quite as much stock during a market dip immediately preceding a steady climb starting at approximately day 100.



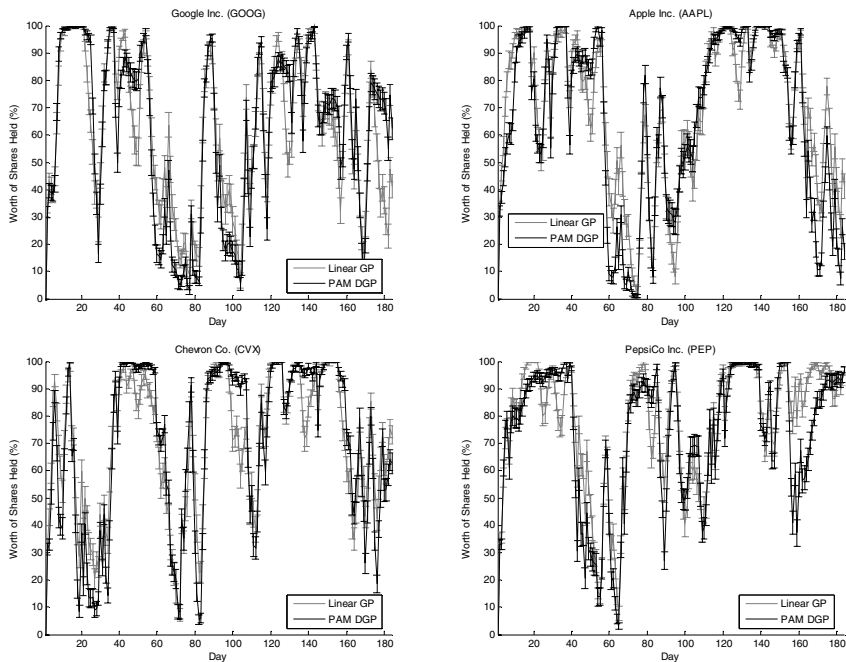
**Fig. 4.** Mean ratio (solid line) of PAM DGP to LGP live trading system total worth over 50 trials with standard error (dotted line). Values greater than 1 indicate greater PAM DGP worth than LGP, values less than 1 *vice versa*.



**Fig. 5.** Boxplot of mean final profit (%) and mean daily 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.

In the boxplots of Figure 5, 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 5, left) for GOOG and AAPL, with LGP slightly outperforming PAM DGP for CVX, and no statistically significant difference in final profits for PEP (all at the 95% confidence interval). Figure 5 (left) also shows impressive final profit for AAPL, PEP, and CVX. There was a general loss for both PAM DGP and 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 5 (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 5 (right) indicates that both PAM DGP and LGP were generally more profitable than buy-and-hold at any given time for all stocks (except, naturally, the case of PEP where naïve buy-and-sell is a very good strategy). PAM DGP was more profitable than LGP at any given time by a large margin for GOOG and AAPL, by a slight margin for PEP, and LGP slightly outperformed PAM DGP for CVX (all at the 95% confidence interval). Number of shares retained daily as a percentage of live trading total worth is shown in Figure 6.

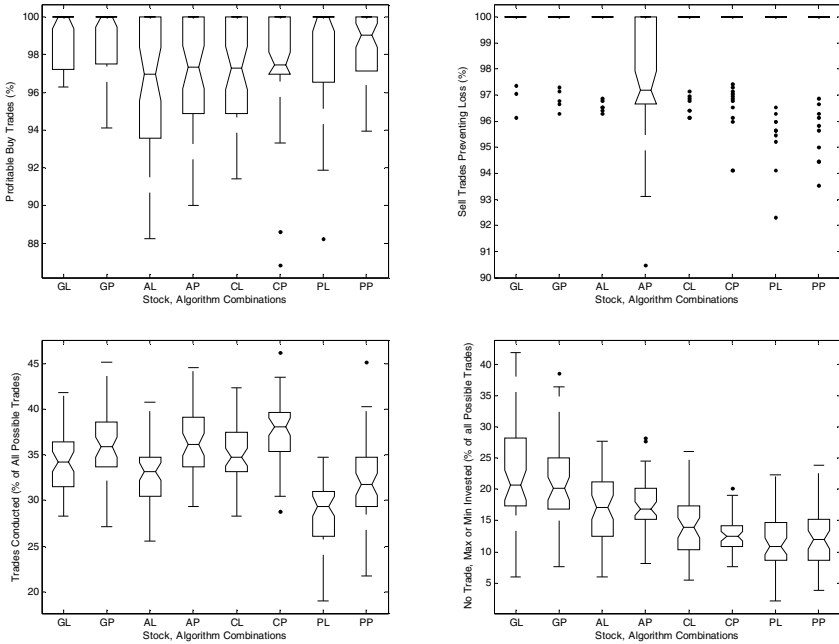


**Fig. 6.** Mean shares held by PAM DGP (black line) and LGP (grey line) live trading systems as a percentage of total worth over 50 trials with standard error

Comparing Figures 3 and 6, 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 3 and 6 also indicate that both algorithms are effective at staying maximally invested during profitable periods. The allocation of resources in or out of the market is a result of the underlying trading



activity, shown in Figure 7. 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 7, respectively. The percentage of trading opportunities where action was taken is shown in Figure 7 (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 7 (bottom right).



**Fig. 7.** 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. First letter of label indicates stock, second letter indicates algorithm.

Figure 7 reveals that both algorithms are extremely accurate at buying to gain profit, with medians of 96% - 100% successful buys. In terms of protecting investment through selling to prevent loss, the median for both algorithms was typically 100%. There is no statistical difference (at the 95% confidence interval) in the ability of PAM DGP or LGP to buy for profit or sell to prevent loss for any of the stocks examined (with the exception of PAM DGP when selling for AAPL). 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, AAPL, and CVX (Figure 7, bottom left), with lower medians for PEP (fewer trades were best to take advantage of the maintained upward trend of PEP). PAM DGP generally conducted more trades (based on spread of data) than LGP for all stocks. Figure 7 (right, bottom) indicates medians between 10% and 22% of trades where the system wished to maintain a maximally or minimally invested position. Compared with Figure 6, it is evident that most of these positions were maximal investment to generate profit. Overall, Figure 7 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.

## 5 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 more traditional LGP on four stocks. Both algorithms were robust to share price fluctuations compared to naïve buy-and-hold strategy. Both algorithms efficiently adapted to guard investments during market downturns and readily take advantage of market gains. PAM DGP definitively outperformed standard LGP in final and cumulative profit in 2 of 4 stocks, with slightly better, competitive, or similar results for the remaining 2 stocks. Both algorithms exhibited impressive accuracy in choosing beneficial trades, both for profitable buys and selling to protect investments, and did so with moderate levels of trading and periods of maximal investing to capitalize on market upswings. 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, Heidelberg (2006)
2. Yan, W., Sewell, M., Clack, C.D.: Learning to Optimize Profits Beats Predicting Returns — Comparing Techniques for Financial Portfolio Optimisation. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2008*, pp. 1681–1688. ACM Press, New York (2008)
3. Grosan, C., Abraham, A.: Stock Market Modeling Using Genetic Programming Ensembles. *Studies in Computational Intelligence* 13, 131–146 (2006)
4. Drezewski, R., Sepielak, J.: Evolutionary System for Generating Investment Strategies. In: Giacobini, M., et al. (eds.) *EvoWorkshops 2008*. LNCS, vol. 4974, pp. 83–92. Springer, Heidelberg (2008)
5. Wilson, G., Heywood, M.: Introducing Probabilistic Adaptive Mapping Developmental Genetic Programming with Redundant Mappings. *Genetic Programming and Evolvable Machines* 8, 187–220 (2007)
6. Brameier, M., Banzhaf, W.: *Linear Genetic Programming*. Springer, New York (2007)