Using Genetic Programming to Predict Financial Data

Hitoshi Iba **Takashi Sasaki**
Dept. of Information and Communication Engineering,
School of Engineering,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, 113-8656, Japan
email: iba,sasaki@miv.t.u-tokyo.ac.jp

Abstract- This paper presents the application of Genetic Programming (GP) to the prediction of the price data in Japanese stock market. The goal of this task is to choose the best stocks when making an investment and to decide when and how many stocks to sell or buy. There have been several applications of Genetic Algorithms (GAs) to the financial problems, such as portfolio optimization, bankruptcy prediction, financial forecasting, fraud detection and scheduling. GP has also been applied to many problems in the time-series prediction. However, relatively fewer studies were made for the purpose of predicting the stock market data by means of GP. This paper describes how successfully GP is applied to predicting the stock data so as to gain the high profit. The comparative experiments are conducted with neural networks to show the effectiveness of the GP-based approach.

1 Introduction

We present the application of Genetic Programming (GP) to a real-world time series prediction, i.e., the prediction of the price data in the Japanese stock market. Our goal is to make an effective decision rule as to when and how many stocks to deal, i.e., sell or buy.

Evolutionary algorithms have been applied to the time series prediction, such as sun spot data [Angeline96] or the time sequence generated from the Mackey-Glass equation [Iba *et al.*96]. Among them, the financial data prediction provides a challenging topic. This is because the stock market data should be handled quite differently from other time series data for the following reasons:

1. The ultimate goal is not to minimize the prediction error, but to maximize the profit gain.
2. Stock market data are highly time-variant, i.e., changeable every minute.
3. The stock market data are given in an event-driven way. They are highly influenced by the indeterminate dealing.

There have been several applications of Genetic Algorithms (GAs) to the financial tasks, such as portfolio optimization, bankruptcy prediction, financial forecasting, fraud detection and scheduling (see Section 5 for related works). However, very few studies were made for the purpose of using GP to predict the stock market data and to maximize the profits. Moreover, relatively fewer works investigated the performance comparison with the traditional methods.

This paper utilizes a GP-based method to predict the price data in Japanese stock market. The financial data we use is the stock price average of Tokyo Stock Exchange, which is called Nikkei225. We show how successfully the decision rule derived by GP predicts the stock pricing so as to gain high profits from the market simulation. The comparative experiments are conducted with neural networks to show the effectiveness of the GP-based approach.

The rest of this paper is structured as follows. Section 2 describes the experimental set-up for GP-based learning. Section 3 explains the experimental results with the market simulation. In section 4, the GP performance is compared with that of the neural network model. Section 5 discusses our approach, followed by some conclusions in Section 6.

2 Experimental Set-Up

2.1 Target Financial Data

The Nikkei225 average is computed by the Nihon Keizai Shimbun-Sha, a well-known financial newspaper publishing firm. The derivation is based upon the Dow formula. As of Feb.,10th,1999, the Nikkei average stood at 13960 yen [1]. However, this average is a theoretical number and should be rigidly distinguished from the real average price in the market place. The computation formula for the Nikkei average is as follows:

$$\text{Nikkei Average} = \frac{\sum_{x \in 225 \text{ stocks}} \text{Price}_x}{D} \qquad (1)$$

The sum of the stock price Price_x is over 225 representative stocks in Tokyo Stock Exchange market. Originally, the divisor D was 225, i.e., the number of component stocks. However, the divisor is adjusted whenever price changes resulting from factors other than those of mar-

[1] More precisely, this is Nikkei225, Osaka, #1. 1US$ was 115.15yen then.

0-7803-5536-9/99/$10.00

ket activity take place. The Nikkei averages are usually given every minute from 9:00am to 12:00pm and from 1:00pm to 3:00pm. The data we use in the following experiments span over a period from April 1st 1993 to September 30th 1993. Fig.1 shows the example tendency of the Nikkei225 average during the above period. All data are normalized between 0.0 and 1.0 as the input value. The total number of data is 33,177. We use the first 3,000 time steps for the training data and the rest for the testing data.

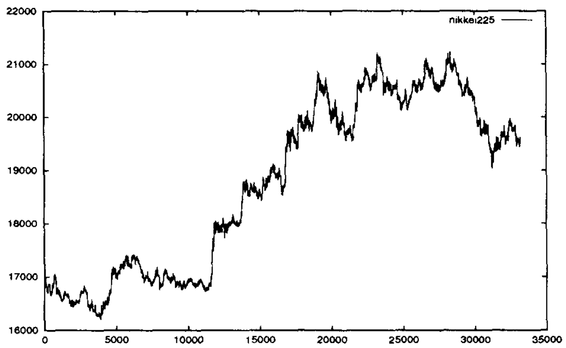


Figure 1: Nikkei225 Data.

2.2 GP Parameters and Experimental Conditions

We chose sgpc1.1, a simple GP system in C language, for predicting the Nikkei225 stock price average. The used parameters are shown in Table 1. For the sake of comparison, GP is run using a variety of terminal sets described below.

- Condition A: The terminal set is $\{\mathrm{y1}, \cdots, \mathrm{y10}, \Re\}$, in which y$i$ is the Nikkei225 price average observed i minutes before the predicted time. That is, if $x(t)$ is the Nikkei225 price average at time t, then y$i = x(t-i)$. $\Re$ is a constant generated randomly.
- Condition B: The terminal set is $\{\mathrm{ave1}, \cdots, \mathrm{ave10}, \Re\}$. The ave$i$ terminal is the average of the Nikkei225 value every 10 minutes, i.e.,

$$\mathrm{ave}i = \frac{\sum_{k=1}^{10} x(t - 10*(i-1) - k)}{10}.$$

- Condition C: The terminal set is $\{\mathrm{m1}, \cdots, \mathrm{m10}, \Re\}$. The m$i$ terminal is the variance of the Nikkei225 value every 10 minutes, i.e.,

$$\mathrm{m}i = \frac{\sum_{k=1}^{10}(x(t - 10*(i-1) - k) - \mathrm{ave}i)^2}{10}.$$

- Condition D: The terminal set is $\{\mathrm{m1}, \cdots, \mathrm{m10}, \mathrm{ave1}, \cdots, \mathrm{ave10}, \Re\}$.
- Condition E: The terminal set is $\{\mathrm{y1}, \cdots, \mathrm{y10}, \mathrm{m1}, \cdots, \mathrm{m10}, \mathrm{ave1}, \cdots, \mathrm{ave10}, \Re\}$.
- Condition F: The terminal set is $\{\mathrm{d1}, \cdots, \mathrm{d10}, \Re\}$, where d$i = x(t-i) - x(t-i-1)$.
- Condition G: The terminal set is $\{\mathrm{v1}, \cdots, \mathrm{v10}, \mathrm{r1}, \cdots, \mathrm{r10}, \Re\}$, where the terminals v$i$ and ri are defined as follows:

$$\mathrm{v}i = |x(t-i) - x(t-i-1)|$$

$$\mathrm{r}i = \frac{x(t-i) - x(t-i-1)}{x(t-i-1)}$$

The predicted value, i.e., the target output of a GP tree, is the current Nikkei225 price average for conditions from A to E. On the other hand, for conditions F and G, the target is the difference between the current Nikkei225 price average and the price observed one minute before. Thus, the fitness value is defined to be the mean square error of the predicted value and the target data. The smaller fitness value, the better.

2.3 Validation Method

In order to confirm the validness of the predictor acquired by GP, we examine the best evolved tree with the stock market simulation during the testing period. Remember that the output prediction of a GP tree is the current Nikkei225 price average for conditions from A to E. Thus, we use the following rule to choose the dealing, i.e., to decide whether to buy or sell a stock. Let $Pr(t)$ be the observed Nikkei225 average at the time step of t.

Step1 Initially, the total budget BG is set to be 1,000,000 yen. Let the time step t be 3000, i.e., the beginning of the testing period. The stock flag ST is set to be 0.

Step2 Derive the output, i.e., the predicted Nikkei225 average, of the GP tree. Let $\widetilde{Pr}(t)$ be the predicted value.

Step3 If $Pr(t-1) < \widetilde{Pr}(t)$ and $ST = 0$, then buy the stock. That is, set ST to be 1.

Step4 Else, if $Pr(t-1) > \widetilde{Pr}(t)$ and $ST = 1$, then sell the stock. That is, set ST to be 0.

Step5 If $ST = 1$, let $BG := BG + Pr(t) - Pr(t-1)$.

Step6 If $BG < 0$, then return 0 and stop.

Step7 If $t < 33,177$, i.e., the end of the testing period, then $t := t + 1$ and go to Step2. Else return the total profit, i.e., $BG - 1,000,000$ yen.

The stock flag ST indicates the state of holding stock, i.e., if $ST = 0$, then no stock is shared at present, whereas if $ST = 1$, then a stock is shared. In **Step5**, the total property is derived according to the newly observed stock price. The satisfaction of the **Step6** condition means that the system has gone into bankruptcy.

For the conditions F and G, the GP tree outputs the difference between the current Nikkei225 price average and the price observed one minute before. Let the predicted output be $\widetilde{Pr}'(t)$. Then the dealing condition depends on the output value itself. More precisely, the above steps are revised as follows:

Step3 If $0 < \widetilde{Pr}'(t)$ and $ST = 0$, then buy the stock. That is, set ST to be 1.

Step4 Else, if $0 > \widetilde{Pr}'(t)$ and $ST = 1$, then sell the stock. That is, set ST to be 0.

We use the above dealing rules for the validation of the acquired GP tree. For the sake of simplicity, we put the following assumptions on the market simulation:

1. At most one stock is shared at any time.

2. The dealing stock is imaginary, in the sense that its price behaves exactly the same as the Nikkei225 average price.

The optimal profit according to the above dealing rule is 80,106.63 yen. This profit is ideally gained when the prediction is perfectly accurate during the testing period.

3 Experimental Results

GP run was repeated under each condition 10 times. The training and the validation performance is shown in Table 2. The MSE values are the average of mean square errors given by the best evolved tree for the training data. The hit percentage means how accurately the GP tree made an estimate of the qualitative behavior of the price. That is, the hit percentage is calculated as follows:

$$\begin{aligned} \text{hit} &= \frac{N_{\text{up\_up}} + N_{\text{down\_down}}}{N_{\text{up\_up}} + N_{\text{up\_down}} + N_{\text{down\_up}} + N_{\text{down\_down}}} \\ &= \frac{N_{\text{up\_up}} + N_{\text{down\_down}}}{30,177}, \end{aligned}$$

where $N_{\text{up\_up}}$ means the number of times when the tree makes an upward tendency while the observed price rises, and $N_{\text{down\_up}}$ means the number of times when the tree makes a downward tendency while the observed price falls, and so on. The total number of the predictions is 30,177, which equals the number of testing data.

All experimental results except condition E show that there seems to be a strong relationship between the MSE value, the hit percentage, and the profit gain. The lower the MSE value is, the higher both the hit percentage and the profit gain are. However, this is not necessarily a matter of course, because achieving the high profit requires more accurate prediction for the critical tendency change, i.e., when the stock price suddenly falls (rises) reversely after the price rises (falls) before. Note that the same weight is put on all of the training data for the MSE calculation.

The table shows that the average and best hit percentages were below 50% under the conditions B, C, D and E, which resulted in the low profit and the negative returns except the condition C. On the other hand, under the conditions A, F and G, the average hit percentage was over 50% and the best one was over 60%, which led to the high and positive profit gain. Especially, GP runs under the condition G resulted in the average hit percentage of 60% and over. The acquisition of profitable GP trees under the condition A might be a counter-evidence to the efficient market hypothesis. However, its performance is not necessarily stable, as can be seen from the average data in the table. It seems to be a puzzle that the performance under the condition E is worse than the condition A, which is a subset of the condition E. This is because the search efficiency is influenced adversely by the size of the terminal set. Thus, the effective choice of terminals is very important for GP search. The example acquired trees, i.e., the best evolved GP predictors, are shown in Appendix A. Fig.2 shows the prediction of the normalized Nikkei225 price by the best evolved tree under the conditions A and G. The predicted value of Nikkei225 price for the first 100 minutes is shown for condition A. The predicted difference between the current Nikkei225 price and the price one minute before is plotted for condition G. Fig.3 illustrates the optimal profit and the profits gained by the predicted trees. These results provide the evidence that the predicted difference under the condition G corresponds to the observed qualitative behavior, i.e., the upward or downward tendency, of the Nikkei225 price. This causes the high profit gain shown in Fig.2.

To summarize the above experimental results, we can confirm the following points:

1. The average or variance terminals were not effective for the prediction.

2. Using only past data or difference values led to the unstable prediction.

3. The most effective terminal set included the absolute values and the directional values of the difference between the current Nikkei225 price and the past one.

As can be seen in the condition E result, the good performance for the training prediction, i.e., the small MSE value, does not necessarily guarantee the high profit gain.

Table 1: GP Parameters for sgpc1.1.

max_generation	100	max_depth_after_crossover	17
population_size	1000	max_depth_for_new_trees	6
steady_state	0	max_mutant_depth	4
grow_method	GROW	crossover_any_pt_fraction	0.2
tournament_K	6	crossover_func_pt_fraction	0.7
selection_method	TOURNAMENT	fitness_prop_repro_fraction	0.1
function set	$\{+, -, *, \%, \sin, \cos, \exp\}$		

Table 2: Experimental Results (GP)

	Training	Testing			
		Hit(%)		Profit gain(yen)	
Condition	MSE	Average	Best	Average	Best
A	1.79e-06	55.02	62.78	12411.01	31256.06
B	1.22e-05	47.47	48.17	-4093.22	-2341.50
C	5.82e-04	50.42	51.00	127.03	305.13
D	1.28e-05	41.09	51.64	-19727.52	-3811.19
E	1.94e-06	49.45	50.34	-806.33	1061.38
F	1.64e-06	55.02	62.49	14276.06	30722.81
G	1.80e-06	61.38	62.56	28942.03	30896.56

This is due to the above-mentioned difficulty with the validation of the dealing simulation. We can avoid this difficulty by evolving a GP tree as as a predictor of the delayed difference instead of the exact Nikkei255 value. The effective dealing rule might as well make precise decisions on upward or downward tendencies as lower the MSE values for the training data. Therefore, GP runs under conditions F and G led to the constantly high profit gains. However, there is not much theoretical background for the above best choice. In general, the terminal and function sets play an essential role in GP search, but they are problem-dependent and not easy to choose. We are currently researching on the further analysis on this topic, i.e., the statistical analysis of the effective choice of the terminal set, by means of the fitness landscape approach [Manderick 91].

4 Comparative Experiment with Neural Networks

For the sake of comparison, we apply Neural Network (NN) to the same prediction task and examine the performance difference. We used the program available at "Neural Networks at Your Fingertips" [Kutza96]. This NN program implements the classical multi-layer backpropagation network with bias terms and momentum. It is used to detect structure in time-series, which is presented to the network using a simple tapped delay-line memory. The program originally learned to predict future sunspot activity from historical data collected over the past three centuries. To avoid overfitting, the termination of the learning procedure is controlled by the so-called stopped training method (see [Rumelhart *et al.*86] for the details).

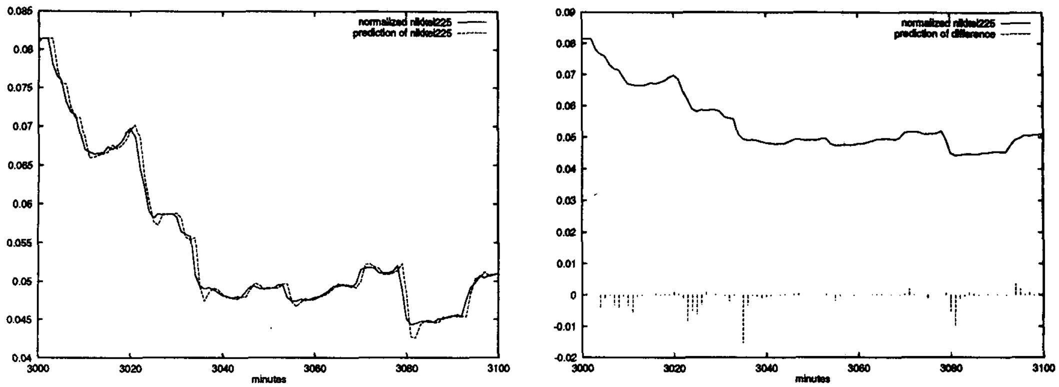
The NN parameters used are shown in Table 3. The network was trained under the previous conditions A and G. That is, the input variables of the network was set to be $\{y1, \cdots, y10\}$ for A, and to be $\{v1, \cdots, v10, r1, \cdots, r10\}$ for G. The random constant $\Re$ is omitted. Table 4 shows the experimental results. The data are averaged over 10 runs. Fig.4 shows the prediction of the normalized Nikkei225 price. Fig.5 illustrates the optimal profit and the profits gained by the neural networks. Comparing these results with the ones in Table 2, we can confirm that NN gave much worse results than GP. The reason seems to be that the neural network suffers from the overfitting, as can be seen in the table. Moreover, the computational time is much longer for the convergence for the neural network. Thus, we can conclude the superiority of GP over NN.

5 Discussion

5.1 Future Works

The above experimental results have shown the effectiveness of GP-based approach for predicting financial data. However, there are several points to be improved for the sake of practical use. For instance, the following extensions should be considered:

1. The dealing simulation should be more realistic including the payment of the commission. The profit gain is offset with the fee.



(a) Condition A. (b) Condition G.

Figure 2: Prediction Results by GP.

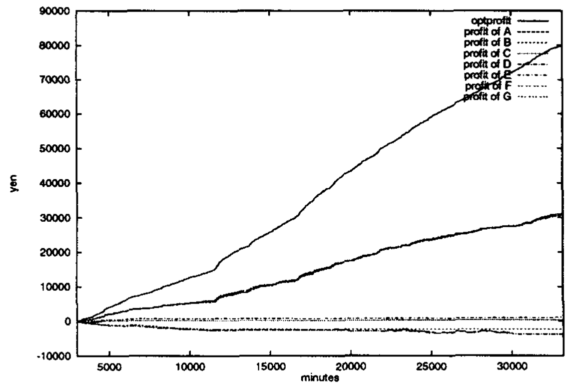


Figure 3: Optimal Profit and Profits Gained by GP.

2. The prediction accuracy should be improved. Especially, we should put much more emphasis on the short-term or real-time prediction, rather than the long-term prediction.

3. The problem-specific knowledge, such as economical index options or foreign exchange rates, could be introduced for the further performance improvement.

As for the third point, we are now in pursuit of the quantitative factor analysis for the purpose of choosing the significant economical features. This will have an essential impact on the prediction accuracy, especially for the short-term prediction.

We are applying STROGANOFF to the financial problem as our main research concerns. STROGANOFF is a numerical GP system, which effectively integrates traditional GP adaptive search and statistical search [Iba *et al.*96]. The preliminary results obtained by STROGANOFF were satisfactory and promising. However, we also observed the overfitting difficulty. This is probably because STROGANOFF used the polynomial regression, which led to finding the highly fit polynomials in terms of MSE or MDL values. But this did not necessarily give rise to the high profit gain as mentioned earlier. We believe that this difficulty will be avoided by using the discrete terminals, such as a step function or a sign function (see [Iba *et al.*96] for the discussion). The extension of STROGANOFF in this direction is our future research topic.

5.2 Related Works

Evolutionary algorithms have been applied to several financial problems. For instance, [Hiemstra96] applied GA to the tactical asset allocation and showed that the established system outperformed a passive rebalancing policy. [Dorsey and Mayer95] examined GA as a method for solving optimization problems in econometric estimation. [Lettau97] used GA as a leaning model to study portfolio decisions of boundedly rational agents in a financial market. He found that the GA-based agent learned to hold too much risk as compared to the optimal portfolio of rational investors and exhibited an asymmetric response after positive and negative returns where the portfolio adjustment is more pronounced after negative returns. Investors in mutual funds showed the same investment patters as the adaptive agents. Thus,

Table 3: Neural Network Parameters

#. of Layers	3	#. of hidden nodes	10
α	0.5	BIAS	1
η	0.05	EPOCHS	10
Gain	1	#. of LOOP	20000

Table 4: Experimental Results (NN)

	Training	Testing			
		Hit(%)		Profit gain(yen)	
Condition	MSE	Average	Best	Average	Best
A	1.23e-6	50.00	50.22	2509.88	3116.81
G	2.62e-6	55.58	60.37	14114.40	28315.88

he concluded that the model with entry and exit of agents could match the mutual fund data very closely. [Arifovic96] employed a model of bounded rationality in which agents updated their decision rules using GA in a version of the Kareken-Wallace two-country overlapping generations model. Agents used GA to update their decisions regarding savings and the portfolio composition, that is, what fraction of savings to place in each currency. The behavior of the exchange rate, portfolio decisions, and composition decisions in the GA simulations exhibited features similar to those observed in the experiments with human subjects. Most of the GA exchange rate values fell within the range of the experimental exchange rate data.

GP has also been applied to some financial problems. [Neely *et al.*97] used GP to find the technical trading rules in the foreign exchange market. They observed the economically significant out-of-sample excess returns to those rules for each of six exchange rates. [Bhattacharyya *et al.*98] applied GP to learn the trading model. The experimental results showed that the acquired solutions were simpler, easier to interpret and less prone to overfit for the high-frequency data from the foreign exchange markets. [Chen *et al.*98] used GP to derive option pricing formulas. They used the real data from S&P 500 index options for the training and testing data. The comparison with the traditional model, i.e., the Black-Scholes formulas, showed the superiority of the GP pricing formulas in their limited experiments. [Chidambaran *et al.*98] also applied GP to approximate the relationship between the price of a stock option, the terms of the option contract, and properties of the underling stock price. They showed the advantages of GP over the Black-Scholes model in the sense that GP-based model could price CBOE index and equity options.

6 Conclusion

This paper presented the application of GP to the prediction of stock price data in order to gain the high profit in the market simulation. We confirmed the following points empirically:

1. GP was successfully applied to predicting the stock price data. That is, the MSE value for the training data was satisfactorily low, which gave rise to the high profit gain in the dealing simulation.
2. The performance under a variety of conditions, i.e., different terminal sets, was compared. Using the terminals based upon the delayed difference of the stock price were more effective than using the exact price values.
3. The GP performance was compared with that of neural networks, which showed the superiority of GP-based method.

We are currently working on the extension of our method for the sake of improving the profit gain in the market simulation. Especially, we are interested in integrating GP-based adaptive search and machine learning techniques, i.e., boosting and bagging methods [Iba99]. Introducing the problem-specific knowledge, such as economical index options or foreign exchange rates, is our future research concerns.

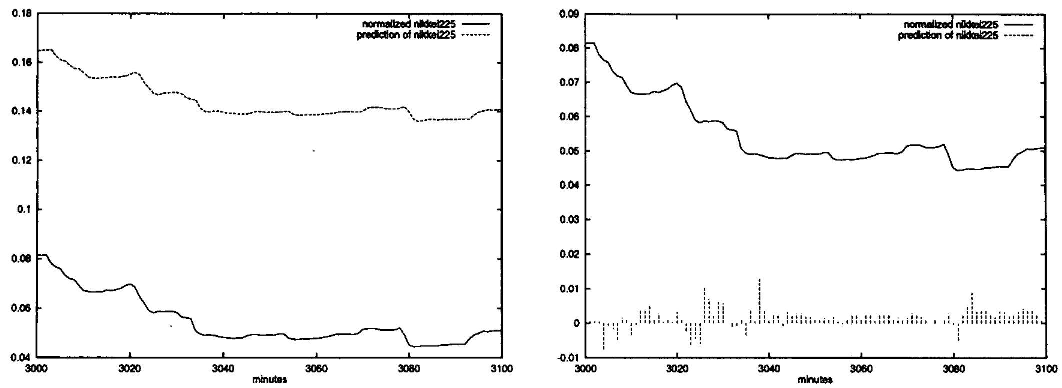
Acknowledgments.
We are grateful to Institute of Investment Technology, Nikko Securities Co., Ltd., for providing the Nikkei225 stock price data, which we used to conduct our comparative experiments.

Bibliography

[Angeline96] Angeline,P., Two Self-Adaptive Crossover Operators for Genetic Programming, *Advances in Genetic Programming 2*, Angeline,P. and Kinnear,K. (eds.), MIT Press, 1996

[Arifovic96] Arifovic,J., The Behavior of the Exchange Rate in the Genetic Algorithm and Experimental Economies, *Journal of Political Economy*, vol.104, no.3, 1996

[Bhattacharyya *et al.*98] Bhattacharyya,S., Pictet,O., and Zumbach,G., Representational Semantics for Genetic Programming based Learning in High-Frequency Financial



(a) Condition A. **(b) Condition G.**

Figure 4: Prediction Results by NN.

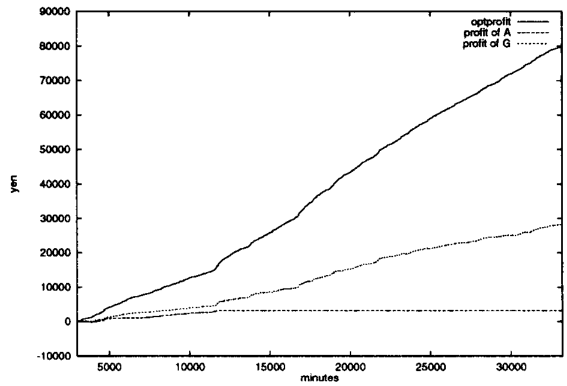


Figure 5: Optimal Profit and Profits Gained by NN.

Data, *Proc. of the 3rd Annual Genetic Programming Conference*, 1998

[Chen *et al.*98] Chen,S.-H., Yeh,C.-H., and Lee,W.-C., Option Pricing with Genetic Programming, *Proc. of the 3rd Annual Genetic Programming Conference*, 1998

[Chidambaran *et al.*98] Chidambaran,N.K., Lee,C.H.J., and Trigueros,J.R., An Adaptive Evolutionary Approach to Option Pricing via Genetic Programming, *Proc. of the 3rd Annual Genetic Programming Conference*, 1998

[Dorsey and Mayer95] Dorsey,R.E. and Mayer,W.J., Genetic Algorithms for Estimation Problems with Multiple Optima, Nondifferentiability, and Other Irregular Features, *Journal of Business & Economic Statistics*, vol.13, no.1, 1996

[Hiemstra96] Hiemstra,Y., Applying Neural Networks and Genetic Algorithms to Tactical Asset Allocation, *Neuro Ve$t Journal*, May/June, 1996

[Iba *et al.*96] Iba, H., deGaris, H. and Sato, T. Numerical Approach to Genetic Programming for System Identification, in *Evolutionary Computation*, vol.3, no.4, 1996

[Iba99] Iba, H. Bagging, Boosting, and Bloating in Genetic Programming, In *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith R. E. (eds.), Morgan Kaufmann, 1999

[Kutza96] Kutza,K., Neural Networks at Your Fingertips, available at http://www.geocities.com/CapeCanaveral/1624/, 1996

[Lettau97] Lettau,M., Explaining the Facts with Adaptive Agents: the Case of Mutual Fund Flows, *Journal of Economic Dynamics and Control*, vol.21, 1997

[Manderick 91] Manderick, B., deWeger, M., and Spiessens, P., The Genetic Algorithm and the Structure of the Fitness Landscape, *Proc. 4th Int. Joint Conf. on Genetic Algorithms (ICGA91)*, 1991

[Neely *et al.*97] Neely,C., Weller,P., and Dittmar,R., Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach, *Journal of Financial and Quantitative Analysis*, vol.32, no.4, 1997

[Rumelhart *et al.*86] Rumelhart,D.E., Hinton,G.E., and Williams,R.J., Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing, Volume 1*, Rumelhart,D.E. and McClelland,J.L. (Eds.) MIT Press, Cambridge, MA, pp. 318-362, 1986

Appendix A: Acquired Trees by GP

Condition	Tree
A:	(+ y1 (* y7 (* (* (* (* (* (* (* (- y1 y3) (% (* y4 y10) (* (+ y6 y10) (* (+ y8 y4) (- (+ y6 y10) (- y9 y6)))))) (% (* y3 y10) (* y2 y9))) (% (* y4 y10) (* (* (- y9 y9) (% (+ y8 y4) (% (- (- (- (* y2 y9) (- y9 y6)) (* (* (- y9 y6) (+ y6 y10)) (* y3 y10))) (* y10 (+ y6 y10))) (- y9 y6)))) (* y4 y10)))) (% (* y2 y9) (* (- (+ y6 y10) (- y9 y6)) (* (+ y8 y4) (+ y8 y4))))) (% (* y3 y10) (* y2 y9))) (% (* y4 y10) (* (* (- y9 y9) (% (* (- y9 y9) (% (- y9 y6) (* y3 y10))) (% (- (- (* (* y3 y10) (* y4 y10)) (* (* (* y3 y10) (+ y8 y4)) (- (* (- y9 y9) (% (- y9 y6) (+ y6 y10))) (- y9 y6)))) (* y3 y10)) (- y9 y6)))) (* y4 y10)))) (% (* y4 y10) (* y2 y9)))))
B:	(+ (* (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ (+ m6 m3) m3) m8) m1) m8) m3) m3) m3) m8) m1) m8) m3) m8) m1) m1) (- (- (* (- m1 m8) m3) (* (* m10 m10) (* (+ m1 m4) (* (+ m1 m4) (+ (+ (- (* (- m1 m8) m3) (* (* (* m10 m10) (- m1 m8)) (* (+ m6 m3) (+ m10 m3)))) m1) m1))))) (* (- m2 m8) (+ (* (- m8 m9) (* (* (+ (+ (+ (+ (+ (+ (+ m6 m3) m3) m8) m1) m8) m3) m3) (+ (+ (+ (+ (+ (+ (+ (+ m6 m3) m3) m8) m3) m8) m1) m8) m3)) (+ (+ (+ (+ (+ (+ (+ (+ m6 m3) m3) m8) m3) m8) m8) m4) m1))) m4)))) m1)
C:	(+ (* m7 m6) (- (* (* (+ (+ m8 m4) (+ m3 m7)) (- (- 2.261061 (* (- (* m7 m6) 2.649165) (% (- 2.261061 (% m1 m1)) (+ m5 3.703032)))) (- m5 2.700706))) (* (* (+ m5 3.703032) (+ (- m3 m9) (% (* (+ (* m9 2.649165) (% m9 (+ m5 3.703032))) (- (% m5 m5) (- m4 m9))) m9))) (+ m5 3.703032))) (% (* (- (* m7 m6) 2.700706) (% (- 2.261061 (% m1 m1)) (+ m5 3.703032))) (* (+ (- (* (* (+ (* (+ (* m9 2.649165) (% (+ (- m4 m9) (* m5 m1)) (- (% m5 m5) (* (+ m5 3.703032) (- (+ m5 3.703032) (- m5 2.700706)))))) (- (% m5 m5) (* (+ m5 3.703032) (- (+ m5 3.703032) (- m5 2.700706))))) (* m2 m5)) (- (% m1 m1) (- m5 2.700706))) (* (* (+ (- (+ (+ (- m4 m9) (* m5 m1)) (* m5 m1)) (* m7 m6)) (+ m5 3.703032)) (+ (- m3 m9) (+ m5 3.703032))) (* (% m5 m5) (% (* (+ (* m9 2.649165) (* m10 3.555135)) (- (% m5 m5) (+ (* m9 2.649165) (% (+ (- m4 m9) (* m5 m1)) (% (% (+ m5 3.703032) m7) m7))))) m9)))) (% (* (- m5 (- (% m5 m5) (- m5 2.700706))) (% (- 2.261061 (% m1 m1)) (+ m5 3.703032))) (* (% m5 m5) (% (* (+ (* m9 2.649165) (% (* (+ (% m2 m7) (* m5 m1)) (- 2.261061 (% m5 m5))) (% (+ m5 3.703032) m7))) (- (% m5 m5) (* m7 m6))) m9)))) (+ m5 3.703032)) (+ (- m3 m9) (% (* (+ (* m9 2.649165) (% (+ (- m4 m9) (* m5 m5)) (% (+ m5 3.703032) m7))) (- (% m5 m5) (* (+ m5 3.703032) m6))) m9))))))
D:	(- (+ (+ m1 (+ m1 (+ (+ m1 (+ m1 (+ (- (+ (+ (- (+ m1 (+ (+ m1 (+ (+ ave1 m1) m1)) m1)) (* (+ ave1 m1) (* (* (+ (+ m2 ave9) m1) (+ (+ m3 ave9) (+ m2 ave9))) (+ ave4 m1)))) m1) m1) (* ave9 (* ave9 (* (+ m2 ave9) (+ m2 (+ m1 (+ (+ (+ ave1 m1) m1) m1))))))) m1))) m1))) m1) (* (- (+ m1 (+ (+ m1 (+ (+ (- (+ (+ (- (+ m1 (+ (+ m1 (+ (+ ave1 m1) m1)) m1)) (* ave9 (* (* (+ m1 m1) (+ (+ m2 ave8) ave8)) (+ m3 m3)))) m1) m1) (* (+ ave1 m1) (+ (+ ave1 m1) m1))) m1) m1)) m1)) (* (+ (+ m2 ave8) m1) (+ (+ m2 ave8) ave8))) (+ (- (+ ave4 m3) (+ m1 (+ (- (+ (+ (- (+ m1 (+ (+ m1 (+ m1 (+ (+ ave1 m1) m1))) m1)) (* (+ ave1 m1) (* (* (+ (+ ave4 m3) m1) (+ (* (+ (+ m2 ave9) m1) (+ (+ m3 ave9) (+ m2 ave9))) m1)) (+ ave4 m3)))) m1) m1) (* ave9 (* ave9 (* (+ m2 ave9) (- (+ m1 (+ (+ m1 (+ (+ m3 ave9) (+ m2 ave9))) m1)) (* (+ ave1 m1) (* (* (+ (+ m2 ave9) m1) (+ (+ m3 ave9) (+ m2 ave9))) ave4))))))) m1))) m2)))
E:	(- (- (- (- (+ (- (- (- (+ m6 (- (- (- (- (- (- (- (- y1 m3) m5) m5) m2) m10) m10) m2) m7)) m7) m5) m4) (* m3 (% (- (+ (- (- (- (% (- (- (- (- y1 m3) m3) (- m7 m2)) m3) (+ (- m7 m2) (+ (- m7 m2) y5))) m7) m10) m5) m6) (- (+ (- m7 m5) y5) m5)) (+ (- m7 m2) (- (+ (- m7 m2) (+ (- m7 m2) (+ (- m7 m2) y5))) m6))))) m4) m3) m5) m3)
F:	(% (- d1 (% (+ d1 d1) (+ (+ 4.834796 d4) (* (% (% d1 (+ (- d6 d4) (- d2 d3))) 1.484999) (+ (% (* (+ (+ d3 (% d1 1.484999)) d1) (- (+ (+ (+ 4.834796 d4) (% (% d1 1.484999) 1.484999)) (% (* (+ (+ 4.834796 d4) (+ (% (+ (% d1 1.484999) d1) (+ d1 d1)) (* (+ 4.834796 d4) (+ (+ 4.834796 d4) (% d1 1.484999))))) (+ (- d6 d4) (- d6 d4))) 1.484999)) d2)) (+ (+ 4.834796 d4) d1)) (* (- d1 d8) (+ (+ 4.834796 d4) (% (+ d1 (+ (% (- d6 d4) 1.484999) (+ d1 (+ (% (- d1 d4) 1.484999) (% (+ (% d1 1.484999) d1) (+ d1 d1)))))) (+ (+ (+ (+ 4.834796 d4) (- d9 d9)) (% (+ d1 (% d1 (+ (+ 4.834796 d9) (% d1 1.484999)))) (+ (+ (+ (% (% d1 1.484999) 1.484999) d4) (- d9 d1)) (% (% d1 1.484999) 1.484999)))) (+ (* (+ (+ (+ 4.834796 d4) d1) (+ (% (+ d3 (- d9 d1)) (+ d1 d9)) (* (+ (+ 4.834796 d4) (% d1 1.484999)) (+ (+ 4.834796 d4) (% d1 1.484999))))) (+ (- d6 d4) (- d6 d4))) (% (% d1 1.484999) 1.484999))))))))))) 1.484999)
G:	(+ (+ (+ (+ (* r1 v1) (* r1 v5)) (* r1 v5)) (+ (* r1 v5) (+ (* r1 v1) (* v4 r2)))) (+ (* r1 v1) (* r1 v5)))