A Genetic Programming Based Stock Price Predictor together with Mean-Variance Based Sell/Buy Actions

Ramin Rajabioun and Ashkan Rahimi-Kian

Abstract— **In this paper first a precise mathematical model is obtained for four competing or cooperating companies' stock prices and then the optimal buy/sell signals are ascertained for five different agents which are trading in a virtual market and are trying to maximize their wealth over one trading year period. The model is so that gives a good prediction of the next 30th day stock prices. The companies used in this modeling are all chosen from Boston Stock Market. Genetic Programming (GP) is used to produce the predictive mathematical model. The interaction among companies and the effect imposed by each of five agents on future stock prices are also considered in our modeling. Namely, we have chosen eight companies in order that there is some kind of interrelation among them. Comparison of the GP models with Artificial Neural Networks (ANN) and Neuro-Fuzzy Networks (trained by the LoLiMoT algorithm) shows the superior potential of GP in prediction. Using these models; five players, each with a specific strategy and all with one common goal (wealth maximization), start to trade in a virtual market. We have also relaxed the short-sales constraint in our work. Each of the agents has a different objective function and all are going to maximize themselves. We have used Particle Swarm Optimization (PSO) as an evolutionary optimization method for wealth maximization.**

Key words: **Stock market model, price prediction, Genetic Programming (GP), wealth maximization, mean-variance portfolio selection, Particle Swarm Optimization (PSO).**

I. Introduction

FORECASTING the change in market prices and making correct decisions is one of the most principal needs of anyone who economical environments concerns him. Time series are the most common methods used in price prediction [1-3]. But the predominant defect of these methods is that they use only the history of a company's price to do a prediction. Recently, there has been growing attention to the models that concern the interaction among companies in modeling and the use of game theory [4-6] in decision making because of providing more realistic models. Because of complexity of the mutual effects of each company on the others, methods like Artificial Neural Networks (ANN), Neuro-Fuzzy Networks and State Space (SS) models are used more often for the stock price modeling. In [7-10] Neural Network is used to model the stock market and make prediction. In [8], Genetic algorithm (GA) is incorporated to improve the learning and generalizability of ANNs for stock market prediction. The proposed approach has reduced the dimensionality of the feature space and has decreased irrelevant factors for stock market prediction. In [11] the difference between the price and the moving average, highest and lowest prices is used as inputs for one-day-ahead price prediction. More over, volume of transactions, market indicators and macro economic data are also considered as input variables [12]. There are also some studies being performed on the fluctuations and the correlations in stock price changes in physics communities, using the concepts and methods in physics [13-14]. In [15] a neuro-genetic stock prediction system is introduced, which is based on the financial correlations among companies. The genetic algorithm is used to select a set of informative input features among them for a recurrent neural network. In [16-17], the neuro-genetic hybrids for stock prediction are proposed. The genetic algorithm (GA) is used to optimize the weights of the neural network.

Producing the right buy/sell signals are also important for those who trade in the stock markets. In [18], two simple and popular trading rules including moving average and trading range breakout are tested in the Chilean stock market. Their results were compared with the buy-and-hold strategy, and both trading rules produced extra returns compared to the buy-and-hold strategy.

Genetic Programming (GP) is a symbolic optimization technique, developed by Koza [19]. It is an evolutionary computational technique (like, e.g., genetic algorithm, evolutionary strategy, etc.) based on the so-called "tree representation". This representation is extremely flexible because trees can represent computer programs, mathematical equations, or complete models of process systems [20]. In [21] GP is used to produce a one-day-ahead model to predict stock prices. This model is tested for a fifty consecutive trading days of six stocks and has yielded relatively high returns on investment.

In this paper we use the GP to find the best mathematical models for the four companies' stock prices under study. Our GP models are able to predict these stock prices for up to the next 30 days with acceptable prediction errors in the market. Because, the GP is a well known algorithm we will not present it in details. However, reference [22] provides a good review of the GP algorithm.

The modeling is done for four companies in Boston Stock Market [23]. Selected companies include: *Advanced Micro Devices (AMD), Ericsson (ERIC), Sony (SNE), Philips (PHG), International Business Machines (IBM), Intel Corporation (INTC), Microsoft (MSFT) and Nokia (NOK)*. These companies are assumed to have a relationship like

R. Rajabioun and A. Rahimi-Kian are with the School of Electrical and Computer Engineering, Control and Intelligent Processing Center of Excellence, Univ. of Tehran, Tehran, Iran (Tell: +98-21-82084303, Fax: +98-21-88633029, Emails: r.rajabioun@ece.ut.ac.ir, arkian@ut.ac.ir).

competition or cooperation and so their stock prices could affect on each other. Letters allocated in parentheses are the symbols using which one can access the price data of each company. We use the price history of these eight companies as inputs to predict our four objective companies' prices including: ***Ericsson (ERIC), International Business Machines (IBM), Sony (SNE) and Philips (PHG)***. Obtained four models precision is compared with two traditional methods: (1) Multi Layer Perceptron (MLP) and (2) Neuro-Fuzzy Network trained by Locally Linear Model Tree (LoLiMoT) method. The data are grouped in two sets: the train data (the first 70%) and the test data (the last 30%). After modeling the four companies' stock prices, we create five agents who trade in a virtual market in order to maximize their wealth. These agents (players) will buy or sell their in hand stocks according to their uniquely defined strategies. Each player has a unique objective function. The Buy/Sell actions of each player are obtained so as to maximize its objective function in each trading period. The maximization is done using the Particle Swarm Optimization (PSO) method [24].

The structure of the rest of paper will be as follows: In section 2, modeling and prediction is discussed. Section 3 demonstrates the virtual stock market and argues its constraints and presumptions. Then in section 4 the results of our simulations are shown and finally the conclusion is done in section 5.

II. Modeling and Prediction

As stated earlier, our primary goal is to obtain a predictive model that is able to predict the future stock prices precisely. The companies that we are going to predict their stocks include: *Ericsson (ERIC), International Business Machines (IBM), Sony (SNE) and Philips (PHG)*. We presume that these companies have some kind of interrelations with four other companies including: *Advanced Micro Devices (AMD), Intel Corporation (INTC), Microsoft (MSFT) and Nokia (NOK)*. So we downloaded these eight companies' price data from the Boston Stock Market [23]. The downloaded data encompasses some information like: *daily opening price, daily closing price, daily highest price, daily lowest price and exchange volume*. In this paper, we predict the *average of daily opening and closing prices*. Our data set contains sampled price data for the interval of 2001/07/08 to 2006/03/11. We divided these data in two groups: train data (the first 70%) and test data (the last 30%). The test data are used to verify the obtained model's accuracies. The criterion used to evaluate these models is the Normalized Mean Square Error (NMSE), which is defined as follows:

$$NMSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2} \qquad (1)$$

where y_i and $\hat{y}_i$ are the original and predicted price values, respectively.

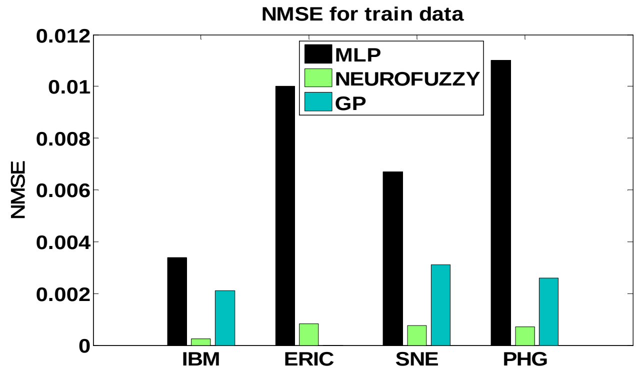

Fig. 1- The prediction error (NMSE) for all companies (using train data)

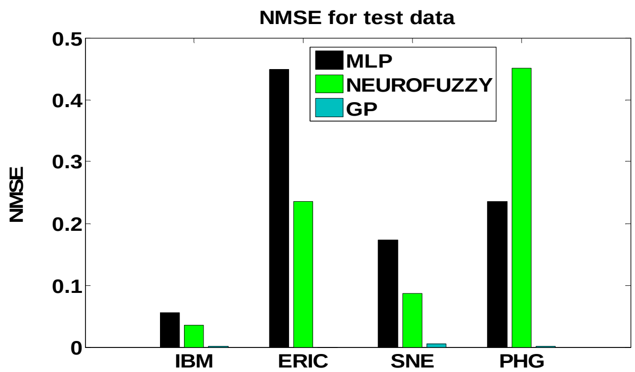

Fig. 2- The prediction error (NMSE) for all companies (using test data)

Figure 1 shows the NMSE values for the train data set using GP, ANN (MLP) and Neuro-Fuzzy networks (trained with LoLiMoT algorithm). Figure 2 depicts this comparison for the test data set.

The GP-based stock price models were initialized with some functions and terminals. The terminals included random number generators together with integers from 1 to 16. The functions included: $\{+, -, \times, \div, \log x, e^x, x^y\}$. The population sizes were set to 600 except for ERIC, which was set to 400. It is noteworthy that the population size was initialized to 200 and then increased. Meanwhile the number of iterations was set to 80. As it can be seen from figures 1 and 2, the GP-based price model prediction errors are acceptable for the training data set and less than both of the MLP and Neuro-Fuzzy models for test data set.

The only drawback of the GP algorithm is its time-consuming modeling characteristics, which is acceptable comparing to its precise modeling, especially for stock price prediction applications that precision is an important factor.

Until now we have modeled the interactions of eight different companies that affect the future price values. But due to the fact that buyers/sellers also affect future stock prices of the companies, it is essential to include such interactions in the modeling. Therefore, after modeling the stock prices for the above mentioned companies, we augment a new term to our price models in order to include the effects of the market players' actions (buy/sell weights) into the future price changes. Since there are not much available data on how the buy/sell volumes of the market players affect the future prices, we decided to add a new

ISBN:978-988-17012-3-7

term to show these effects in our price prediction models as follows:

$$augmented\ term = \gamma \times W \times a \times Price\_vector \quad (2)$$

where:

γ: is a weighting coefficient that regulates the impact of the augmented term on the models. When **γ** is large the augmented term makes the model deviate from the trend of the time-series market historical data. Therefore, one should be careful in choosing the **γ** factor.

W: is a weight vector that its elements show each company's stock trade impact on future prices. The elements of this vector are between 0 and 1.

a: is the action vector of all players. Its elements are between -1 and 1 that show the buy/sell rates of the stocks. The negative elements depict selling and the positive ones indicate buying the stocks.

Price_vector: contains the current stock price values in the market.

The best value for the **γ** factor obtained to be 0.1. The W-vector was chosen as follows: **W** = [0.1 0.05 0.1 0.2 0.2 0.2 0.05 0.1]. This corresponding companies' symbol vector are: [AMD ERIC IBM INTC MSFT NOK PHG SNE].

The augmented term makes it possible for us to see the effect of each player's market decision on the stock prices and other players' wealth (similar to a non-cooperative game).

Our objective in the next section would be to find the best market actions (sell/buy of each stock) of each player so as to maximize its expected objective function (wealth) in the market. Our market simulation studies are done in a Virtual Stock Market and by means of an evolutionary optimization algorithm (the Particle Swarm Optimization (PSO) method). In our simulations a common PSO with inertia was used. Table1 shows the parameters used in the PSO optimization.

III. Virtual Stock Market

We assume five players (agents) in the stock market. We also assume that these players have no stocks at the beginning of the market. They just have 5,000,000 USD and intend to buy stocks that would maximize their expected wealth in the market. The players are free to buy and sell stocks in each round of the market. There are 1,000,000 stocks assumed to be available from each company in our virtual stock market (4,000,000 stocks in total). The only limitation imposed by the market is the maximum number of stocks each player can buy or sell each day. This buy/sell volume is limited to 1000 stocks trading per day for each company. This constraint is essential because if there is no limitation the whole stocks might be bought at the beginning of the trading period by one of the players. This way there will be no chance for other players to buy some of the stocks. Through the augmented term added to the stock price models we can see the effect of each agent's action (sell/buy stocks) on the future prices and other agents' wealth in the market.

Table 1- The PSO model parameters

Parameter Range	[-1, 1]
Maximum optimization iterations each day	200
Population size	180
Acceleration constant 1	2
Acceleration constant 2	2
Initial inertia weight	0.9
Final inertia weight	0.4
Minimum error gradient	1×10^{-25}
Epochs before error gradient termination	15

We assume five players (agents) with different objective functions and different strategies in the market, but we assume that all the agents have access to the stock price models (developed in section 2) symmetrically.

The players' strategies are as follows:

Strategy of player 1:

This player buys the maximum number of allowed stocks when the prediction shows an increase in next 30 day prices compared to the average prices of the last 10 days. Also it sells the maximum number of allowed stocks when there is a decrease in next 30 day prices compared to the average prices of the last 10 days.

Strategy of player 2:

This player uses the Mean-Variance Analysis (MVA). He chooses the standard deviation of the expected return (r_p) as a measure of risk (σ_p). He plots the opportunity set (efficient frontier) for a four-asset portfolio and takes an average risk and for an average return each day. A sample opportunity set for a four-asset portfolio is shown in figure 3.

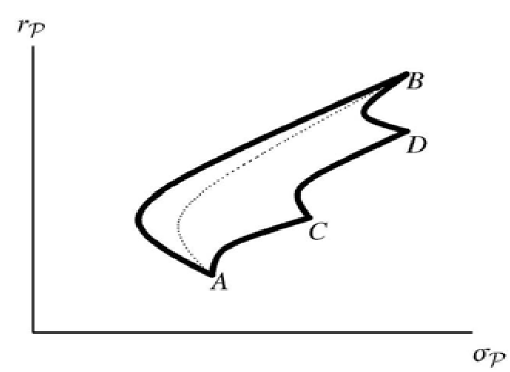

Fig. 3 – A sample opportunity set for a four-asset portfolio (dotted line: The opportunity set with A and B assets only)

Strategy of player 3:

This player believes in *Random Walk Theory*. He believes that the stock prices are unpredictable and therefore, he buys and sells stocks randomly.

Strategy of player 4:

This player acts just like player 2. The only difference is in his risk behavior. This player is risk averse and therefore, in each stage plots the efficient frontier of the four-asset portfolio and then selects the buy/sell weights on the knee of this curve. Therefore, he selects the minimum risk with minimum expected return.

Strategy of player 5:

This player also acts like player 2 with the difference that this player is a risk lover. Therefore, in each stage this player plots the efficient frontier of the four-asset portfolio and then selects the buy/sell weights with the maximum risk and maximum expected return.

The working regions of players 2, 4 and 5 on the risk-return efficient frontier are shown in figure 4.

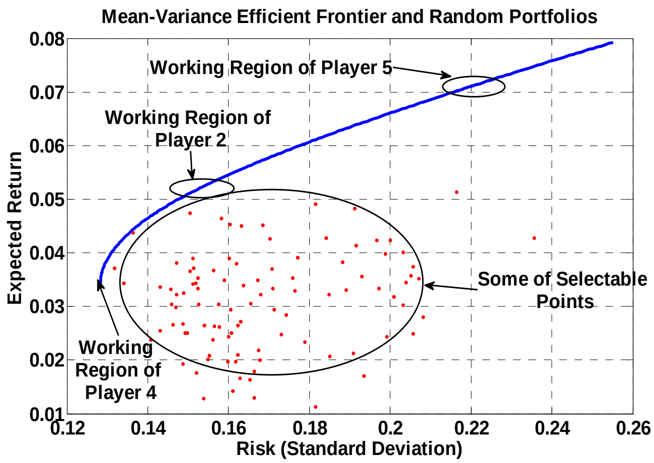

Fig. 4 – The working regions of players 2, 4 and 5 on the risk-return efficient frontier (the red points can be selected in each trading day)

For more information on Modern Portfolio Theory and Mean Variance Analysis (MVA) refer to [25].

These five players buy and sell in the virtual stock market. In the related literature it is usually seen that short-sales are disregarded when optimizing the players' objective functions and the optimization is just done through stock purchases. However, in this paper we have relaxed this constraint and allowed the players to buy and sell their stocks when needed.

As stated before, the players try to maximize their wealth in each trading day, using the 30-day-ahead price predictive models. In the following, we define the objective functions for all players and demonstrate their optimization process. For players 2, 4 and 5 that the risk values are important in their decisions, we define their objective function as follows:

$$E_i = \lambda E(r_{p_i}) - (1-\lambda)\sigma_{p_i} \qquad , \ i = 2,4,5 \tag{3}$$

where:

E_i: is the Expected return of player *i*.

λ: is a constant between 0 and 1. In fact, this is a weight that shows the relative importance of the expected return ($E(r_p)$) versus the risk (σ_p) of player-*i*.

For λ=1 the risk term disappears form the objective function of player-i: $E_i = E(r_{pi})$.

In our market simulation studies, we chose λ={0, 0.5, 1} for players {2, 4, 5} respectively according to their defined risk behaviors in the market.

The players' objective functions were optimized with respect to their decision variables (stock sell/buy actions) using the Particle Swarm Optimization method and the results are presented and analyzed in the following section.

IV. The Market Simulation Results

The market simulation results for the five players are presented and analyzed in this section.

Figures 5 and 6 show the optimal buy/sell actions for players 1 and 5 for each company's stock (ERIC, IBM, PHG and SNE). The optimal buy/sell actions for players 2, 3 and 4 are shown in the appendix. In these figures, the buy actions are positive, sell actions are negative and no-actions are zero.

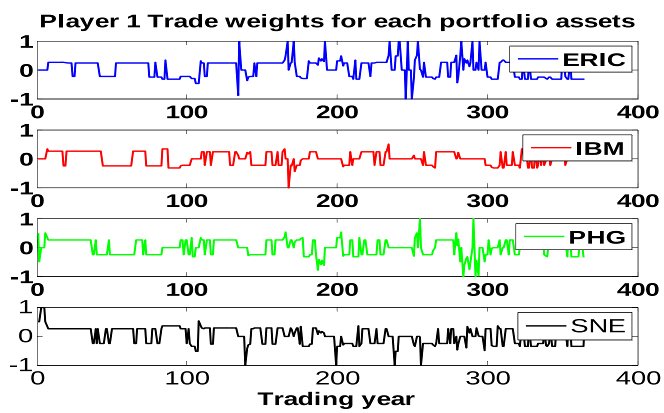

Fig. 5 – The optimal trading actions of player 1 for all companies' stocks

If the buy action gets +1, then the player should buy the maximum number of stocks allowed for that company and when the sell action gets -1, it should sell the maximum number of stocks allowed for that company. In figure 7, the wealth of each player is shown for one trading year period. The wealth is measured as the values of the in-hand stocks added to the cash amount in-hand for each player.

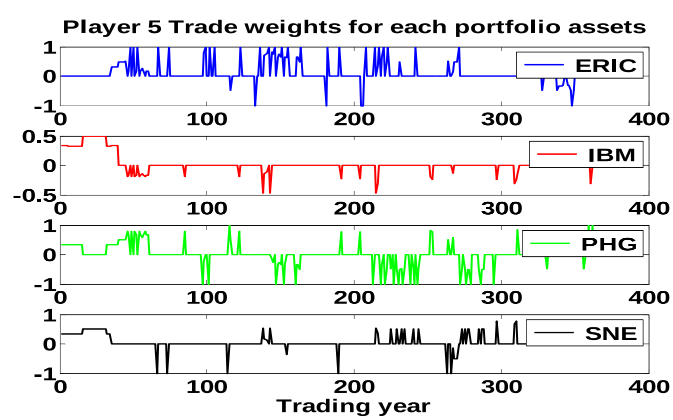

Fig. 6 – The optimal trading actions of player 5 for all companies' stocks

ISBN:978-988-17012-3-7

As can be seen from figure 7, players 1 and 5 have done better than the rest of them in terms of the wealth maximization for one year stock trading. In Table 2, the average wealth of each player for the trading year is shown. Figure 8 shows the expected risk values in each trading day for each player. As we expected, player 1 has the minimum expected risk over the trading period and has also obtained the maximum return from the market.

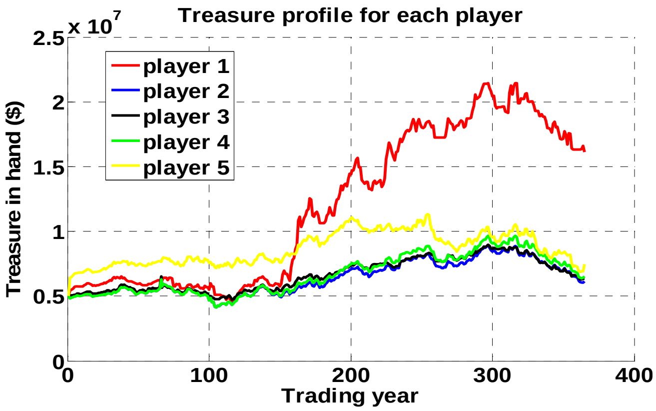

Fig. 7 – The wealth of all players for all days of the trading year

Its strategy was to buy/sell maximum stocks with respect to the comparison of the predicted future-prices' trends with those of the moving average 10-day-before prices. Since the GP prediction models had small prediction errors for the test data, this player did well in the market by relying on the prediction results.

Among players 2, 4 and 5 (by referring to figure 7), we can see that player 5 with the maximum risk level has made the most wealth (expected returns) and stands in the second rank (after player 1) in terms of market returns. Player 3's strategy was to buy and sell randomly; by referring to figures 7 and 8, one can see that his expected returns are similar to those for player 2 but, his expected risks values are more than other players.

TABLE 2 – THE AVERAGE WEALTH OF EACH PLAYER DURING THE ONE YEAR TRADING PERIOD IN THE VIRTUAL STOCK MARKET (IN MILLION DOLLARS)

Player1	Player2	Player3	Player4	Player5
11.95	6.388	6.626	6.659	8.651

V. CONCLUSION

In this paper we first obtained precise price predictive models for four companies' stocks using the Genetic Programming. This model incorporated the effects of the players' actions (sell/buy) on the stock prices and other players' wealth. After the GP model was verified (using the test data from the Boston Market), it was used for making sell/buy decisions by five agents that traded in a virtual stock market. The trading period was considered one year for our market simulation studies. Five different strategies and objective functions were defined for the market trading agents (with different risk attitudes).

The PSO algorithm was used (as an evolutionary optimization method) to obtain the optimal buy/sell actions

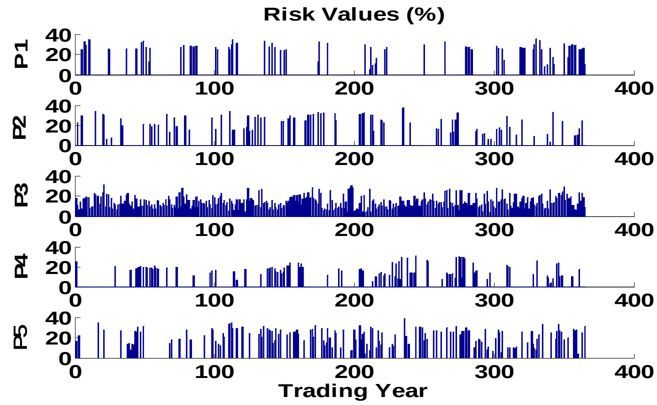

Fig. 8 – The expected risk values for each player over the one trading year (P1 to P5 from top to bottom graphs)

for each player in order to maximize their objective functions (expected returns).The players' strategies and their expected risk-returns were obtained and analyzed for the one year trading period.

Our market simulation studies showed that the player (P1) who made his buy/sell decisions based on the GP model future price trends (compared to the 10-day-before moving average prices) and traded the maximum number of stocks in each trading day was the most successful one in our virtual stock market.

REFERENCES

[1] R. Thalheimer, M. M. Ali, “Time Series Analysis and Portfolio Selection: An Application to Mutual Savings Banks”, *Southern Economic Journal*, Vol. 45, No. 3, pp. 821-837, Jan. 1979.

[2] M. Pojarliev and Wolfgang Polasek, “Applying multivariate time series forecasts for active portfolio management”, *Financial Markets and Portfolio Management*, Vol. 15, pp. 201-211.

[3] D. S. Poskitt and A. R. Tremayne, “Determining a portfolio of linear time series models”, *Biometrika*, Vol. 74, No. 1, pp.125-137, 1987.

[4] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*. San Diego, CA : Academic Press, 1995.

[5] J. W. Weibull, *Evolutionary Game Theory*. London: MIT Press, 1995.

[6] D. Fudenberg, J. Tirole, *Game Theory*. Cambridge, MA: MIT Press, 1991.

[7] S. Lakshminarayanan,”An Integrated stock market forcasting model using neural network”, M. Sc. dissertation, College of Engineering and Technology of Ohio University, August 2005.

[8] K. J. Kim, W. B. Lee, “Stock market prediction using artificial neural networks with optimal feature transformation”, *journal of Neural Computing & Applications*, *Springer*, pp 255-260, 2004.

[9] K. Schierholt, C. H. Dagli, ”Stock market prediction using different neural network classification architectures”, *Computational Intelligence for Financial Engineering*, Proceedings of the IEEEIAFE Conference, pp. 72-78, 1996.

[10] K. Nygren, “Stock Prediction – A Neural Network Approach”, M. Sc. dissertation, Royal Institute of Technology, KTH, March 2004.

[11] E. P. K. Tsang, J. Li, and J. M. Butler, “EDDIE beats the bookies”, *International Journal of Software, Practice and Experience*, Vol. 28, No. 10, pp. 1033–1043, 1998.

[12] D. S. Barr and G. Mani, “Using neural nets to manage investments”, *AI EXPERT*, Vol. 34, No. 3, pp. 16–22, 1994.

[13] R. N. Mantegna and H. E. Stanley, “An introduction to econophysics: Correlation and Complexity in Finance”, Cambridge Univ. Press, Cambridge, MA, 2000.

[14] J. P. Bouchaud and M. Potters, “Theory of Financial Risks: From Statistical Physics to Risk management”, Cambridge Univ. Press, Cambridge, MA, 2000.

[15] Y. K. Kwon, S. S. Choi, B. R. Moon, "Stock Prediction Based on Financial Correlation", *In Proceedings of GECCO'2005*, pp. 2061-2066, 2005.

[16] Y. K. Kwon and B. R. Moon, "Daily stock prediction using neuro-genetic hybrids", *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 2203–2214, 2003.

[17] Y. K. Kwon and B. R. Moon, "Evolutionary ensemble for stock prediction", *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1102–1113, 2004.

[18] F. Parisi, A. Vasquez, "Simple technical trading rules of stock returns: evidence from 1987 to 1998 in Chile", *Emerging Market Review*, vol. 1, pp. 152–64, 2000.

[19] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Evolution*. Cambridge, MA: MIT Press, 1992.

[20] J. Madar, J. Abonyi, and F. Szeifert, "Genetic Programming for the Identification of Nonlinear Input-Output Models", *Ind. Eng. Chem. Res.*, Vol. 44, pp. 3178-3186, 2005.

[21] M. A. Kaboudan, "Genetic Programming Prediction of Stock Prices", *Computational Economics*, Vol. 16, No. 3, pp. 207 – 236, 2000.

[22] S. Sette, L. Boullart, "Genetic Programming: principles and applications", *Eng. Appl. Artif. Intell.*, vol. 14, pp. 727-736, 2001.

[23] Boston Stock Group web page: http://boston.stockgroup.com

[24] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942-1945, 1995.

[25] D. Maringer, *Portfolio Management with Heuristic Optimization*. Springer, 2005.

VI. Appendix

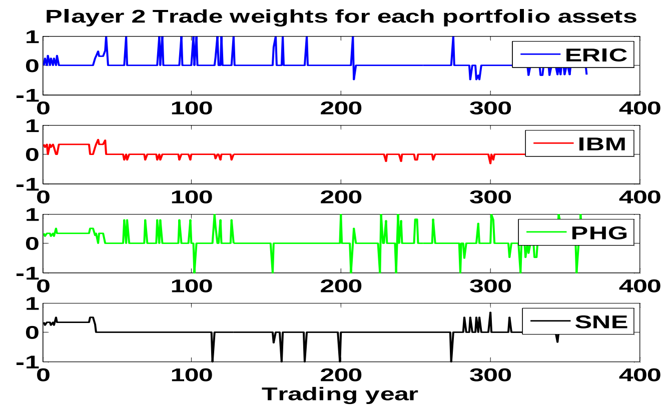

Fig. 1 – The optimal trading actions of player 2 for all companies' stocks

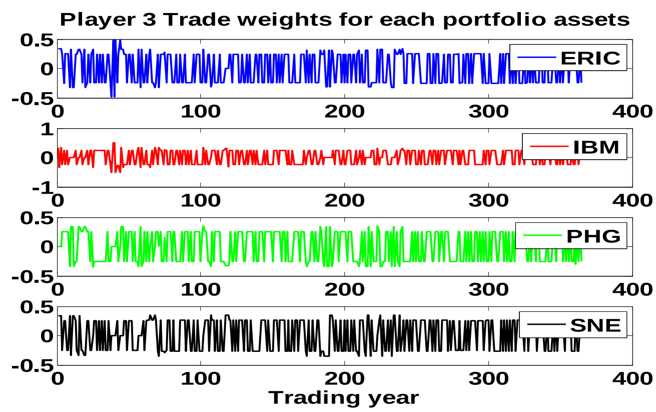

Fig. 2 – The optimal trading actions of player 3 for all companies' stocks

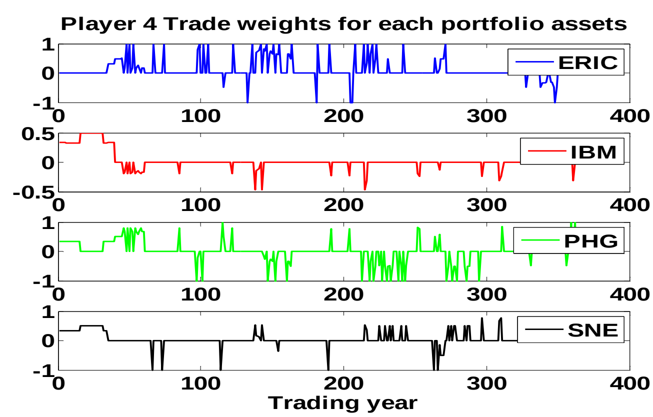

Fig. 3 – The optimal trading actions of player 4 for all companies' stocks

ISBN:978-988-17012-3-7