"Dead" chromosomes and their elimination in the neuro-genetic stock index prediction system

Jacek Mańdziuk[1] and Marcin Jaruszewicz[2]

[1] Warsaw University of Technology, Faculty of Mathematics and Information Science, Plac Politechniki 1, 00-661 Warsaw, POLAND, mandziuk@mini.pw.edu.pl, WWW home page: http://www.mini.pw.edu.pl/~mandziuk/
[2] jaruszewicz@data.pl

Abstract. This paper presents a method for a short-term stock index prediction. The source data comes from the German Stock Exchange (being the target market) and two other markets (Tokyo Stock Exchange and New York Stock Exchange) together with EUR/USD and USD/JPY exchange rates. Neural networks supported by a genetic algorithm (GA) are used as the prediction engine. Except for promising numerical results attained by the system the special focus in the paper is on the problem of elimination of *dead* chromosomes, i.e. the ones which cannot be properly assessed.

1 Introduction

Computational Intelligence (CI) methods are widely used for stock market prediction. Most of them, however, require specific set up of steering parameters and suitable selection of the input data. Both these issues can be effectively approached with the help of genetic algorithms (GAs). For example, in [1] the authors propose a system for stock market prediction relying on neural networks and fundamental analysis, where the selection of variables is performed by the GA. Alternatively the topology of the network or its internal parameters can be determined by the GA [2]. GAs are also applied to the task of generation of optimal prediction rules/models [3, 4] or transformation of the knowledge hidden inside neural network's weights to a self-explanatory set of rules [5].

The prediction system discussed in this paper is a suitable combination of the computational power of neural networks with flexibility and optimization capabilities of genetic algorithms. The goal is to predict the closing value of German Stock Exchange (GSE) index DAX for the next day. The approach relies solely on technical analysis data. Except for the target market the data from two other international markets namely American New York Stock Exchange (NYSE) with DJIA index and Tokyo Stock Exchange (TSE) with NIKKEI 225 (NIKKEI) index is also considered, together with exchange rates of EUR/USD and USD/JPY.

The most suitable set of input variables is chosen by the GA and validated by a simplified neural network training and testing procedure. Due to high volatility of mutual dependencies between input variables the set chosen by the GA is only used for 5 trading days. After this period a new set of variables is chosen for the next 5 trading days, and so on.

Proposed solution combines the power of the GA in a wide and parallel search with the intrinsic ability of neural networks to discover relations within input data. The system works without human intervention choosing desired input variables autonomously from a large number of possibilities. In independent prediction steps variables are selected with noticeable sense. The existence of characteristic, repeatable patterns of variables within subsequent 5-day windows can be observed.

Whilst the GA was probing different sets of variables several interesting observations were made. For example, the existence of chromosomes in a population, that coded neural networks unable to learn was discovered. Such chromosomes were called *dead*. They were present especially in early iterations of the GA. With appropriate choice of mutation conditions the algorithm tended to consequently eliminate dead chromosomes.

The reminder of this paper is organized as follows: in the next section an overview of the system is presented, section 3 summarizes experimental results published in part in previous papers and section 4 presents and discusses the problem of dead chromosomes in the population. Conclusions are placed in the last section.

2 An overview of the system

The first step of the prediction algorithm is initial data pre-selection. It starts with analysis of plain values of an index, proceeds with defining averages and oscillators, and ends with application of pattern extraction techniques. Since the goal is defined as a one-day prediction, fundamental analysis is not very helpful [6]. Instead, the main focus is on technical analysis and its indicators. Another possibility would be the use of portfolio management methods, which also proved to be efficient in short-term prediction [4, 7]. The source data is the opening, highest, lowest and closing values of the index of the selected stock markets in subsequent days. This data serves as the basis for further transformations, e.g. the percentage change of opening value through the last 5, 10 and 20 days, the last percentage change of closing value or averages of opening values through the last 5, 10 and 20 days.

The above variables are intuitively useful for prediction. They provide simple presentation of past values in a compressed way. Especially moving averages allow the algorithm to omit sudden local changes when looking for prediction rules. The predefined periods for averages and changes (5, 10, 20 days) were chosen based on preliminary simulations and suggestions from the literature [8]. Generally speaking these periods need to be adjusted to the assumed prediction horizon (which is equal to one day in this paper). The same time periods are also used in more complex transformations, e.g. in definitions of the oscillators.

A chart of index values with corresponding changes in the period April 7, 2004 – August 26, 2004 is presented in Fig. 1. Sudden changes in different directions are noticeable. On the other hand, general rises and falls of trend are also clearly visible.

On the same basis the oscillators, known in technical analysis, are calculated [6]. The following oscillators are chosen in our experiment: MACD; Williams; Two Averages; Impet; RSI; ROC; SO and FSO. Except for raw values, the buy/sell signals generated by the oscillators are also considered.

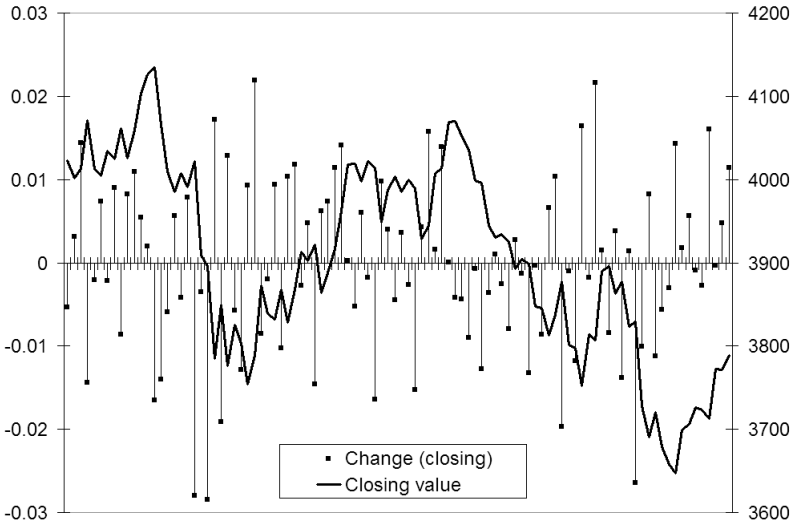

Fig. 1. Closing DAX value and its changes in the period 2004/04/07 to 2004/08/26.

After the pre-processing step a large number of variables are available for the algorithm's selection. A method of selecting sub-optimal, in the current situation, set of variables and a neural network-based predictor which uses them as the input are described below. The final selection of the input variables is performed by the GA. Each chromosome codes a set of input variables to be used by a neural network. A fitness function is reverse-proportional to the magnitude of an error made by a trained network on a sample test set. After the GA finishes its work the best fitted chromosome provides a suitable set of variables to be used in the final training. These variables are considered to be a sub-optimal selection for the current 5-day window.

Neural networks form the basis of the proposed prediction system. Feed-forward networks with one hidden layer are chosen based on preliminary experiments. The size of the hidden layer equals the *floor* function of the half of the size of an input layer (e.g. in case of 9 input neurons, the hidden layer is composed of 4 units). One neuron is used in the output layer. Therefore the architecture of each network used in the system during the GA phase or for the final prediction is defined completely by the number of input variables. The weights of such a network are randomly initialized. The method of back propagation with momentum is used for training.

The neural network training process is efficient if the size of the network is relatively small. This goal is typically achieved by adaptation of network's topology [2, 9–11].

Another possibility of optimization is efficient selection of input variables. In order to select small, but relevant number of variables for a neural network the GA is applied. Due to a large number of available input variables and changing dependencies between them, a selection of a suitable subset of inputs is the main idea of the proposed system. The quality of prediction strictly depends on the effectiveness of the GA. The chromosome is defined as the list of variables used by a neural network for training

and prediction. Therefore each chromosome defines a specific network's architecture, as described above. For example if a chromosome codes the following set of variables: closing value of DAX, average change in 20 days, MACD oscillator value, percentage change in 20 days, then the network defined by this chromosome has 4 input neurons, 2 neurons in a hidden layer and 1 output neuron. For each input neuron the value of the respective variable is provided. The fitness function is reverse-proportional to the error of a network after training. The smaller the error on validation samples, the higher the fitness of a chromosome. For each chromosome several neural networks with different initial weights, but the same architecture are considered. The fitness function can be calculated with respect to either the minimum or the average error of all networks assigned to the chromosome. If the average error is used the algorithm is less sensitive to accidental increase or decrease of the learning process effectiveness caused by the initial weights choice.

There are two main operators in the GA. A crossover is responsible for creating new chromosomes using information coded by both parents. Mutation brings random changes in chromosomes, which expand their capabilities of searching for the result in the whole solution space.

Two crossover methods are implemented in the proposed system. The first one is classical crossover with one cutting point. The cutting point can by selected randomly or chosen according to the best fitness of the resultant child chromosome (see fig. 2). The alternative crossover method, a combined crossover, depends on the common part of two parent chromosomes and random selection of the rest of variables. The scheme of the combined crossover is presented in fig. 3. This kind of crossover promotes variables that are repeated in the population.

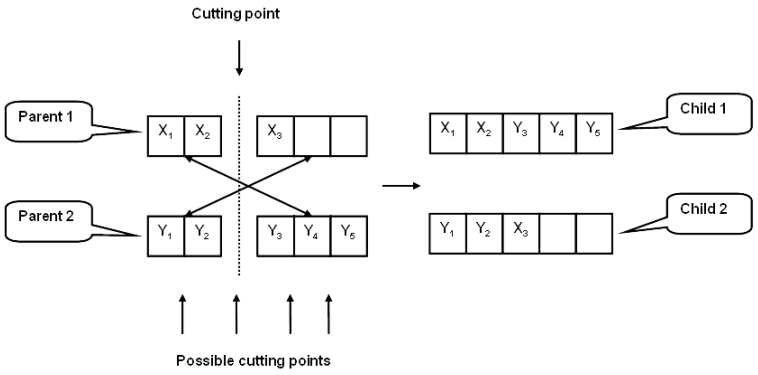

Fig. 2. The scheme of the classical crossover operation with one cutting point.

In any case, two offspring are created as a result. Selection of chromosomes for crossover is performed by the rank method. The probability of crossover equals 1, however, a parent is exchanged with assigned child only if the latter is better fitted. Using

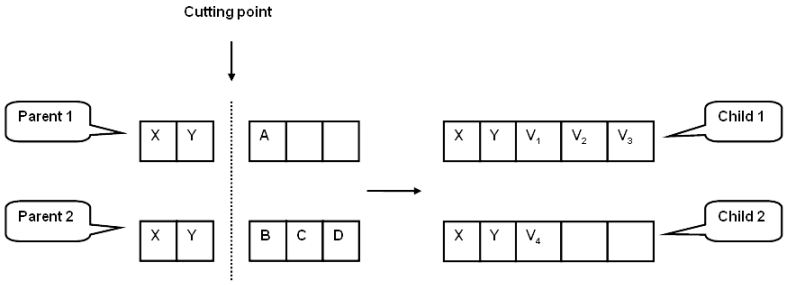

Fig. 3. The scheme of the combined crossover operation with a random selection of variables. V_1, V_2, V_3, V_4 are randomly selected from the remaining pool of variables (i.e. except for X and Y) with additional condition that they are pairwise different within each of the chromosomes, i.e. $V_2 \neq V_3, V_3 \neq V_4$ and $V_2 \neq V_4$.

the above described crossover does not alter population size. Sizes of individual chromosomes generally change towards the preferred/winning ones.

During mutation variables coded by the chromosome are exchanged with variables chosen randomly from the remaining subset of all available ones. A single mutation can affect only one randomly selected variable. The probability of mutation depends on the type of the chromosome (dead or alive) - see section 4.

The first step of GA is creation of the initial population. Variables coded by chromosomes are randomly selected from the initial pool. Diversity of the population is assured by creating chromosomes of different sizes (numbers of coded variables). All chromosomes' sizes from a predefined range are represented in the initial population [12].

In each iteration of the GA new chromosomes are created as a result of crossover and mutation. Gradually the average error of prediction made by neural networks coded by the chromosomes decreases.

After every iteration the best chromosome is appointed. A chromosome can be considered the best one only if all neural networks used in the fitness evaluation are able to learn (is alive for every network). Finally, in the last iteration of the GA the network with the lowest error for the best chromosome is saved and subsequently used in the final test.

3 Summary of results

In order to examine the effectiveness and usefulness of proposed system of prediction several simple prediction models were implemented for comparison. All of them are based on generating buy/sell signals to make the maximum profit of transactions on index value. In the initial state there is a fixed budget for transactions. At the end of the day the signal for the next day is generated. Transaction is done with the next day opening value. If the signal is "buy" as many indices as possible with the current budget are bought. Remaining money (the amount smaller than the cost of one index) becomes the budget for the next day. If the "sell" signal is provided all possessed indices are sold.

The total budget is available for future transactions. The final result is the total budget after selling all indices at the last day.

Model 1 - *"buy and hold"* strategy: the first day's signal is *"buy"*, the last day's one is *"sell"*, and there are no other buy/sell signals generated. As a result the profit equals the change of an index value in a given time period.

Model 2 - signals are generated assuming that the direction of change of an index value of the next day will be the same as the last change direction.

Model 3 - signals are calculated using the next day prediction of a neural network - *this model*

Model 4 - signals are calculated using the MACD oscillator. Signals are generated at the crossing of lines of the oscillator [6].

Model 5 - signals are generated using the actual knowledge of the next day's value. This, prothetic and omnipotent, model gives the best possible result under defined conditions.

Models 1 and 2 are constructed using simple heuristics with no particular prediction system. Model 4 is based on the very popular MACD oscillator. Model 5 defines the upper limit of a profit that can be achieved assuming that the exact knowledge of future index values is available beforehand.

A single experiment is divided into steps. Each step consists of neural network training on 290 samples and 5 validation records. The number of training samples is chosen to cover a year of observations. On the other hand it is a reasonable number of samples for training the average-size neural network used in the proposed system. The neural network selected in the current step is tested on the subsequent 5 records not used in previous phases. In the next step the process is repeated using samples shifted by 5 records. Large part of the data is shared between steps since in subsequent ones the time-window is shifted by 5 days forward (test samples from immediately previous step become validation ones and validation samples become the last part of the training days).

Four different experiments (one for each model except for the buy and hold one), each consisting of 20 steps were performed in order to present usefulness of the above approach. Consequently, in each experiment the consistent set of 100 (20×5) samples was considered as the test data (covering the period from 2004/04/07 to 2004/08/26). In each step the following sequence of activities was performed: **A:** creating initial pool of variables, **B:** finding the best chromosome by the GA, and **C:** performing the final training with neural network and input variables coded by that chromosome.

The basic GA parameters were defined based on some preliminary tests. The population size was equal to 48, the number of generations was equal to chromosomes 500, inial sizes of the chromosomes were between 4 and 11, the number of neural networks used in each chromosome for fitness calculation was set to 3 and the number of iterations during neural network learning for fitness calculation in GA was equal to 200.

Results of experiments performed according to the above described scenarios are summarized in Table 1. In each experiment the amount of initial money was equal to $100,000$ units. Please recall, that Model 5 assumes the exact knowledge about the future. The result of Model 1 (which implements *buy and hold* strategy) is proportional to the change of index value in the considered period of time. Furthermore the repeatability

Model/Experiment	Return [%]
Model 1	-5.46
Model 2	-7.54
Model 3	9.31
Model 4	-3.59
Model 5	39.56

Table 1. Return of proposed models attained in different experiments.

of results was tested by running several experiments in the same time period and under the same experiment's settings except for initial neural networks' weights. The results are presented in Table 2. Moreover, a detailed analysis of the sets of variables chosen is

$\#1$	$\#2$	$\#3$	$\#4$	$\#5$	$\#6$	**avg.**
9.31	4.68	3.51	3.41	5.96	7.67	5.44

Table 2. Return $[\%]$ of proposed model attained in six experiments performed with the same algorithm settings and for the same period of time.

subsequent steps of the experiments was performed, proving their "visible sense". The results confirm the hypothesis about changing in time dependencies between variables. The frequency analysis of choosing variables from three different stock markets to the current input set proved the hypothesis that stock markets are related and that the influence of NYSE on the GSE (generally, on European stock markets) is prevailing the one of Japanese TSE. These variables were also more preferable than the exchange rates. The analysis of the algorithm's variable selection process is presented in [12] and [13].

4 *Dead* and *alive* chromosomes

One of the interesting conclusions drawn from the analysis of experimental data was the existence of specific sets of input variables for which the neural networks were unable to achieve acceptable level of an error. This usually happened in the initial population of chromosomes, but sometimes also in further generations, when some sets of the variables were not suitable for learning. Chromosomes which coded such useless sets were called *dead*. The remaining chromosomes were called *alive*. More precisely, a chromosome was defined as dead if and only if none of the three neural networks defining chromosome's fitness was unable to learn, i.e. achieved unsatisfactory test result on validation records (not used in the training).

Since chromosomes in the initial population are selected randomly (within some predefined ranges) the number of dead individuals at the beginning of the GA process can be relatively high. Despite the initial number of them, the system tends to successfully eliminate such useless chromosomes.

Two typical situations are presented in fig. 4, where two populations from two different steps of the same experiment are described. Each population consists of 48 individ-

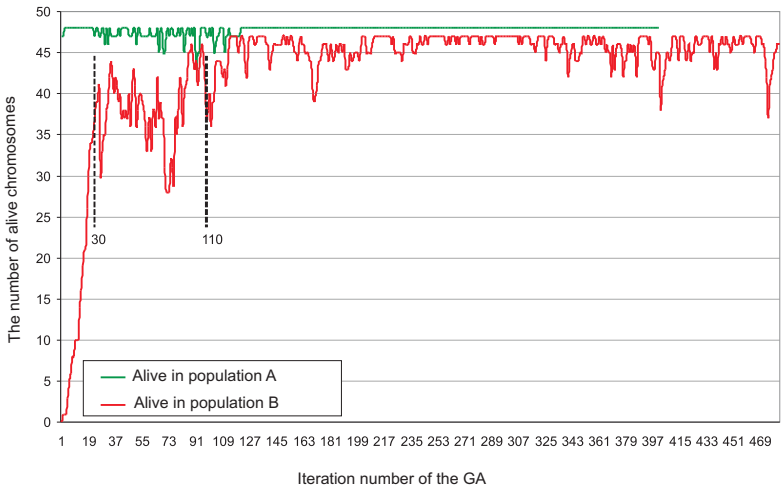

Fig. 4. The numbers of alive chromosomes in two exemplar populations. See description within the text.

uals. All algorithm's parameters are the same for both steps, except for different 5-day time periods covered by each step. In the population A the number of dead chromosomes in the initial generation is relatively small and this situation does not drastically change until the final generation (though some fluctuations in the initial phase can be observed). The initial situation in population B looks quite differently. Here the number of dead chromosomes is high (actually only some of them are alive), but the ratio between alive and dead individuals is quite rapidly improving in the initial period of GA. After 30 generations this ratio becomes relatively high and from the 110th generation the number of alive chromosomes in the population stabilizes at the level of about 95%.

In order to alleviate the negative influence of dead chromosomes additional constraints were imposed on the mutation operator, whose probability depended on the type of a chromosome. Generally speaking, probability for dead chromosomes was much higher than for alive ones. This setting forced the undesirable chromosomes to be changed. Changes in alive chromosomes were rare - just enough to keep the diversity of the population. In particular, the probability of mutation of a dead chromosome was set to 1 in the whole experiment. The respective probability of alive chromosomes was higher in the initial phase of the GA (equal to 0.2) and lowered subsequently to 0.05. Furthermore, the probability of mutation of alive chromosomes was automatically increased (from 0.05 to 0.2) if the difference between the maximum fitness in the population and the average fitness was below some predefined threshold. Additionally, mutations of alive chromosomes were only allowed if the number of such chromosomes exceeded 90% or was less than 50% of the population size. In the remaining interval ($[50\%, 90\%]$) only dead chromosomes were allowed to be mutated.

The above steering parameters were crucial for proper maintenance of the population, especially for keeping the number of alive individuals at a high level. On the other

hand some amount of dead chromosomes was also allowed to exist in the population in order to keep it divers and having potential for further development.

5 Conclusions and future work

A hybrid neuro-genetic method of prediction with application to financial task, which is prediction of the closing value of DAX - the German Stock Exchange index, is presented and examined in the paper. Assuming the flow of information between different stock markets and their mutual relations the proposed system uses data from five different sources: three stock markets and two exchange rates.

Due to changing dependencies between variables describing financial markets (indices, oscillators, exchange rates, etc.) and assuming that the usefulness of any variable is limited in time, the proposed approach relies on applying the GA for *frequent* input data selection (among the predefined pool of variables from the above mentioned five sources).

The standard GA procedure is enhanced by adding a new type of crossover operator and a mechanism that controls the range of chromosomes' sizes. Furthermore, the chromosomes are categorized as being either *alive* (i.e. the neural net architectures coded by them are able to learn) or *dead* (in the opposite case). Based on the ratio between alive and dead chromosomes the probabilities of genetic operators are changing accordingly in order to get rid of useless, non-alive chromosomes.

In future work we plan to apply a systematic procedure for selection system's internal parameters (population size, the ranges within which the dead chromosomes are allowed to be mutated, etc.). In the current version of the algorithm the above choices are the result of non-exhaustive trial and error procedure. Also more numerical evaluations are planned in order to further validate the efficacy of proposed neuro-evolutionary method.

References

1. Thawornwong, S., Enke, D.: The adaptive selection of financial and economic variables for use with artificial neural networks. Neurocomputing **56** (2004) 205–232
2. Fieldsend, J., Singh, S.: Pareto evolutionary neural networks. IEEE Transactions on Neural Networks **16**(2) (2005) 338–354
3. Dempster, M., Payne, T., Romahi, Y., Thompson, G.: Computational learning techniques for intraday FX trading using popular technical indicators. IEEE Transactions on Neural Networks **12**(4) (2001) 744–754
4. John, G., Miller, P., Kerber, R.: Stock selection using rule induction. IEEE Expert (1996) 52–58
5. Duch, W., Setiono, R., Zurada, J.: Computational intelligence methods for rule-based data understanding. Proceedings of the IEEE **92** (2004) 771–805
6. Murphy, J.: Technical Analysis of the Financial Markets. New York Instituite of Finance (1999)
7. Cover, T.: An algorithm for maximizing expected log investment return. IEEE Transactions on Information Theory **30**(2) (1984) 369–373
8. Schwager, J.D.: Getting started in Technical Analysis. John Wiley & Sons (1999)

9. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In Touretzky, D., ed.: Advances in Nueral Information Processing Systems 2, San Mateo, CA, USA, Morgan Kaufmann (1990) 524–532
10. LeCun, Y., Denker, J., Solla, S.: Optimal brain damage. In Touretzky, D., ed.: Advances in Nueral Information Processing Systems 2, San Mateo, CA, USA, Morgan Kaufmann (1990) 598–605
11. Hassibi, B., Stork, D.: Secong order derivatives for network pruning: Optimal Brain Surgeon. In Touretzky, D., ed.: Advances in Nueral Information Processing Systems 5, San Mateo, CA, USA, Morgan Kaufmann (1993) 164–171
12. Mańdziuk, J., Jaruszewicz, M.: Neuro-evolutionary approach to stock market prediction. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN07), Orlanodo, FL, USA, IEEE Press (2007) 2515–2520
13. Jaruszewicz, M., Mańdziuk, J.: Neuro-genetic system for DAX index prediction. A. Cared and L. Rutkowski and R. Tadeusiewicz and J. Zurada (Ed.), Artificial Intelligence and Soft Computing (2006) 42–49