# Predicting Turning Points in Financial Markets with Fuzzy-Evolutionary and Neuro-Evolutionary Modeling

Antonia Azzini, Célia da Costa Pereira, and Andrea G.B. Tettamanzi

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
via Bramante 65, I-26013 Crema, Italy
azzini,pereira,tettamanzi@dti.unimi.it

**Abstract.** Two independent evolutionary modeling methods, based on fuzzy logic and neural networks respectively, are applied to predicting trend reversals in financial time series, and their performances are compared. Both methods are found to give essentially the same results, indicating that trend reversals are partially predictable.

## 1 Introduction

Even the most casual observer of a financial time series will notice that prices of financial instruments move up and down [8]; furthermore, this behavior happens and can be observed at all scales [9]. However, price movements are not regular and look unpredictable.

In general, market action consists of alternating up-trends and down-trends, separated by turning points, which correspond to maxima and minima. A trader able to buy at minima and sell at maxima, i.e., trade exactly at the turning points, would gain the maximum profit possible. For this reason, the main objective of financial market forecasting techniques is to call turning points consistently and correctly. Two approaches for calling turning points in a financial time series are possible:

- reveal turning points when they occur, or just after they have occurred — this is what most technical analysis indicators are all about, starting from simple moving averages to the most sophisticated indicators;
- predict the price at which the next turning point will most likely occur — this is the approach we follow in this paper.

To do that, we summarize the past history of the series up to the last confirmed turning point by applying a noise-eliminating filter. The output of the filter is given as input to a predictive model, whose output provides an estimate of the price at which the next turning point is going to happen.

Biologically inspired methods have become extremely popular as tools for modeling financial markets [5, 6]; these include evolutionary algorithms, neural networks, and fuzzy logic. In this paper we employ two types of models, namely fuzzy rule bases, i.e., sets of fuzzy IF-THEN rules, and feedforward neural networks. Both types of models are designed and optimized by means of evolutionary algorithms. The performance of models of the two types are compared with the aim of assessing which one is more suited to the task.

The paper is organized as follows: Section 2 states the problem, Section 3 provides a brief description of the two modeling methods, and Section 4 reports the experiments and discusses their results. Section 5 concludes.

## 2 Problem Description

To eliminate noise, the time series of prices of the financial instrument under study is pre-processed by applying the *zig-zag* filter [1] with a threshold $\theta$. The *zig-zag* decomposes the input time series into a series of alternating up- and down-swings. The length of each swing is given by the price difference of its ends. To make this datum adimensional, the length of each swing is divided by the length of the previous swing. The series $\{r_t\}_t$ of the resulting ratios is the output of the noise-eliminating filter.

The problem of predicting the price at which the next turning point will occur can thus be transformed into the problem of predicting the next ratio of the series of ratios of swing lengths, since the price of a turning point $x_{t+1}$ can be calculated by knowing the price of the two previous turning points $x_{t-1}$ and $x_t$ and the ratio $r_t = \frac{|x_{t+1}-x_t|}{|x_t-x_{t-1}|}$.

The working hypothesis is that such prediction can be done at any time by considering the $n$ most recent ratios. In other words, we are searching the space of all autoregressive models of $\{r_t\}_t$ of order $n$.

A baseline for any model is provided by the simplest model possible, namely a model that always predicts $E[r_t]$, without taking the last $n$ ratios into account. $E[r_t]$ can be estimated by taking the longest available time series for the financial instrument considered and computing the average ratio of a swing length to the length of the previous swing. Such a model has a mean absolute error $E[|r_t - E[r_t]|]$ and a mean square error $E[(r_t - E[r_t])^2] = \mathrm{Var}[r_t]$, while the relative mean absolute error corresponds to $E[|r_t - E[r_t]|]/E[r_t]$. This, by the way, is the best performance one would expect from a predictive model if the time series under observation were completely random or, which is roughly equivalent [9], if the market were perfectly efficient.

## 3 Evolutionary Optimization

### 3.1 Fuzzy Rule Base Optimization

Data mining is a process aimed at discovering meaningful correlations, patterns, and trends between large amounts of data collected in a dataset. A model is determined by observing past behavior of a financial instrument and extracting the relevant variables and correlations between the data and the dependent variable. We describe below a data-mining approach based on the use of EAs, which recognize patterns within a dataset, by learning models represented by sets of fuzzy rules.

**Fuzzy Models** A model is described through a set of fuzzy rules, made by one or more antecedent clauses ("IF ...") and a consequent clause ("THEN ..."). Clauses are represented by a pair of indices referring respectively to a variable and to one of its fuzzy sub-domains, i.e., a membership function.

Using fuzzy rules makes it possible to get homogenous predictions for different clusters without imposing a traditional partition based on crisp thresholds, that often do not fit the data, particularly in financial applications. Fuzzy decision rules are useful in approximating non-linear functions because they have a good interpolative power and are intuitive and easily intelligible at the same time. Their characteristics allow the model to give an effective representation of the reality and simultaneously avoid the "black-box" effect of, e.g., neural networks.

The intelligibility of the model is useful for a trader, because understanding the rules helps the user to judge if a model can be trusted.

**The Evolutionary Algorithm** The described approach incorporates an EA for the design and optimization of fuzzy rule-based systems originally developed to learn fuzzy controllers [10, 11], then adapted for data mining, which has already been used for financial modeling by two of the authors [7].

A model is a rule base, whose rules comprise up to four antecedent and one consequent clause each. Input and output variables are partitioned into up to 16 distinct linguistic values each, described by as many membership functions. Membership functions for input variables are trapezoidal, while membership functions for the output variable are triangular. Models are encoded in three main blocks:

1. a set of trapezoidal membership functions for each input variable; a trapezoid is represented by four fixed-point numbers;
2. a set of symmetric triangular membership functions, represented as an area-center pair, for the output variable;
3. a set of rules, where a rule is represented as a list of up to four antecedent clauses (the IF part) and one consequent clause (the THEN part); a clause is represented by a pair of indices, referring, respectively, to a variable and to one of its membership functions.

An island-based distributed EA is used to evolve models. The sequential algorithm executed on every island is a standard generational replacement, elitist EA. Crossover and mutation are never applied to the best individual in the population.

The recombination operator is designed to preserve the syntactic legality of models. A new model is obtained by combining the pieces of two parent models. Each rule of the offspring model can be inherited from one of the parent models with probability 1/2. When inherited, a rule takes with it to the offspring model all the referred domains with their membership functions. Other domains can be inherited from the parents, even if they are not used in the rule set of the child model, to increase the size of the offspring so that their size is roughly the average of its parents' sizes.

Like recombination, mutation produces only legal models, by applying small changes to the various syntactic parts of a fuzzy rulebase.

Migration is responsible for the diffusion of genetic material between populations residing on different islands. At each generation, with a small probability (the migration rate), a copy of the best individual of an island is sent to all connected islands and as many of the worst individuals as the number of connected islands are replaced with an equal number of immigrants.

A detailed description of the algorithm and of its genetic operators can be found in [10].

### 3.2 Neuro Genetic Optimization

The second evolutionary approach [2–4] considered in this work evolves a population of NNs, encoding multilayer perceptrons (MLPs), a type of feed-forward NN. The algorithm is based on the joint optimization of structure and weights, and uses the error back-propagation (BP) algorithm to decode a *genotype* into a *phenotype* NN. Accordingly, it is the genotype which undergoes the genetic operators and which reproduces itself, whereas the phenotype is used *only* for calculating the genotype's fitness. The rationale for this choice is that the alternative of using BP, applied to the genotype, as a kind of "intelligent" mutation operator would boost exploitation while impairing exploration, thus making the algorithm too prone to being trapped in local optima.

The individuals are not constrained to a pre-established topology, and the population is initialized with different hidden layer sizes and different numbers of neurons for each individual according to two exponential distributions, in order to maintain diversity among all of them in the new population. Such dimensions are not bounded in advance, even though the fitness function may penalize large networks. The number of neurons in each hidden layer is constrained to be greater than or equal to the number of network outputs, in order to avoid hourglass structures, whose performance tends to be poor. Indeed, a layer with fewer neurons than the outputs destroys information which later cannot be recovered.

The evolutionary process adopts the convention that a lower fitness means a better NN, mapping the objective function into an error minimization problem. Therefore the fitness is proportionate to the mean square error and to the cost of the considered network.

**Evolutionary Process** In the overall evolutionary process the population is randomly created, initialized, and the genetic operators are then applied to each network until termination conditions are not satisfied.

At each generation, a population consisting of the best $\lfloor n/2 \rfloor$ individuals is selected by truncation from a population of size $n$; the remaining NNs are then duplicated in order to replace those eliminated, and finally, the population is randomly permuted. Elitism allows the survival of the best individual unchanged into the next generation and the solutions to get better over time. Then, for all individuals of the population the algorithm mutates the weights and the topology of the offsprings, trains the resulting network, calculates fitness on the test set, and finally saves the best individual and statistics about the entire evolutionary process.

Weights mutation perturbs the weights of the neurons before performing any structural mutation and applying BP to train the network. All the weights and the corresponding biases are updated by using variance matrices and evolutionary strategies applied to the synapses of each NN, in order to allow a control parameter, like mutation variance, to self-adapt rather than changing their values by some deterministic algorithm. Finally, the topology mutation is implemented with four types of mutation by considering neurons and layer addition and elimination. The addition and the elimination of a layer and the insertion of a neuron are applied with three independent probabilities, while the elimination of a neuron is carried out only if the contribution of that neuron is negligible with respect to the overall network output.

## 4 Experiments and Results

We have considered the time series of daily prices of the S&P500 index from September 26, 1985 to October 31, 2008, filtered with two different values of $\theta$, namely $\theta = 0.01$ and $\theta = 0.02$. For $\theta = 0.01$, $E[r_t] = 1.2918$, $\text{Var}[r_t] = 1.1062$, relative error 0.5586; for $\theta = 0.02$, $E[r_t] = 1.2418$, $\text{Var}[r_t] = 0.6895$, relative error 0.49624. With $n = 12$, the application of the filter produces datasets of, respectively, 2,769 and 1,095 records.

Following the commonly accepted practice of machine learning, the problem data are partitioned into training, test and validation sets, used, respectively for training, to stop learning avoiding overfitting, and to test the generalization capabilities of each model.

We have set aside, for validation, the 200 most recent records for $\theta = 0.01$ and the 100 most recent records for $\theta = 0.02$. Of the remaining records, a random 10% is used as test set by the fuzzy-evolutionary method and 18% and 10% respectively by the neuro-genetic method.

We have performed 20 runs for either dataset with the neuro-genetic method and 10 runs for either dataset with the fuzzy-evolutionary method, which is more demanding in terms of computational resources. The results of those runs are shown in Table 1, that also shows a consistent, if not dramatic, improvement with respect to the baseline model, whose prediction is always $E[r_t]$, for both methods and both datasets.

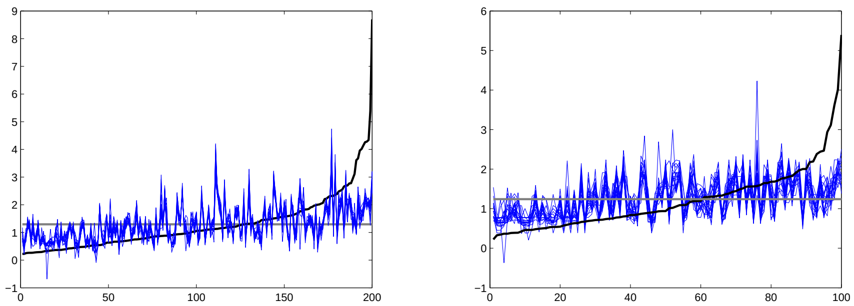
The fact that two independent methods, both based on global optimization methods like evolutionary algorithms, yet using two radically different “languages” to represent models, have obtained almost identical results, can be taken

**Table 1.** A comparison of the results obtained by the two methods on the two datasets generated for $\theta = 0.01$ and $\theta = 0.02$ when applied to the validation set (out of sample).

| Dataset → | $\theta = 0.01$ | | | $\theta = 0.02$ | | |
|---|---|---|---|---|---|---|
| Method | mean | stdev | best | mean | stdev | best |
| Fuzzy-Evolutionary | 0.4720 | 0.0055 | 0.4657 | 0.4110 | 0.0049 | 0.4057 |
| Neuro-Evolutionary | 0.4813 | 0.0096 | 0.4740 | 0.40307 | 0.0250 | 0.3958 |

as a guarantee that there is very little room for further improvement, if any, and their results are very close to the optimum.

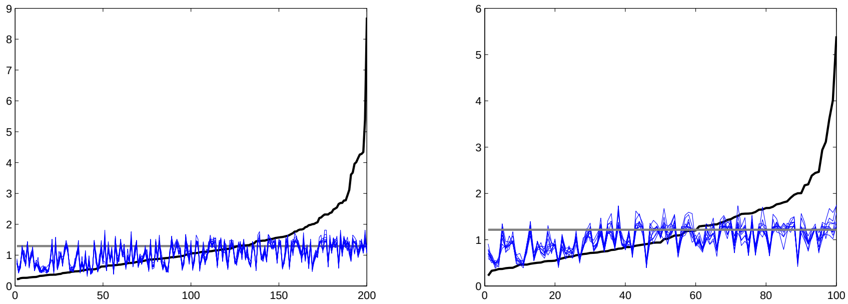
Figures 1 and 2 plot, respectively, the performance of the best neural networks and fuzzy rule-based models produced by each run, while Figure 3 compares the two best solutions obtained from the fuzzy and neuro-genetic approaches over the validation set for, respectively, $\theta = 0.01$ and $\theta = 0.02$.



$\theta = 0.01$ $\qquad$ $\theta = 0.02$

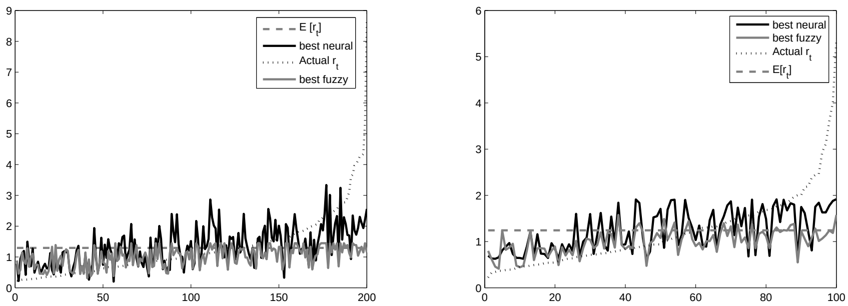
**Fig. 1.** Performance on the validation set (out-of-sample data) of the neural networks evolved by independent runs of the neuro-genetic method for the two datasets considered. The records of the validation set have been sorted by increasing $r_t$. The thick black line is the actual $r_t$, the thick light-grey line is $E[r_t]$, while the other lines represent the predictions of each neural network.

The best topologies obtained from the neuro-genetic approach for each of the two datasets are shown in Figures 4 and 5, corresponding, respectively, to $\theta = 0.01$ and 0.02. An interesting aspect that can be highlighted is that, while the first corresponds to a more usual network with normal connections, in the second network (see Figure 5) the last hidden connection is set, together with the bias of the last node, to a negative value. Such connection could represent a reversal (through a NOT operator) of the information flow; however, it seems to be an effect of the backpropagation used to train the networks.



$\theta = 0.01$ $\theta = 0.02$

**Fig. 2.** Performance on the validation set (out-of-sample data) of the fuzzy rule bases evolved by independent runs of the fuzzy-evolutionary method for the two datasets considered. The records of the validation set have been sorted by increasing $r_t$. The thick black line is the actual $r_t$, the thick light-grey line is $E[r_t]$, while the other lines represent the predictions of each fuzzy model.




$\theta = 0.01$ $\theta = 0.02$

**Fig. 3.** Comparison of the best results on the validation set of the fuzzy- and the neuro-evolutionary approaches. The dashed black line is the actual $r_t$, the dashed grey line is $E[r_t]$, while the grey line represents the best fuzzy model and the black line the best neural model.

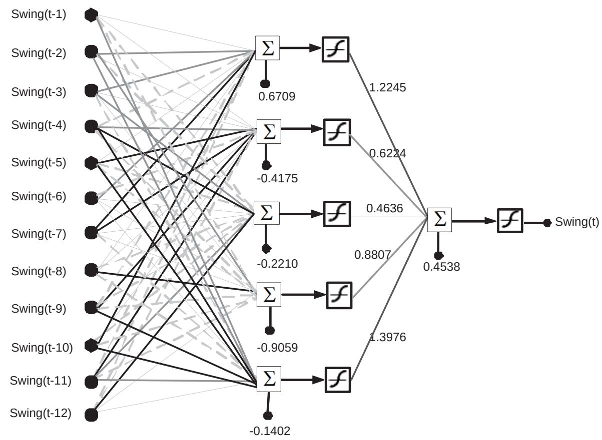



**Fig. 4.** Topology of the best neural network for $\theta = 0.01$. The thick black line refers to connection weights $w \geq 1$. The thick dark grey line corresponds to values defined as $0.5 \leq w < 1$, while the light grey line to those defined as $0 \leq w < 0.5$. Finally the dashed light grey line corresponds to negative correlations $w < 0$.

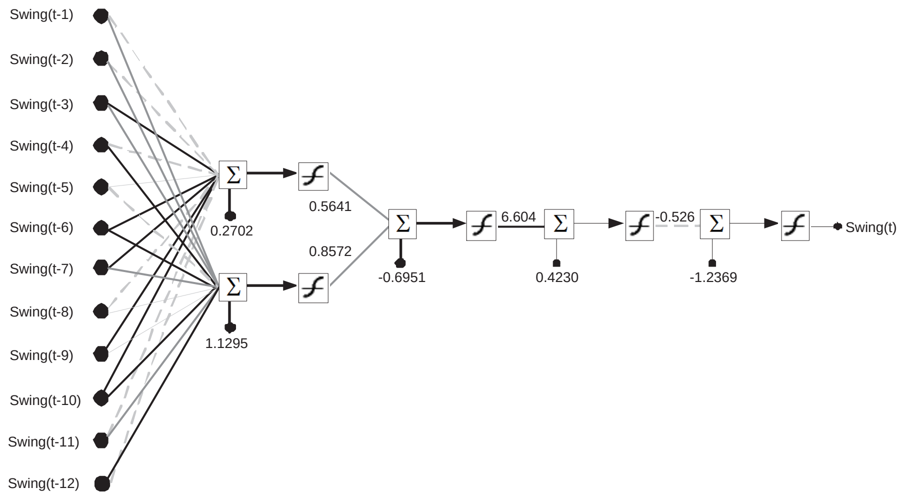



**Fig. 5.** Topology of the best neural network for $\theta = 0.02$. As reported in Figure 4, the thick black line refers to connection weights $w \geq 1$. The thick dark grey line corresponds to $0.5 \leq w < 1$, and the light grey line to $0 \leq w < 0.5$. Finally the dashed light grey line corresponds to negative correlations $w < 0$.

The best fuzzy rule bases obtained by the fuzzy-evolutionary approach for either dataset are shown in Figures 6 and 7. It can be observed that both rule bases feature a sort of *default* rule, which always fires, and sets the prediction of swing($t$) to 1.44 for $\theta = 0.01$ and 1.57 for $\theta = 0.02$, while all the remaining rules appear to be there to recognize and handle significant patterns where different prediction can be made. Interestingly, the rule base in Figure 6 uses but the three most recent swings to predict the next one; the most recent swings have a prevailing role in the rule base in Figure 7 as well, although "older" swings are looked at in a few rules.

**IF TRUE THEN** swing($t$) is 1.44
**IF** swing($t-1$) is medium-large **THEN** swing($t$) is 0.12
**IF** swing($t-3$) is very-large **AND** swing($t-11$) is small-to-medium **THEN** swing($t$) is 0.12
**IF** swing($t-1$) is medium **AND** swing($t-2$) is medium-to-large **THEN** swing($t$) is 0.12
**IF** swing($t-1$) is medium-to-huge **AND** swing($t-2$) is medium **THEN** swing($t$) is 0.12
**IF** swing($t-1$) is medium-to-huge **THEN** swing($t$) is 0.12
**IF** swing($t-1$) is medium-to-huge **THEN** swing($t$) is 0.12
**IF** swing($t-1$) is medium **AND** swing($t-2$) is medium-to-large **THEN** swing($t$) is 0.12

**Fig. 6.** The best fuzzy rule base for $\theta = 0.01$. The linguistic values have been manually given meaningful names to improve readability.

Although at first sight the predictions provided by the models found by both methods may appear disappointing, a closer examination of the graphs in Figures 1 and 2 reveals something interesting: a clear tendency can be observed for models evolved by independent runs of both methods to provide similar predictions for the same records. This cannot be a coincidence, but it must be understood as evidence that the models are striking trade-offs among errors committed when predicting $r_t$ in context that look similar as far as the recent previous history of the time series is concerned.

## 5 Conclusion and Future Work

Two independent evolutionary modeling methods have been applied to predicting trend reversals in financial time series.

The results obtained indicate that the lengths of price swings of financial instruments (detailed results have been shown regarding the S&P500 index, but the time series of all the other instruments we have tried to apply the same approach to show the same behavior) follow a largely, but not exclusively, random pattern. However, the non-random part of their behavior can be modeled and predicted. Whether this predictable part of a financial time series behavior can be effectively exploited for gaining excess returns is an open question, but we believe our results are yet another small piece of evidence against the efficient market hypothesis.

**IF** swing($t-7$) is large **AND** swing($t-11$) is medium-large **AND** swing($t-5$) is large **AND** swing($t-2$) is medium **THEN** swing($t$) is 8.65
**IF** swing($t-7$) is large **AND** swing($t-5$) is medium-large **AND** swing($t-1$) is large **THEN** swing($t$) is 15
**IF TRUE THEN** swing($t$) is 1.57
**IF** swing($t-1$) is large **AND** swing($t-8$) is medium-small **THEN** swing($t$) is 0.13
**IF** swing($t-1$) is small-to-large **AND** swing($t-2$) is medium **THEN** swing($t$) is 0.13
**IF** swing($t-2$) is medium **THEN** swing($t$) is 0.13
**IF** swing($t-6$) is around 11 **AND** swing($t-12$) is between 6 and 10 **AND** swing($t-8$) is medium-small **THEN** swing($t$) is 0.13
**IF** swing($t-1$) is small-to-large **THEN** swing($t$) is 0.13
**IF** swing($t-1$) is large **AND** swing($t-9$) is small **THEN** swing($t$) is 0.13
**IF** swing($t-1$) is large **THEN** swing($t$) is 0.13

**Fig. 7.** The best fuzzy rule base for $\theta = 0.02$. The linguistic values have been manually given meaningful names to improve readability.

## References


1. S. B. Achelist. *Technical analysis from A to Z.* Probus Publisher, Chicago, IL, 1995.
2. A. Azzini and A. Tettamanzi. A neural evolutionary approach to financial modeling. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'06*, vol. 2, pp. 1605–1612. Morgan Kaufmann, San Francisco, CA, 2006.
3. A. Azzini and A. Tettamanzi. Neuro-genetic single position day trading. In *Workshop Italiano di Vita Artificiale e Computazione Evolutiva, WIVACE 07*, 2007.
4. A. Azzini and A. Tettamanzi. Evolutionary Single-Position Automated Trading. In *Proceedings of European Workshop on Evolutionary Computation in Finance and Economics,, EVOFIN'08*, pp. 1605–1612. 2008.
5. Anthony Brabazon and Michael O'Neill. *Biologically Inspired Algorithms for Financial Modelling.* Springer, Berlin, 2006.
6. Anthony Brabazon and Michael O'Neill. *Natural Computing in Computational Finance.* Springer, Berlin, 2008.
7. Célia da Costa Pereira and Andrea Tettamanzi. Fuzzy-evolutionary modeling for single-position day trading. In Anthony Brabazon and Michael O'Neill, editors, *Natural Computing in Computational Finance*, volume 100 of *Studies in Computational Intelligence*, pages 131–159. Springer, Berlin, 2008.
8. A. Herbst. *Analyzing and Forecasting Futures Prices.* Wiley, New York, 1992.
9. E. Peters. *Chaos and Order in the Capital Markets*, 2nd Edition. Wiley, New York, 1996.
10. A. Tettamanzi R. Poluzzi, G. G. Rizzotto. An evolutionary algorithm for fuzzy controller synthesis and optimization based on SGS-Thomson's W.A.R.P. fuzzy processor. In L. A. Zadeh E. Sanchez, T. Shibata, editor, *Genetic algorithms and fuzzy logic systems: Soft computing perspectives.* World Scientific, Singapore, 1996.
11. A. Tettamanzi. An evolutionary algorithm for fuzzy controller synthesis and optimization. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5/5, pages 4021–4026. IEEE Systems, Man, and Cybernetics Society, 1995.
12. X. Yao. *Evolutionary Optimization.* Kluwer Academic Publishers, Norwell, Massachusetts, 2002.
13. X. Yao and Y.Liu. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 8(3):694–713, May 1997.