A Genetic-Based Fuzzy Grey Prediction Model

Yo-Ping Huang and Chih-Hsin Huang
Dept. of Computer Science and Engineering
Tatung Institute of Technology
Taipei, Taiwan 104
R.O.C.

ABSTRACT

Three different modeling techniques, i.e., genetic algorithm, fuzzy logic, and grey theory, are integrated to become a practical model for prediction purpose. The grey system is used to predict the next output from an unknown plant. Since the prediction error is inevitable, a fuzzy controller is designed to learn how to compensate for the output from the grey system. The roughly determined fuzzy rule base is then tuned by the genetic algorithms. The results show that the proposed technique outperforms the conventional grey system's. Also, the proposed model demonstrates its simplicity in modeling, its applicability to real-world prediction problem, and its extension ability for future intelligent control.

1. INTRODUCTION

The application of fuzzy logic to the prediction control of an unknown plant has become one of the most popular research topics [1-3]. A fuzzy rule base is designed to replace the mathematics-based model to characterize a system such that the next output from the system can be easily predicted. However, a complicated rule base has to be constructed to obtain a better prediction accuracy. In case a system has few past outputs available, the constructed rule base may not well represent the characteristics of the unknown plant. As a result, the predicted result may not be the desired one. Thus, a model which can be applied to precisely predicting the output from the system is indispensable.

The grey theory was first proposed by Prof. Deng in 1982 [4], which was 17 years later after Prof. Zadeh's fuzzy set theory [6]. A grey system is a partially known and partially unknown system. The so-called partially known or partially unknown is relative to each other. It represents our knowledge of the system status at a specific moment. Through the accumulated generating operation (AGO) [5], the seemed-to-be random output data may become exponentially behaved such that a first-order differential equation can be used to characterize the system behavior. To improve the prediction accuracy a fuzzy controller is incorporated into the grey prediction model to compensate the predicted outcome.

Since a fuzzy controller is taken to compensate the output, the roughly determined membership functions must be adjusted to satisfy this purpose. The gradient descent method has been widely applied to performing the tuning job [7]. The tuned result, however, is heavily dependent on the initial settings and the shapes of the membership functions. Besides, in the gradient descent approach, we have to take the derivatives of the system error with respect to every parameter. Instead of taking the derivatives, the genetic algorithms are exploited to adjust the membership functions simultaneously. This will in turn allows us to tune a complicated system without choosing an *ad hoc* set of initial settings.

Genetic algorithms (GAs) are search procedures based on the principles of natural evolution [8-9]. Although there are many variations on these algorithms, we will focus on the simple genetic algorithm (SGA) that processes a finite population of fixed length binary strings. The SGA consists of three major operations, i.e., reproduction, crossover, and mutation. Reproduction is the "survival-of-the-fittest" within the genetic algorithms. Crossover is an exploration operator. Mutation is another operator which occasionally alters a value in a selected string to create a new generation.

2. ADJUSTMENT OF FUZZY SET PARAMETERS

2.1 GM(1,1) model

The GM(1,1) model is a single variable first order grey model. Its scheme can be simply described in the following steps:

(1) Given initial data $X(0)=[x(1),x(2),\cdots,x(n)]$ where $x(i)$ corresponds to the system output at time i. We try to predict the next $x(n+k)$, $k\geq 1$

(2) From the initial $X(0)$ a new sequence $X(1)$ is generated by the accumulated generating operation (AGO). Where $X(1)=[x^1(1),x^1(2),\cdots,x^1(n)]$ is derived as follows:

$$x^1(k)=\sum_{m=1}^{k}x(m) \tag{1}$$

(3) From $X(1)$ we can form the following first order differential equation:

$$\frac{dx^1}{dt}+ax^1=u \tag{2}$$

(4) From Step (3) we have

$$\hat{x}^1(k+1)=(x(1)-\frac{u}{a})e^{-ak}+\frac{u}{a} \tag{3}$$

$$\hat{x}(k+1)=\hat{x}^1(k+1)-\hat{x}^1(k) \tag{4}$$

where

0-7803-2559-1/95 $4.00

$$\hat{a} = \begin{bmatrix} a \\ u \end{bmatrix} = (B^T B)^{-1} B^T y_N \tag{5}$$

$$B = \begin{bmatrix} -0.5(x^1(1)+x^1(2)) & 1 \\ -0.5(x^1(2)+x^1(3)) & 1 \\ \cdots & \\ -0.5(x^1(n-1)+x^1(n)) & 1 \end{bmatrix} \tag{6}$$

$$y_N = [x(2), x(3), \cdots, x(n)]^T \tag{7}$$

and $\hat{x}(k+1)$ is the predicted value of $x(k+1)$ at time $k+1$.

2.2 Simplified Fuzzy Reasoning

In fuzzy modeling, the performance of the designed model is essentially dependent on the validity of the acquired information. The information can either generate from the input and output data pairs or obtain from the expert knowledge. Among the methods that have been proposed so far for the automatic adjustment of fuzzy set parameters, the gradient descent method [7] provides an efficient way to build real-time learning algorithms if an *ad hoc* designed initial settings are given. Empirically, an ill start for the gradient descent method may affect the final performance. As a result, the fuzzy set parameters are fine tuned by the genetic algorithms.

When a set of input-output data is given, the inference rules can be easily described as follows:

Rule i : If x_1 is A_{i1} and x_2 is A_{i2} ... and x_m is A_{im}, then y is w_i (i=1,...,n),

where x_j's are the system input data, y is the output, A_{ij}'s are the membership functions in the antecedent part, and w_i can be either a real value, a fuzzy number, or a simple function of the inputs. In this paper, the fuzzy consequence is not considered due to the unknown of the fuzzy rules.

2.3 Simple Genetic Algorithms

Genetic algorithms (GAs) are powerful search and optimization algorithms [10-11], which resembles natural operation on genes. A candidate solution is called a chromosome and consists of a linear list of genes. A population consists of a finite number of chromosomes.

The simple genetic algorithms are listed as follows:

(1) Randomly generate an initial population G(t).
(2) Evaluate the fitness function of each individual of the current population G(t).
(3) Generate an intermediate population I(t+1) by applying the reproduction operators to G(t).
(4) Apply other operations to I(t+1).
(5) t=t+1; If (NOT_END) goto Step (2).

Here the G(t) is the current population, the I(t+1) is the intermediate population, and the G(t+1) is the next population after all operations. The genetic algorithms start by randomly generating a population of strings and each string represents one possible solution to the problem. Each of these strings is decoded to yield the actual parameters in the fuzzy logic controller. The process continues until convergence is achieved or a suitable solution is found.

There are two main factors to be considered on the GAs. One is the fitness function and the other is the encoding scheme. The fitness function that we use is given by

$$F = \frac{1}{2}\sum (y - y^r)^2 \tag{8}$$

where y^r is the desired output and y is the actual output of our model. The dimension of search spaces will be further affected by another net coding feature: the bit length; i.e., the coding granularity. Granularity [12] is a coding parameter that has always been specified in design phase. If it is too coarse a granularity may prevent the existence of a suitable solution, while if it is too fine a granularity may lead to huge search space. So, how to select the coding granularity is a very important job.

3. THE PROPOSED PREDICTION MODEL

In our model, we first use the grey prediction model to get the next output value. We take advantages of the grey model to predict the output of an unknown system. That is, we can obtain the predicted output by using the scheme described in Section 2.1. Moreover, we can obtain the error and use the error to build a fuzzy reasoning model for getting the compensation. As we know, in the past research, the gradient descent method was used to adjust the fuzzy set parameters. Now, we use the genetic algorithms to replace the gradient descent method. The characteristics of the genetic algorithms are the initial value can be randomly generated, that is, the initial values are not dependent on the expertise. And there are no learning constants needed to be adjusted as the gradient descent method did. Note that the parameters we selected can be applied to solving the underlying problems. That means that the proposed model can predict the output of an unknown system without having a rule-base or expertise knowledge.

3.1 How to build the fuzzy model

Although grey prediction model is an effective prediction model for an unknown system, it may produce sever error in some cases. This motivates us to use the fuzzy reasoning to compensate for the output from the grey system. As described in Section 2.1, we can get the grey prediction output by GM(1,1) model. Then, we can calculate the error between the prediction output and the actual output. By using the difference between every two months, we can set up the premise part in the fuzzy rule. The quantity to be compensated constitutes the consequent part of the fuzzy rule.

The inference rule is as follows :

Type I:

Rule i : If x_1 is A_i and x_2 is B_i and x_3 is C_i , then y is W_i.

In this case the output is a crisp value.

Type III:

Rule i : If x_1 is A_i and x_2 is B_i and x_3 is C_i , then y = $k_0 + k_1x_1 + k_2x_2 + k_3x_3$.

Here the output is the linear combination of the inputs.

3.2 How to tune the fuzzy set parameters

As we know, the gradient descent method is a popular method for tuning the fuzzy set parameters. To prevent from doing the tedious work in the gradient descent method, we use the genetic algorithms to replace the gradient descent method to tune the fuzzy set parameters.

In our simulations, the genetic algorithms are applied to both premise and consequent parts. The parameters which include the centers and the widths of the membership functions in the premise part and the values induced in the consequent part are tuned by the genetic algorithms. Therefore, no prior knowledge about the initial values of the membership functions or the expertise to tune the learning parameters. The genetic algorithms can solve these problems automatically.

The diagram of the genetic-based fuzzy grey model is shown in Fig. 1. In Fig. 1, x's are the input data, y is the final output value, y^* is the grey prediction output from the system, Δy represents the difference between the actual and predicted values. Combination unit represents a simple operation such as the addition operation. The genetic algorithms are both used in the premise and consequence parts in fuzzy modeling.

We can perform the proposed model as follows:

(1) Read in the input data;

(2) Use GM(1,1) model to predict the next output from the plant;

(3) Calculate the prediction error;

(4) Treat the error as the consequent part in the fuzzy rule and tune the fuzzy parameters by the genetic algorithms;

(5) Obtain the quantity of compensation;

(6) Add the compensation value to the prediction one to get the actual output.

There are two ways to terminate the genetic algorithms. One is to enforce it to stop after a fixed number of iterations such as 100 or 1000. The other one is to judge whether the objective function is met the acceptable predefined value or not.

3.3 The Hybrid Method

The simple genetic algorithms have the problem of leading to premature convergence. Therefore, we adopt a hybrid method which can avoid the premature problem. The hybrid method is shown in Fig. 2. There are two stages in the hybrid method. The first stage is to randomly generate two groups of population. Then, each group is undergone the genetic algorithms. At the end of the first stage, we choose the better chromosomes to form a new group of population. Then, this brand new group is undergone the genetic algorithms again. The ending condition of the hybrid genetic algorithms is the same as the original genetic algorithms. In the genetic algorithms, every string actually offers partial information about the expected fitness values of all possible schemata. This characteristic is termed implicit parallelism [8]. So, we can apply the first stage of the hybrid method in the parallel machine to save the computing time.

4. SIMULATION RESULTS

Since asymmetric triangular membership functions are considered here, only the center point, left spread, and right spread of each triangle should be encoded in performing the genetic algorithms. Besides, each variable in the fuzzy rule contains five labels; therefore, a total of 15m parameters has to be encoded if there are m variables in the antecedent part. To verify the applicability of the proposed model, we forecast the monthly mean temperatures for two cities, i.e., Taipei and St. Louis. For the city of Taipei, we forecast the temperatures from Jan. 1992 to Dec. 1993, if the past 18 years' data are available. While for the city of St. Louis, the 1973's and 1974's were forecast based on the data collected from 1955 to 1972. Note that in our simulations, the parameters set for the genetic algorithms are tabulated in Table 1.

We apply the genetic-based fuzzy grey model to the temperature prediction. Fig. 3 shows the monthly average temperatures for the city of St. Louis, Missouri. Fig. 4 is the prediction results based on our method for the city of St. Louis. Fig. 5 depicts the outcome from type III fuzzy model. We also repeat the simulations for the city of Taipei, Taiwan. The results are plotted in Fig. 6 and Fig. 7. The forecast error percentages are listed in Table 2 for comparisons. As can be seen from Table 2, the proposed model can improve the prediction accuracy. The only problem left for the city of St. Louis is that the error percentage cannot be reduced to a satisfactory level. This is due to the fact that there are sharp variations of temperatures between two consecutive months. This may cause the grey model to fall into the forbidden region. As a result, the compensation cannot be easily achieved. Nonetheless, our result does show that even under such a poor condition, the predicted accuracy can be further improved.

From the simulation results, we can find that the proposed method possesses a better prediction capability. However, the compensation job performed by the proposed model is not so satisfied. This is because the searching space is implicit and the genetic algorithms cannot display its capability in finding the optimal value of compensation.

We next try two other functions. The first function is:

$$F(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2,$$
$$-2.048 \le x_1, x_2 \le 2.048$$

and is shown in Fig. 8. The simulation results are depicted in Fig. 9. It is obvious that the hybrid method can reach the global optimal value but the original method cannot.

The second function under simulation is:

$$g(u) = 6\sin(\pi u) + 3\sin(3\pi u) + \sin(5\pi u),$$

$$u = \sin(\frac{2\pi k}{250}),$$
$$275 \le k \le 375, \qquad k\text{:integer}$$

and is plotted in Fig. 10. Again, the simulation results are shown in Fig. 11. It is clear that the hybrid method performs better than the original one.

In fact, we know that in the genetic algorithms the multiple-peak function usually causes a dilemma while the single-peak function is easy for the genetic algorithms. We can conclude that in multi-optimal functions, the proposed method is better than the original one because we can avoid the premature convergence problem and has better chance to reach the global optimal quickly.

Table 1. The parameters used for the genetic algorithms.

Population Size	30
Chromosome Length : Type I	170
Chromosome Length : Type III	545
Binary Bits Length : Type I	7
Binary Bits Length : Type III	9
Crossover Rate	0.8
Mutation Rate : Type I	0.000084
Mutation Rate : Type III	0.000021

Table 2. The average error percentages from different models for both cities of St. Louis and Taipei.

	St. Louis	Taipei
Grey model	15.7174 %	9.4163 %
Fuzzy Grey Method (I)	7.2893%	5.1141%
Fuzzy Grey Method (III)	8.4253 %	6.0166 %

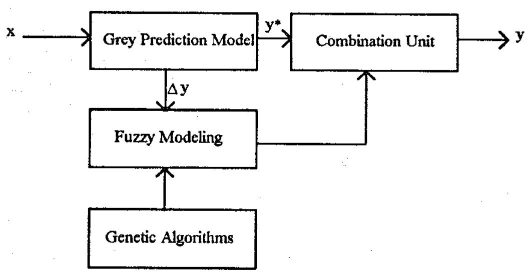

Fig. 1. The diagram of the genetic-based fuzzy grey model.

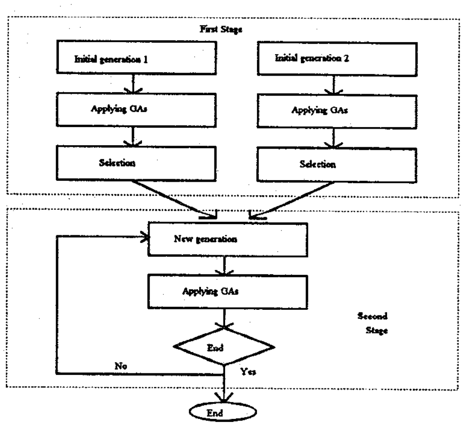

Fig. 2. The diagram of the proposed hybrid method.

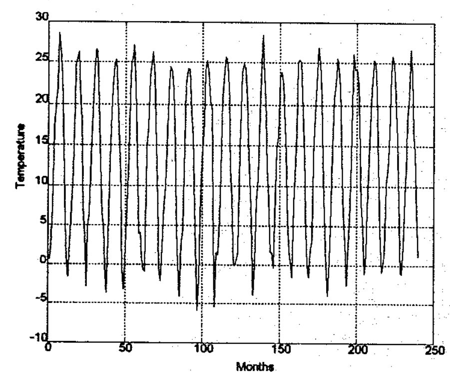

Fig. 3. The monthly average temperatures for the city of St. Louis, Missouri, from Jan. 1955 to Dec. 1974

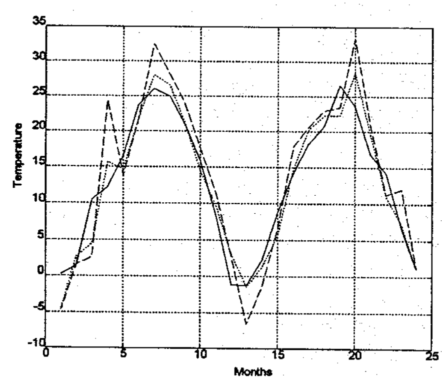

Fig. 4. The forecast outputs between Jan. 1973 and Dec. 1974 for St. Louis. Solid, dashed, and dotted curves represent the outputs from the actual, the GM(1,1), and our models with type I, receptively.

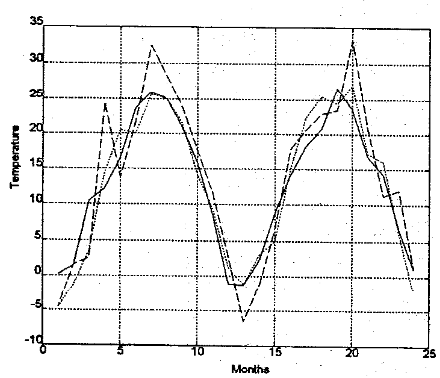

Fig. 5. The forecast outputs between Jan. 1973 and Dec. 1974 for St. Louis. Solid, dashed, and dotted curves represent the outputs from the actual, the GM(1,1), and our models with type III, receptively.

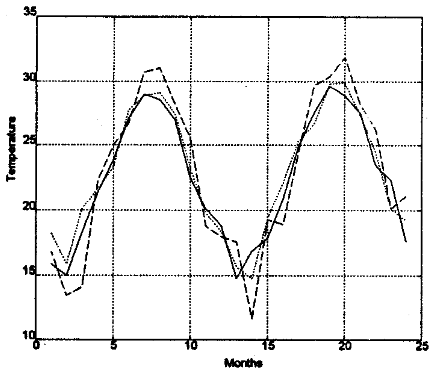

Fig. 6. The forecast outputs between Jan. 1992 and Dec. 1993 for the city of Taipei by the type-I fuzzy rules. In this figure, the solid, dashed, and dotted curves represent the outputs from the actual, the GM(1,1), and our models, respectively.

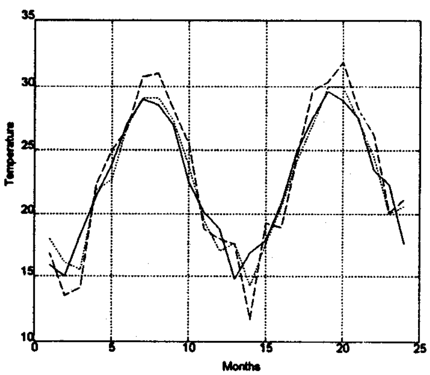

Fig. 7. The forecast outputs between Jan. 1992 and Dec. 1993 for the city of Taipei by the type-III fuzzy rules. In this figure, the solid, dashed, and dotted curves represent the outputs from the actual, the GM(1,1), and our models, respectively.

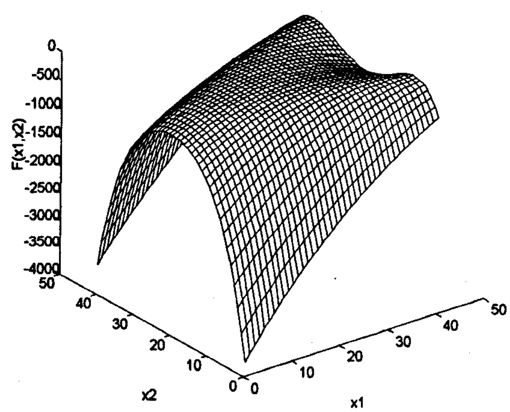

Fig. 8. The first example of multiple-peak mathematical function.

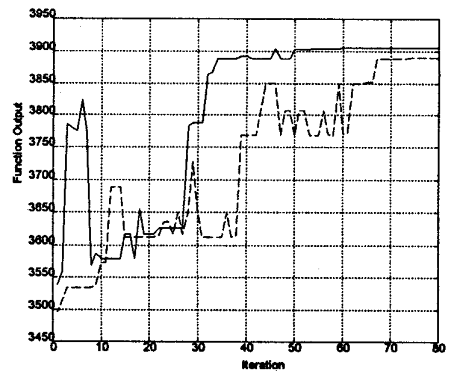

Fig. 9. The simulation results from the first test function. The solid curve is the outputs of the hybrid method and the dashed one is the outputs of the original genetic algorithms.

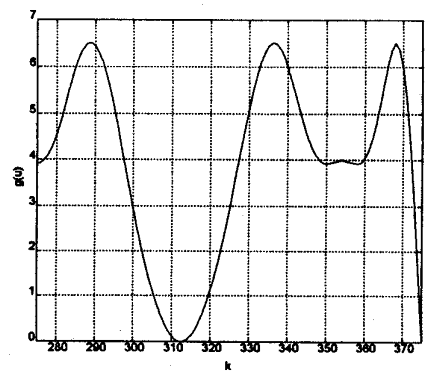

Fig. 10. The second example of multiple-peak mathematical function.

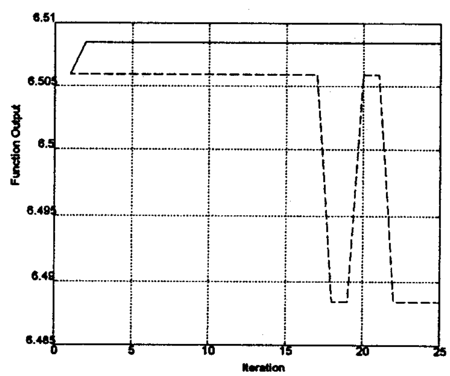

Fig. 11. The simulation results from the second test function. The solid curve is the outputs of the hybrid method and the dashed one is the outputs of the original genetic algorithms.

V. CONCLUSIONS

The procedures for designing the proposed fuzzy grey prediction model can be summarized as follows:

(1) Use the sampled outputs from the system to form some basic grey models. Four data points are the minimum requirement for constructing a grey model. Our purpose is to predict the next output from the system from the constructed grey models.
(2) In the training phase, treat the predicted error caused by the grey model as the consequent part and the data points used in constructing the grey model as the antecedent part in each fuzzy rule. The inference type considered here is the well-known Takagi-Sugeno one.
(3) In case there are a lot of past data available, (for example, use the past 20 years' temperatures to predict the just past two years') use the grey relational method to select some crucial patterns which have the similar weather behavior with the current one to expedite the tuning process. If not enough past data available, then just skip the grey relational analysis.
(4) Use the genetic algorithm to fine tune the membership functions.
(5) In the prediction phase, the current input pattern is then fed to the well-tuned system to determine the quantity of compensation. The compensation is later added to the predicted output such that the prediction result can be further improved.

Based on the simulation results presented here, we found that the proposed compensation model can improve the prediction performance. The hybrid genetic algorithms provide a better chance to reach the optimal solutions while keeping the final population a compact size. The type II fuzzy inference model is not yet considered. This is because we have no prior information for constructing the rule base. As a result, there is no way to find the application of genetic algorithms. Future work will focus on how to form the rule base such that the genetic algorithms can be used to adjust the parameter set.

REFERENCES

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 15, no. 1, pp.116-132, Jan./Feb., 1985.

[2] C.C. Lee, "Fuzzy logic in control systems : Fuzzy logic controller - Part I," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 20, no. 2, pp.404-418, March/April, 1990.

[3] C.C. Lee, "Fuzzy logic in control systems : Fuzzy logic controller - Part II," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 20, no. 2, pp.419-435, March/April, 1990.

[4] J.L. Deng, "Control problems of grey systems," *Systems and Control Letters*, 5, pp.228-294, 1982.

[5] J.L. Deng, "Introduction to grey system theory," *The J. of Grey System*, vol. 1, no. 1, pp.1-24, 1989.

[6] L.A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp.338-352, 1965.

[7] H. Nomura, I. Hayashi, and N. Wakami, "A learning method of fuzzy inference rules by descent method," in *Proc. FUZZ-IEEE*, pp.203-210, 1992.

[8] D.B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp.3-14, Jan., 1994.

[9] K.D. Jong, "Adaptive system design; A genetic approach," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 10, no. 9, pp.566-574, Sept., 1980.

[10] C. Karr, "genetic algorithms for fuzzy controllers," *AI Expert*, pp.26-33, Feb., 1991.

[11] C. Karr, "Applying genetics to fuzzy logic," *AI Expert*, pp.38-43, March., 1991.

[12] V. Maniezzo, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Trans. on Neural Networks*, vol. 5, no. 1, pp.39-53, Jan., 1994.