Neurocomputing 72 (2009) 2418–2432

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Reinforcement group cooperation-based symbiotic evolution for recurrent wavelet-based neuro-fuzzy systems

Yung-Chi Hsu, Sheng-Fuu Lin *

*Department of Electrical and Control Engineering, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC*

A R T I C L E I N F O

*Article history:*
Received 9 August 2007
Received in revised form
26 June 2008
Accepted 8 December 2008
Communicated by T. Heskes
Available online 23 January 2009

*Keywords:*
Neuro-fuzzy system
Symbiotic evolution
Control
Reinforcement learning
Recurrent network

A B S T R A C T

This paper proposes a recurrent wavelet-based neuro-fuzzy system (RWNFS) with a reinforcement group cooperation-based symbiotic evolution (R-GCSE) for solving various control problems. The R-GCSE is different from the traditional symbiotic evolution. In the R-GCSE method, a population is divided to several groups. Each group formed by a set of chromosomes represents a fuzzy rule and cooperates with other groups to generate better chromosomes by using the proposed elite-based compensation crossover strategy (ECCS). In this paper, the proposed R-GCSE is used to evaluate numerical control problems. The performance of the R-GCSE in the simulations is excellent compared with other existing models.



## 1. Introduction

In recent years, fuzzy logic or artificial neural networks used to solve control problems have become a popular research topic [1–10]. The reason is that classical control theory usually requires a mathematical model for designing controllers. The inaccuracy of mathematical modeling of plants usually degrades the performance of the controllers, especially for nonlinear and complex control problems [11–14]. Fuzzy logic has the ability to express the ambiguity of human thinking and to translate expert knowledge into computable numerical data.

A fuzzy system consists of a set of fuzzy IF–THEN rules that describe the input–output mapping relationship of networks. Obviously, it is difficult for human experts to examine all the input-output data from a complex system to find proper rules for a fuzzy system. To cope with this difficulty, several approaches used to generate the fuzzy IF–THEN rules from numerical data have been proposed [2,3,6]. These methods were developed for supervised learning; i.e., the correct "target" output values are given for each input pattern to guide the learning of the network. However, most of the supervised learning algorithms for neural fuzzy networks require precise training data in order to tune the networks for various applications. For some real world applications, precise training data are usually difficult and expensive, if not impossible, to obtain. For this reason, there has been a growing interest in reinforcement learning algorithms for neural controller [15–18] or fuzzy [19–21] design.

In designing a fuzzy controller, adjusting the required parameters is important. To do this, back-propagation (BP) training was used in [3,6–8]. It is a powerful training technique that can be applied to networks with a forward structure. Since the steepest descent technique is used in BP training to minimize the error function, the algorithms may reach the local minima very fast and never find the global solution. To solve these problems, several evolutionary algorithms, such as genetic algorithm (GA) [22], genetic programming [23], evolutionary programming [24], and evolution strategies [25], have recently been proposed. They are parallel and global search techniques. Because they simultaneously evaluate many points in the search space, they are more likely to converge toward the global solution. For this reason, evolutionary methods, which are used for training fuzzy models, have become an important field.

The evolutionary fuzzy model generates a fuzzy system automatically by incorporating evolutionary learning procedures [26–33]. The most well-known evolutionary learning procedure is GAs. Several genetic fuzzy models have been proposed [26–31]. In [26], Karr applied GAs to design the membership functions of a fuzzy controller with its fuzzy rule set being assigned in advance. Since the membership functions and rule sets are co-dependent, simultaneous design of these two approaches is a more appropriate methodology.

Based on this concept, many researchers have applied GAs to optimize both the parameters of the membership functions and

* Corresponding author.
*E-mail address:* sflin@mail.nctu.edu.tw (S.-F. Lin).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.12.027

the rule sets [27–29]. Lin and Jou [30] proposed GA-based fuzzy reinforcement learning to control magnetic bearing systems. Juang et al. [31] proposed using genetic reinforcement learning to design fuzzy controllers. The GA adopted in [31] was based on traditional symbiotic evolution which, when applied to fuzzy controller design, complements the local mapping property of a fuzzy rule. In [32] Tang proposed a hierarchical genetic algorithm. The hierarchical GA enables the optimization of the fuzzy system design for a particular application. Juang [33] proposed the combination of online clustering and Q-value based GA for reinforcement fuzzy system (CQGAF) to simultaneously design the number of fuzzy rules and the free parameters in a fuzzy system.

However, these approaches encounter one or more of the following major problems: (1) all the fuzzy rules are encoded into one chromosome; (2) the population cannot evaluate each fuzzy rule locally.

Recently, Gomez and Schmidhuber [34,35] proposed solutions for these problems. The proposed enforced sub-populations (ESP) used sub-populations of neurons for the fitness evaluation and overall control. As shown in [34,35], the sub-populations that are used to evaluate the solution locally can obtain better performance compared to systems of only one population which are used to evaluate the solution.

As with [34,35], in this paper, a RWNFS with a reinforcement group cooperation-based symbiotic evolution (R-GCSE) is proposed for solving the problems mentioned above. In the proposed R-GCSE, each chromosome represents only one fuzzy rule, and the $n$-rules fuzzy system is constructed by selecting and combining $n$ chromosomes from several groups. The R-GCSE, which promotes both cooperation and specialization, ensures diversity and prevents a population from converging to suboptimal solutions. In the R-GCSE, compared with normal symbiotic evolution, several groups are in the population. Each group formed by a set of chromosomes represents a fuzzy rule. Compared with [34,35] to let the well-performing groups of individuals cooperate to create better generations, an elite-based compensation crossover strategy (ECCS) is proposed in this paper. In the ECCS, each group cooperates to perform the crossover steps. Therefore, the better chromosomes of each group will be selected to perform the crossover steps in the next generation.

The advantages of the R-GCSE are summarized as follows: (1) the R-GCSE uses group-based populations to evaluate the fuzzy rule locally; (2) the R-GCSE uses the ECCS to allow better solutions from different groups to cooperate in order to generate better solutions in the next generation; (3) it indeed performs better performance and converges more quickly than some traditional genetic methods.

This paper is organized as follows. In Section 2, the RWNFS is introduced. In Section 3, the proposed group cooperation-based symbiotic evolution (GCSE) is described. In Section 4, the reinforcement group cooperation-based symbiotic evolution (R-GCSE) using for constructing the RWNFS model is introduced. In Section 5, the simulation results are presented. The conclusions are summarized in the last section.

## 2. Structure of a RWNFS

In this section, the structure of RWNFS shown in Fig. 1 will be introduced. For TSK-type fuzzy networks [1,5], the consequence of each rule is a function input linguistic variable. A widely adopted function is a linear combination of input variables plus a constant term. This study adopts a nonlinear combination of input variables (i.e., wavelet neural network (WNN)). The advantages of the WNN are as follows: (1) its ability to find "universal approximation"; (2) an explicit link between the wavelet transform and the network coefficient is completed, and an initial guess of network parameters can be derived by the decomposition of a wavelet formula; (3) it probably obtains the same approximation performance as a smaller size network; in addition, wavelet networks are optimal approximators since the smallest number of bits are required to obtain an arbitrary precision [36].

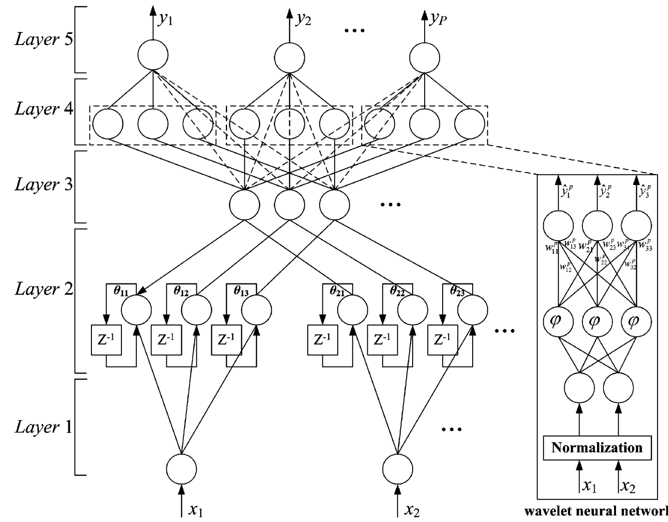


**Fig. 1.** Schematic diagram of RWNFS model.

In RWNFS, each fuzzy rule corresponds to a sub-WNN which consists of single-scaling wavelets [37]. The non-orthogonal and compact wavelet functions used as the node function (wavelet bases) are adopted in this paper. The purpose of introducing a fuzzy model into WNN is to improve the accuracy of function approximation based on the dilation and translation parameters of wavelets while not increasing the number of wavelet bases. A RWNFS is composed of fuzzy rules that can be presented in the following general form:

$$R^j: \text{ If } I_1 \text{ is } A_{1j} \text{ and} \ldots I_{ij} \text{ is } A_{ij} \text{ and } \ldots \text{and } I_{nj} \text{ is } A_{nj}$$
$$\text{Then } \hat{y}_j^1 = \sum_{k=1}^{M} w_{jk}^1 \varphi_{a.b} = w_{j1}^1 \varphi_{0.0} + w_{j2}^1 \varphi_{1.0} + w_{j3}^1 \varphi_{1.1} \cdots$$
$$\text{and } \hat{y}_j^2 = \sum_{k=1}^{M} w_{jk}^2 \varphi_{a.b} = w_{j1}^2 \varphi_{0.0} + w_{j2}^2 \varphi_{1.0} + w_{j3}^2 \varphi_{1.1} \cdots$$
$$\vdots \tag{1}$$

where $R^j$ denotes the $j$th rule; ($I_{1j},\ldots, I_{ij},\ldots,I_{nj}$) is the network input pattern ($x_1,\ldots, x_i,\ldots,x_n$) plus the temporal term for the linguistic term of the precondition part $A^j = (A_{1j}, \ldots, A_{ij}, \ldots, A_{nj})$; the local WNN model's outputs $\hat{y}_j^1$ and $\hat{y}_j^2$ are calculated for outputs $y_1$ and $y_2$ of rule $R^j$.

Next, the signal propagation is indicated, along with the operation functions of the nodes in each layer. In the following description, $I_i^{(h)}$ denotes the $i$th node's input in the $h$th layer, and $O_i^{(h)}$ denotes the $i$th node's output in layer $h$.

In layer 1, nodes just transmit input signals to the next layer directly, that is,

$$O_i^{(1)} = I_i^{(1)} \tag{2}$$

where $I_i^{(1)} = (x_1, \ldots, x_i, \ldots, x_n)$. Each precondition part of the $j$th rule $A^j = (A_{1j}, \ldots, A_{ij}, \ldots, A_{nj})$ (a group of fuzzy sets) is described here by a Gaussian-type membership function; that is, the membership value specifying the degree of how an input value belongs to a fuzzy set is determined in layer 2. The Gaussian