Information Sciences 270 (2014) 154–171

Contents lists available at ScienceDirect

Information Sciences

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

# Neural fuzzy inference systems with knowledge-based cultural differential evolution for nonlinear system control

Cheng-Hung Chen *, Sheng-Yen Yang

*Department of Electrical Engineering, National Formosa University, No. 64, Wunhua Rd., Huwei Township, Yunlin County 632, Taiwan*

ARTICLE INFO

*Article history:*
Received 27 November 2011
Received in revised form 9 January 2014
Accepted 11 February 2014
Available online 26 February 2014

*Keywords:*
Cultural algorithm
Differential evolution
Neural fuzzy inference system
Water bath temperature system
Backing up the truck
Ball and beam system

ABSTRACT

This study proposes a knowledge-based cultural differential evolution (KCDE) method for neural fuzzy inference systems (NFIS). Cultural algorithms acquire the belief space from the evolving population space and then exploit that information to guide the search. The proposed KCDE method adopts the mutation strategies of differential evolution as knowledge sources to influence the population space. The proposed KCDE method uses these knowledge sources, including normative knowledge, situational knowledge, domain knowledge, history knowledge, and topographic knowledge, to optimize the parameters of the NFIS model to avoid falling in a local optimal solution and to ensure the searching capacity of approximate global optimal solution. Experimental results demonstrate that the proposed NFIS-KCDE method performs well in nonlinear system control problems.



## 1. Introduction

Fuzzy systems and neural networks have recently become effective methods of solving nonlinear system control problems [1–3]. Compared to traditional control methods, they provide the major advantage of not requiring any complex mathematical models of the plants. Neural fuzzy inference systems (NFIS) [4–8] have been used to solve many engineering problems. These systems combine the learning capability of neural networks with the capability of fuzzy reasoning using linguistic information pertaining to numerical variables. Therefore, this study adopts a neural fuzzy inference system (NFIS). The NFIS model is based on our previous research [9], and the consequent part of the NFIS model is a functional link neural network (FLNN) [10] that consists of a nonlinear combination of input variables. The FLNN method uses a basic function approximation theorem and random linear combination of input signals to extend the input space. This design successfully omits the hidden layer of multilayer perceptron neural networks. Thus, the FLNN model has a faster convergence rate and a smaller computational load than multilayer neural networks.

Recent developments in genetic algorithms (GAs) have provided a method for neural fuzzy inference systems design. Genetic fuzzy systems (GFSs) [11–16] hybridize the approximate reasoning of fuzzy systems with the learning capability of genetic algorithms. GAs are highly effective techniques for evaluating system parameters and finding global solutions while optimizing the overall structure. Thus, many researchers have developed GAs to implement fuzzy systems and neural fuzzy inference systems to determine structures and parameters. Similar to GAs, the differential evolution (DE) algorithm [17] also belongs to the broad class of evolutionary algorithms. However, DE has advantages over GAs or any other traditional optimization approach especially for real valued problem. These advantages include a strong search ability and fast convergence

* Corresponding author. Tel.: +886 56315612.
*E-mail address:* chchen.ee@nfu.edu.tw (C.-H. Chen).

http://dx.doi.org/10.1016/j.ins.2014.02.071
0020-0255/

[17–19]. The DE algorithm has gradually increased in popularity, and has been used in many practical areas. Previous studies [20–22] demonstrate that DE is robust, simple to implement and use, easy to understand, and requires only a few control parameters. The cultural algorithm (CA) [23] can exploit the information of a specific belief space to guide the feasible search space, and can change the direction of each individual in the solution space. The cultural algorithm contains two basic components: the belief space and the population space. The belief space is an information repository in which the individuals store their experiences for other individuals to learn from them indirectly. The population space comprises a set of possible solutions to the problem, and can be modeled using any population-based approach. Researchers have successfully applied CA to many optimization problems [24–28].

This study presents a knowledge-based cultural differential evolution (KCDE) method for neural fuzzy inference systems (NFIS). The proposed KCDE method uses five mutation strategies from differential evolution to implement five knowledge sources in the belief space of the cultural algorithm. These five knowledge sources are chosen using a roulette wheel selection method to generate a new population in KCDE. The proposed KCDE method offers the following advantages: (1) the proposed KCDE method uses the five kinds of different knowledge sources to increase the global search ability; (2) the proposed KCDE method is less likely to fall into the local optimal solution; and (3) the proposed KCDE method can obtain a lower RMS error than other methods.

The remainder of this study is organized as follows. Section 2 describes the basic concept of differential evolution and cultural algorithm. Section 3 describes the structure of the NFIS model. Section 4 introduces the proposed KCDE method. Section 5 presents the simulation results for nonlinear control problems. Finally, Section 6 draws conclusions.

## 2. Differential evolution and cultural algorithm

### 2.1. Differential evolution algorithm

Differential evolution (DE) [17] is a parallel direct search method. The DE method generates new parameter vectors by adding the weighted difference between two population vectors to a third vector. DE is usually implemented in a set of vectors, and it contains the D-dimensional vector of real values. The population $\mathbf{P}_x^g$ is formed by the NP target vector $\mathbf{x}_i^g$, defined as follows:

$$\mathbf{P}_x^g = (\mathbf{x}_i^g), \quad i = 0, 1, \ldots, NP-1, \quad g = 0, 1, \ldots, g_{\max} \tag{1}$$

$$\mathbf{x}_i^g = \left(x_{j,i}^g\right), \quad j = 0, 1, \ldots, D-1$$

where $g$ represents the evolution generation, $i$ represents the $i$th vector, and $j$ represents the dimension of vectors. The classical DE has three major components: mutation, crossover, and selection. The basic DE strategy can be described as follows:

#### 2.1.1. Mutation

The main difference between DE and other evolutionary algorithms is the mutation operation. The mutation operation applies a vector difference to create a perturbation vector to a third vector. The following equation shows how to randomly select three vectors to create a mutant vector:

$$\mathbf{v}_i^g = \mathbf{x}_{r0}^g + F \cdot (\mathbf{x}_{r1}^g - \mathbf{x}_{r2}^g) \tag{2}$$

where $r_0$, $r_1$, $r_2 \in \{1, \ldots, NP\}$ are randomly selected indexes in which $r_0 \neq r_1 \neq r_2 \neq i$ and $F$ is a scaling factor ($F \in (0, 1+)$), which is a positive real number.

#### 2.1.2. Crossover

To complement the differential operation search strategy, DE uses a uniform crossover. The element of trial vector $\mathbf{u}_i^g$ are inherited from the target vector $\mathbf{x}_i^g$ and the mutant vector $\mathbf{v}_i^g$, determined by a parameter called crossover probability ($C_r$), as follows:

$$\mathbf{u}_i^g = u_{j,i}^g = \begin{cases} v_{j,i}^g & \text{if } (rand_j(0,1) \leqslant C_r) \\ x_{j,i}^g & \text{otherwise} \end{cases} \tag{3}$$

where $C_r \in [0, 1]$ is a user-defined value and $rand_j(0, 1)$ is a uniform random number generator.

#### 2.1.3. Selection

If the trial vector $\mathbf{u}_i^g$ performs better than the target vector $\mathbf{x}_i^g$, the trial vector replaces the target vector in the next generation. The inheritance parameters for each trial vector and target vector are as follows:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g & \text{if } f(\mathbf{u}_i^g) \leqslant f(\mathbf{x}_i^g) \\ \mathbf{x}_i^g & \text{otherwise} \end{cases} \tag{4}$$

The DE method repeats the steps above until it reaches the maximum evolution generation or finds the optimal solution.