Applied Mathematical Modelling 39 (2015) 383–395

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Compensatory neural fuzzy network with symbiotic particle swarm optimization for temperature control

Chun-Cheng Peng [a], Cheng-Hung Chen [b,*]

[a] Department of Information and Communication Engineering, Chaoyang University of Technology, 168, Jifeng E. Rd., Wufeng District, Taichung City 41349, Taiwan
[b] Department of Electrical Engineering, National Formosa University, 64, Wunhua Rd., Huwei Township, Yunlin County 63201, Taiwan

ARTICLE INFO

Article history:
Received 1 June 2012
Received in revised form 2 May 2014
Accepted 29 May 2014
Available online 11 June 2014

Keywords:
Water bath temperature system
Neural fuzzy networks
Symbiotic evolution
Particle swarm optimization

ABSTRACT

This study proposes a symbiotic particle swarm optimization (SPSO) algorithm for compensatory neural fuzzy networks (CNFN). The CNFN model using compensatory fuzzy operators makes fuzzy logic systems more adaptive and effective. The proposed SPSO algorithm adopts a multiple swarm scheme that uses each particle to represent a single fuzzy rule and each particle in each swarm evolves separately to avoid falling into a locally optimal solution. Additionally, the SPSO embeds the symbiotic evolution scheme in a specific particle swarm optimization (PSO) to accelerate the search and increase global search capacity. Finally, the proposed CNFN with SPSO (CNFN-SPSO) method is applied to control a water bath temperature system. Results of this study demonstrate the effectiveness of the proposed CNFN-SPSO method.

1. Introduction

Temperature control is an important factor in many process control systems. If the temperature is too high or too low, the final product may be seriously affected. Therefore, it is necessary to reach desired temperature points quickly and avoid large overshoots. Because process-control systems are often nonlinear and tend to change in unpredictable ways, they are not easy to control accurately. Adaptive control [1–4] is a model-free controller that can be used to control nonlinear systems. Most adaptive controllers involve certain types of function approximators derived from input/output experiments. Generally, the basic objective of adaptive control is to maintain consistent performance of a control system in the presence of the design parameters. However, traditional adaptive control theory only deals with systems with known dynamic structures, not unknown parameters. Furthermore, traditional adaptive controllers cannot make use of human experience, knowledge that is usually expressed in linguistic terms.

Neural fuzzy networks (NFN) have been demonstrated to solve many engineering problems [5–7]. They combine the capability of neural networks to learn from processes and the capability of fuzzy reasoning under linguistic information pertaining to numerical variables. On the other hand, recent development in genetic algorithms (GAs) has provided a new method for neural fuzzy network design. Genetic fuzzy systems (GFSs) [8–13] hybridize the approximate reasoning of fuzzy systems with the learning capability of genetic algorithms.

GAs represent highly effective techniques for evaluating system parameters and finding global solutions while optimizing the overall network structure. Thus, many researchers have developed GAs to implement fuzzy systems and neural fuzzy

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

http://dx.doi.org/10.1016/j.apm.2014.05.040

networks to automate the determination of structures and parameters [14–17]. In the aforementioned literature, it has been fully demonstrated that GAs are very powerful in searching for the true profile (global optimum). However, the search is extremely time-consuming, and represents one of the basic disadvantages of all GAs. Although the convergence in some special cases can be improved by hybridizing GAs with some local search algorithms, it is achieved at the expense of versatility and simplicity of the algorithm. In addition, a new optimization algorithm, called particle swarm optimization (PSO), appears to be better than genetic algorithms. It is an evolutionary computation technique that was developed by Kennedy and Eberhart in 1995 [18,19]. The underlying motivation for the development of the PSO algorithm is the social behavior of animals, such as bird flocking, fish schooling, and bee swarm theory. Furthermore, Shi and Eberhart [20,21] proposed a linearly decreasing inertial weight method (LPSO) to provide a balance between global and local explorations. Another important modification to PSO uses a constriction factor into PSO (CPSO), originally proposed by Clerc and Kenedy [22,23]. PSO has been successfully applied to many optimization problems, such as control problems [24–27] and feed-forward neural network designs [28–31].

This study proposes a novel symbiotic particle swarm optimization (SPSO) for a compensatory neural fuzzy network (CNFN). The neural fuzzy network is based on our previous research [32,33] with adaptive compensatory fuzzy reasoning to dynamically adjust fuzzy operators. The proposed SPSO embeds the symbiotic evolution (SE) scheme into PSO to accelerate the search and increase global search capacity. Unlike the GAs where an entire population represents a full solution to a problem, SE [34,35] assumes that each individual in a population represents only a partial solution to a problem. Complex solutions result from the combination of several individuals in the population. Furthermore, Lin et al. [36] proposed a self-evolving evolutionary learning algorithm (SEELA) that uses a subgroup symbiotic evolution and an elite-based structure strategy to determine the number of fuzzy rules. However, the SEELA still presents the formula (the local best and global best) of the traditional PSO to evolve, and therefore, may find a suboptimal solution. The proposed SPSO where each particle represents a single fuzzy rule differs from original SE [34] to be able to adopt a multiple swarm scheme. A fuzzy system with *R*-rules is constructed by selecting and combining *R* particles from each swarm, and allowing the rule itself to evolve. In SPSO, each particle is updated by applying local best, global best, and cooperative best factors differ from SEELA [36]. Our proposed method improves the search ability and greatly increases the convergence speed in simulations.

This study is organized as follows. Section 2 describes the basic concept of particle swarm optimization and SE and Section 3 illustrates the structure of the CNFN. Section 4 presents the SPSO approach for the CNFN model. Next, the proposed CNFN-SPSO method is applied to control water bath temperature to demonstrate its learning capability in Section 5. Finally, the last section discusses and concludes this paper.

2. Particle swarm optimization and symbiotic evolution

This section describes basic concepts concerning both the PSO and SE approaches. The specialization property of PSO and SE is consistent with the learning algorithm property of a neural fuzzy network. Therefore, the development of a neural fuzzy network based on PSO and SE is considered valuable.

2.1. Particle swarm optimization

Being a population-based optimization approach, the original PSO method was firstly introduced in 1995 [18,19], where the population is called a swarm. By definition, each swarm consists of many particles, while each particle, with a velocity vector v_i and a position vector x_i, depicts a possible solution to a given problem. In other words, considering an optimization task that requires the simultaneous optimization of variables, a swarm of particles is used to define and assign initialized positions for all particles within the *P*-dimensional problem space. Then, via individual vector v_i the particles move around rapidly in order to search the given solution space. The simple rule then applied in PSO is that, each of the particle positions is scored to obtain a fitness value, based on how well it defines the solution to a problem. Afterwards, a new velocity for each particle [20] is calculated by taking both the local best position (*Lbest*) of each particle and the global best position (*Gbest*) in the swarm into account, i.e.,

$$v_i(k+1) = \omega v_i(k) + \varphi_1 r_1 (Lbest - x_i(k)) + \varphi_2 r_2 (Gbest - x_i(k)), \tag{1}$$

where ω, φ_1, and ϕ_2 are respectively called the coefficients of the inertia, cognitive, and society terms. In the original PSO method, ω is set in the range of [0.4, 0.9], both ϕ_1 and ϕ_2 are set to 2, and r_1 and r_2 are yield uniformly distributed random numbers in [0, 1]. In addition, the term v_i is limited to the range between $\pm v_{\max}$. If a particle's velocity violates this limit, then it is set to the actual limit. Changing the corresponding velocity enables each particle to further search around its individual and global best positions. Based on the updated velocities, each particle changes its position vector via

$$x_i(k+1) = x_i(k) + v_i(k+1). \tag{2}$$

2.2. Symbiotic evolution

By evolved artificial antibodies to match or detect artificial antigens, the SE [34,35] notion is applied the similar mechanism of implicit fitness sharing within an immune system model. Since each antibody can only match a single antigen