Engineering Applications of Artificial Intelligence 38 (2015) 59–69

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Neuro-fuzzy approach for online message scheduling

Mu-Song Chen *

Department of Electrical Engineering, Da-Yeh University, ChangHua, 51591 Taiwan, China

ARTICLE INFO

Article history:
Received 20 May 2014
Received in revised form
14 September 2014
Accepted 3 October 2014
Available online 18 November 2014

Keywords:
Message scheduling controller
Neuro-fuzzy network
Radial basis function network
EDF

ABSTRACT

Due to timing constraints, uncertain natures, dynamical characteristics, and lack of exact mathematical model of the message scheduling system, an analytical solution of the optimal scheduling sequence is not easy to obtain. Regarding this issue, the concept of message scheduling controller (MSC) is presented in this article. This study makes use of an online neuro-fuzzy approach to realize the MSC in dealing with this subject, under the conditions of multiple queues and deadline constraints of messages. The neuro-fuzzy networks (NFNs) can bring the low-level learning of the radial basis function networks (RBFNs) into the high-level fuzzy systems, and also provide the reasoning characteristic of fuzzy systems into the RBFNs. Furthermore, the proposed NFN-based MSC (MSC_NFN) with novel learning strategy not only automates the adding or pruning fuzzy rules, but also allocates suitable position of fuzzy membership functions as well as values of consequent parameters to perform subsequent optimization efficiently. Specifically, adaptations of the network structure and parameters are performed to explore the dynamical behavior of the message scheduling schemes. Simulation results illustrate the superiority of the MSC_NFN when compared with the dynamic earliest-deadline first (EDF) scheduling and the RBFN-based MSC (MSC_RBFN), in terms of untimely service ratios, quality of service, and number of fuzzy rules.

1. Introduction

Real-time systems can span from microcontrollers in automobile engines to very complex applications, such as flight control in airport or process control in manufacturing plants. Definitely, event[1] scheduling in such systems is especially important as it aims towards devising a feasible schedule, subject to event characteristics, timing constraints, and resource requirements, etc. For example, an anti-lock brake system is expected to release a vehicle's brake and prevents dangerous wheel locking in a predictable time interval. This is particularly important when the loss of life may occur as a result of system failures. Thus, it is vital to enable certain events to complete on time, minimize waiting time, and maximize utilization of resources. Usually, many real-time systems are dynamic and stochastic in nature and rely on dynamic scheduling algorithms to make a sequence of decisions at runtime. In (Chipalkatti et al., 1989), a number of dynamic schemes, e.g. minimum laxity threshold and queue length threshold, had been proposed. These algorithms attempted to reduce the performance degradation by making priorities of events more adaptive. An adaptive weighted round robin was also developed by (Joutsensalo et al., 2003), in which the weight parameters were updated using revenue as a target function. Another dynamic priority-driven strategy is the earliest-deadline first (EDF) (Shin, 1991), where the scheduling decisions are altered, depending on time-based competition. Even though the EDF is known to be optimal in a large class of real-time applications (Pedreiras and Almeida, 2002), it has a higher overhead and is difficult to implement since it requires updating the priority of events at each arbitration. In addition, the EDF scheduling is not adequate in the presence of load variations and nonlinearity in system dynamics. Detail descriptions of the EDF can refer to our recent publication (Chen and Yen, 2013). Moreover, there always exists certain degree of uncertainties in a variety of real-time systems that can affect the reliability of applied scheduling methods. For example, airport timetables are subject to change as flights could be modified, delayed, or canceled in the course of execution (Hildum, 1994). These uncertainties can cause the scheduling model to become either incorrect or inefficient. As a consequence, the EDF scheduling approach and many others can eventually prevent the system from accomplishing its objective efficiently.

To deal with the aforementioned problems presented in the real-life environment, the idea of combining intelligent control with

* Tel.: +886 4 8511888; fax: +886 4 8511245
E-mail address: chenms@mail.dyu.edu.tw

[1] The word “event” is a generic word that can represent many real-world phenomena such as airplanes arriving to an airport, shoppers in a grocery store, or messages in a queue waiting to be executed, etc. In this study, we focus on message scheduling.

http://dx.doi.org/10.1016/j.engappai.2014.10.002
0952-1976/

Nomenclature

The following notations provide some of the additional insights needed for understanding the background literature

$x \in R^n$ n-dimensional input vector
$x_e = [x^T, 1]^T \in R^{n+1}$ augmented input of x
$y \in R^q$ q-dimensional output vector and q is the number of queues
W_i, $W_{i,H}$, and $\overline{W}_i$ waiting time, the largest waiting time, and average waiting time of messages in queue Q_i
D_i deadline of messages in queue Q_i
δ_i difference of waiting time and deadline, i.e. W_i-D_i
φ_{ss} and φ_{ps} structure space and parameter space
A_{ik} kth fuzzy membership function of the ith fuzzy rule
c_i center of the fuzzy membership of the ith fuzzy rule
σ_{ik} width of the kth fuzzy membership function of the ith fuzzy rule
$\boldsymbol{\theta}_i^p = \{\boldsymbol{\theta}_{ik}^p\}_{k=1,2,\ldots,n}$ premise parameters of the ith fuzzy rule
$\boldsymbol{\theta}_i^c = \{\boldsymbol{\theta}_{ij}^c\}_{j=1,2,\ldots,q}$ consequent parameters of the ith fuzzy rule
τ_i firing strength of the ith fuzzy rule
$f_i = [f_{i1}, f_{i2}, \ldots, f_{iq}]^T$ consequent part of the ith fuzzy rule
ρ_i significance of the ith fuzzy rule

feedback scheduling method has emerged as a promising technology. Related literatures are reviewed. In (Matsumoto, 1994), a server in a multiple queues polling system was proposed according to the neural network policy. The simulation results showed that this method is quite beneficial, especially for asymmetric polling systems. This approach, however, is based on the assumption that the controller has the global knowledge of system states. In (Papadimitriou et al., 2004), authors also presented a learning-automata-based approach for polling in order to improve the throughput-delay performance. The learning automaton updated each client's choice probability, based on the feedback information. Siripongwutikorn et al. Siripongwutikorn et al., (2005) considered the fuzzy control concept in adaptive bandwidth control for a quantitative packet loss rate to aggregate traffic in packet switched networks. The fuzzy logic control was used to keep an average queue length at an appropriate value. Recently, Xia et al. (2008, 2009) also suggested a neural (or fuzzy) feedback scheduling scheme to optimize the overall quality of control of multitasking systems. With a modular architecture, the scheduler mainly consisted of a monitor, a predictor, a regulator, and an actuator. An Elman neural network was also employed to predict the network conditions, and then the control period was dynamically adjusted in response to network utilization. Without resorting to sophisticated control law, the included learning method prefers simpler algorithms with robust control.

Inspired by the referred researches, the authors have successfully applied the machine learning algorithm in real-time message scheduling (Chen and Yen 2011). The use of radial basis function network (RBFN) as a message scheduling controller (MSC) can effectively prevent possible causes for non-uniform bandwidth allocation among requesting messages and reduce possibilities of untimely service ratio in advance. Nevertheless, RBFN is often considered as a black-box model. This black-box behavior is a serious weakness as it prevents the extraction of relevant information to explain its internal mechanisms. This is the problem of interpretability. Instead of the RBFN approach, fuzzy rules can represent human knowledge using linguistic descriptions and benefit from the provision of explanation capabilities. As a consequence, the obtained system constitutes a promising and interpretable intelligent model rather than a black box. However, the fuzzy system may suffer from a loss of accuracy. In fact, to obtain high degree of intelligibility and high accuracy is a contradictory issue. It is difficult to find a proper trade-off between both properties. Fortunately, it is possible to combine the advantages of fuzzy rules with the learning ability of the RBFN to overcome the drawbacks of them. In this sense, the resulting neuro-fuzzy networks (NFNs) can bring the low-level learning of the RBFNs into the high-level fuzzy systems, and also provide the reasoning characteristic of fuzzy systems into the RBFNs. This paper thus substantially extends our preliminary work (Chen and Yen 2011) by applying the neuro-fuzzy networks, incorporating with structure identification and parameter learning at the same time, for multi-class message scheduling.

The rest of the paper is organized as follows. Section 2 introduces the concept and framework of the Message Queue Pool[2] (MQP) model for online multi-class message scheduling. Section 3 gives an overview of the neuro-fuzzy network. Related tunable parameters are also discussed. Detailed design procedures for NFN are described to conform to the goal of MSC in Section 4. Section 5 reports the results of simulation experiments and finally conclusions are drawn in Section 6.

2. Model descriptions

In this section, a multi-class queueing model, consisting of q finite-capacity parallel queues and a single server is addressed. Typically, messages arrive at the ith queue Q_i, $i = 1,2,,\ldots,q$, according to the independent Poisson process with rate λ_i and general independent service rate μ_i. This constitutes the Message Queue Pool. In the MQP, each queue is dedicated to a particular class of messages with the same priority and a class-specific deadline. The deadline, which is very critical in real-time systems, of a message is the point in time before which a message must be served completely. If timing constraints cannot be met, messages could miss their deadlines and cause unpredictable performance degradations. Therefore, the queue service policy is responsible to allocate limited resources by dispatching messages from the MQP to achieve the desirable quality of service (QoS). Our goal thus focuses on minimizing the message waiting times $W_i(t_k)$ at any time instant t_k, especially when messages are bounded by the deadlines. As a result, the goal of the dedicated model is given by

$$\text{minimize } \textstyle\sum_{\forall i,k} \delta_i(t_k), \text{ subject to } \delta_i(t_k) \le 0 \quad (1)$$

where $\delta_i(t_k)$ is defined as $W_i(t_k)$-D_i and D_i is the deadline of messages in Q_i. It is obvious that δ_i increases when a message remains un-served. Intuitively, the optimal solutions of Eq. (1) depend on the queue service policy. Actually, Eq. (1) belongs to the category of constrained optimization problems (Runarsson and Yao 2005) and is frequently encountered in a variety of science and engineering applications. However, the exact solutions cannot be obtained analytically. The difficulties come from the absence of satisfactory mathematical descriptions of the MQP model. In order to resolve this problem, a controller-plant configuration in Fig. 1 has been proposed to form a closed-loop feedback control (Yen 2011). A significant benefit of this scheme is to incorporate the time-varying and uncertain characteristics of the MQP into the design specifications. The message scheduling controller in Fig. 1, which can convert the constrained optimization problems into

[2] The MQP is a class-based queueing model, where each queue can accommodate messages with the same priority.