Applied Soft Computing 8 (2008) 1295–1304

Contents lists available at ScienceDirect

Applied Soft Computing

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

Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment

Key K. Lee *

Institute of Industrial Management Research, School of Management, Inje University, 617 Obang-dong, Kimhae, Kyungnam 621-749, South Korea

ARTICLE INFO

Article history:
Received 8 October 2007
Accepted 8 November 2007
Available online 27 March 2008

Keywords:
Dynamic learning classifier
Distributed fuzzy rule
Job-dispatching
Flexible manufacturing system

ABSTRACT

This paper proposes a fuzzy rule-based system for an adaptive scheduling, which dynamically selects and applies the most suitable strategy according to the current state of the scheduling environment. The adaptive scheduling problem is generally considered as a classification task since the performance of the adaptive scheduling system depends on the effectiveness of the mapping knowledge between system states and the best rules for the states. A rule base for this mapping is built and evolved by the proposed fuzzy dynamic learning classifier based on the training data cumulated by a simulation method. Distributed fuzzy sets approach, which uses multiple fuzzy numbers simultaneously, is adopted to recognize the system states. The developed fuzzy rules may readily be interpreted, adopted and, when necessary, modified by human experts. An application of the proposed method to a job-dispatching problem in a hypothetical flexible manufacturing system (FMS) shows that the method can develop more effective and robust rules than the traditional job-dispatching rules and a neural network approach.

1. Introduction

Optimizing production scheduling is one of the most important ingredients for high productivity in modern manufacturing industries. Traditional approaches to solve the scheduling problems can be classified into three categories: analytical, heuristic, and simulation approaches [1]. The analytical approach uses mathematical programming models, stochastic models, and control theory. However, it is applicable to only small-sized problems because of NP-completeness inherent to the scheduling problems [18,23].

To overcome the mathematical difficulties, heuristic approaches have often been adopted for efficiency at the cost of optimal decision. Most heuristics have focused on dispatching rules that determine the dispatching priorities of machines, automatically guided vehicles (AGVs), or jobs. While many dispatching rules have been proposed and evaluated, it remains hard to prove the general usefulness of a rule in spite of various system characteristics.

Blackstone et al. [4] showed that although no heuristic rule can always be the best in all the states of floor, some rules tended to perform consistently better than others in certain situations. Based on this finding, opportunistic adaptive scheduling strategies, which dynamically selected the most suitable rule considering the current state of the system, were proposed [3,5,13,17,20–22,26,28]. These strategies required evaluation methods that determined the best rule among candidates in a given state. Simulation was primarily used for assessing the performance of the candidate rules [6,7,28]. At each decision point, each dispatching rule was evaluated through simulation for a certain time period and the rule with the best performance is applied to decision-making in the time period. However, the opportunistic approach using simulation at every decision point is seldom used for real-time scheduling domain because of its time-consuming property.

Another method of rule selection for the adaptive or dynamic strategies would be the knowledge-based method in that decisions are made according to the mapping from the current system state to an appropriate rule using prearranged knowledge attained from the experience of the system [11]. An obvious source of such knowledge about the effectiveness of rules in different system states is the human expertise. However, extracting human scheduling knowledge of this kind is not easy since much of its quantitative part is implicit or subconscious [19]. Moreover, in very complex modern manufacturing environments, the accumulation of human expertise may be hampered by low repetition rates of the same states and noisy feedback about decisions previously made [15]. For this reason, some researchers doubt the quality of human expertise in scheduling problems and suspect that human expertise may even not exist in scheduling environments [1,24]. The need for automated knowledge acquisition stems from this difficulty of obtaining knowledge from human experts as well as the amount of data to be considered.

* Tel.: +82 55 320 3878; fax: +82 55 335 3317.
E-mail address: kiklee@inje.ac.kr.

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2007.11.005

It is the automatic acquisition of scheduling knowledge for which machine learning has been usefully employed [1,26]. Simulation has frequently been employed as a tool for the machine learning mostly because of its advantage over the use of the real field data in that a great number of trials and observations are obtainable in a short period of time [2,17,21,22]. In knowledge acquisition phase, the simulation is performed to generate a set of training data and a rule base is built from the training data set. Then the rule base is applied to real-time dynamic scheduling problems in the decision-making phase. Note that the simulations in the knowledge-based approaches and in the adaptive strategies have different objectives from each other. That is, the simulation results of the knowledge-based scheduling are used for deriving and cumulating effective scheduling rules for a subsequent reuse rather than directly used for scheduling decision.

This paper proposes a method to extract automated fuzzy rules from a continuously updated database for a knowledge-based adaptive scheduling based on the attained rule base. The knowledge-based adaptive scheduling may be thought of as a classification problem mapping from decision situations (i.e., system states) onto appropriate use of decision criteria. The decision situations are represented by state vectors of which elements are valued by fuzzy numbers. Decision-making rules are determined by choosing weights for participating decision criteria. The fuzzy rules are extracted by accumulating the simulation performance results of possible weights of the decision criteria for each state vector. The extracted rules are dynamically modified as the suitability of its fuzzy rules is refined based on the performance results, which are continuously accumulated through a feedback loop. Distributed fuzzy sets, which use multiple fuzzy numbers in concert, are employed in calculating the suitability index of the weights of criteria for the given selected state vectors. In determining the appropriateness of weights for the decision-making of adaptive scheduling, also used are distributed fuzzy sets to smooth the decision behavior [12].

Section 2 presents the specification of the problem domain and the knowledge representation of the method. The proposed knowledge acquisition and decision-making process for the dynamic adaptive scheduling are described in Section 3 and Section 4. Section 5 gives a numerical example for the proposed method. Section 6 explains the performance test results of the proposed method. Finally, conclusions are stated in Section 7.

2. The problem domain and knowledge representation

2.1. The system and job-dispatching problem

The problem domain considered in this paper is the dynamic job-dispatching problem. A typical work environment may include several workstations among which a part should be transferred for completion of a job through sequential treatments in a predefined order. When a workstation completes its current processing, it must determine the next part to process among the waiting ones. In most of dynamic manufacturing systems, such a decision is usually made by applying dispatching rules that assign priority index to each waiting part; a job with the highest priority is selected for immediate processing. Table 1 shows a list of widely used dispatching rules and their descriptions, which will be compared with the proposed method in their performance [16].

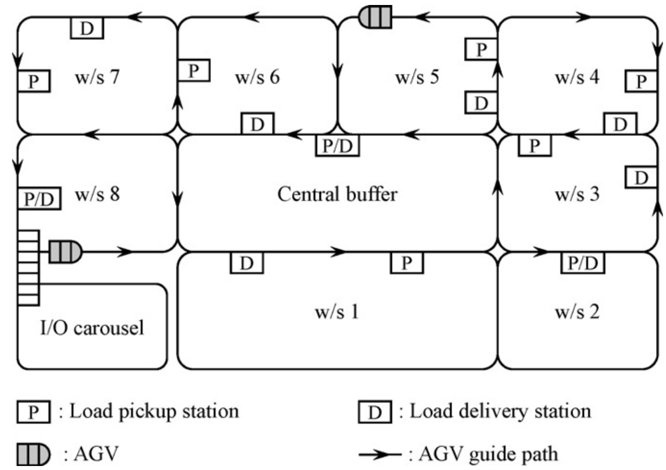

Fig. 1. A hypothetical FMS.

Fig. 1 shows a hypothetical flexible manufacturing system (FMS) that possesses the representative characteristics of the problem domain. The system consists of eight workstations, two AGVs, one input/output carousel and one central buffer for preventing a deadlock. The arriving parts are held in the input/output carousel and released into the system with FCFS dispatching rule in Table 1 only when both the input buffer space at the destination workstation and an AGV are available. All finished products are inspected in workstation 8 and exit the system through input/output carousel. The input and output buffers of workstations except workstation 8 are assumed to have limited but equivalent spaces. Each AGV transfers parts from one workstation to another along predetermined unidirectional paths. When an AGV completes a part transfer, the AGV stays at the station and waits for another move request. If both two AGVs are available for transferring a part, then the nearest AGV is selected.

2.2. Knowledge representation

The problem and solution must be based on a well-established knowledge representation. In this method, it includes the definitions of the scheduling decision criteria, the priority index

Table 1
Job-dispatching rules

Heuristics	Description
SPT	Select a part with the shortest-processing-time
SPT.TOT	Select a part with the smallest value obtained by multiplying the processing time by the job processing time
SPT/TOT	Select a part with the smallest ratio of the processing time to the job processing time
LPT	Select a part with the longest operation processing time
LPT.TOT	Select a part with the largest value obtained by multiplying the processing time by the job processing time
LPT/TOT	Select a part with the largest ratio of the processing time to the job processing time
MWKR	Select a part with the most work remaining
LWKR	Select a part with the least work remaining
FCFS	Select a part that arrived at the workstation first
FAFS	Select a part that entered in the system first
LWNQ	Select a part with the least work in next workstation queue
RANDOM	Job priority is random