Available online at www.sciencedirect.com

Engineering Applications of Artificial Intelligence 19 (2006) 247–255

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

A comparison of machine-learning algorithms for dynamic scheduling of flexible manufacturing systems

Paolo Priore*, David de la Fuente, Javier Puente, José Parreño

Escuela Técnica Superior de Ingenieros Industriales, Universidad de Oviedo, Campus de Viesques, 33203, Gijón, Spain

Received 3 November 2004; received in revised form 16 September 2005; accepted 26 September 2005
Available online 14 November 2005

Abstract

Dispatching rules are frequently used to schedule jobs in flexible manufacturing systems (FMSs) dynamically. A drawback, however, to using dispatching rules is that their performance is dependent on the state of the system, but no single rule exists that is superior to all the others for all the possible states the system might be in. This drawback would be eliminated if the best rule for each particular situation could be used. To do this, this paper presents a scheduling approach that employs machine learning. Using this latter technique, and by analysing the earlier performance of the system, 'scheduling knowledge' is obtained whereby the right dispatching rule at each particular moment can be determined. Three different types of machine-learning algorithms will be used and compared in the paper to obtain 'scheduling knowledge': inductive learning, backpropagation neural networks, and case-based reasoning (CBR). A module that generates new control attributes allowing better identification of the manufacturing system's state at any particular moment in time is also designed in order to improve the 'scheduling knowledge' that is obtained. Simulation results indicate that the proposed approach produces significant performance improvements over existing dispatching rules.

Keywords: Knowledge-based systems; Scheduling; FMS

1. Introduction

The different approaches available to solve the problem of flexible manufacturing system (FMS) scheduling can be divided into the following categories: the analytical, the heuristic, the simulation-based and the artificial intelligence-based approaches. The analytical approach considers an FMS scheduling problem as an optimisation model with an objective function and explicit constraints. An appropriate algorithm resolves the model (see for example, Stecke, 1983; Kimemia and Gershwin, 1985; Shanker and Tzen, 1985; Lashkari et al., 1987; Han et al., 1989; Hutchison et al., 1989; Shanker and Rajamarthandan, 1989; Wilson, 1989). In general, these problems are of a NP-complete type (Garey and Johnson, 1979), so heuristic and off-line type algorithms are usually used (Cho and Wysk, 1993; Chen and Yih, 1996). The problem is that these analytical models include simplifications that are not always valid in practice. Moreover, they are not efficient for reasonably large-scale problems.

These difficulties led to research into many heuristic approaches, which are usually dispatching rules. These heuristics employ different priority schemes to order the diverse jobs competing for the use of a machine. A priority index is assigned to each job and the one with the lowest index is selected first. Many researchers (see for example, Panwalkar and Iskander, 1977; Blackstone et al., 1982; Baker, 1984; Russel et al., 1987; Vepsalainen and Morton, 1987; Ramasesh, 1990; Kim, 1990) have assessed the performance of these dispatching rules on manufacturing systems using simulation, concluding that performance depends on several factors, such as the due date tightness (Baker, 1984), the system's configuration, the workload, and so on (Cho and Wysk, 1993). FMSs led to many studies evaluating the performance of dispatching rules in

*Corresponding author. Tel.: +0034985182107; fax: +0034985182010.
E-mail address: priore@epsig.uniovi.es (P. Priore).

0952-1976/$ - see front matter
doi:10.1016/j.engappai.2005.09.009

these systems (see for example, Stecke and Solberg, 1981; Egbelu and Tanchoco, 1984; Denzler and Boe, 1987; Choi and Malstrom, 1988; Henneke and Choi, 1990; Montazeri and Wassenhove, 1990; Tang et al., 1993).

Given the variable performance of dispatching rules, it would be interesting to change these rules dynamically and at the right moment according to the system's conditions. Basically, there are two approaches to modifying dispatching rules dynamically in the literature. Firstly, the rule is chosen at the right moment by simulating a set of pre-established dispatching rules and selecting the one that gives the best performance (see for example, Wu and Wysk, 1989; Ishii and Talavage, 1991; Kim and Kim, 1994; Jeong and Kim, 1998). The second approach, from the field of artificial intelligence, employs a set of earlier system simulations (training examples) to determine what the best rule is for each possible system state. These training cases are used to train a machine-learning algorithm (Michalski et al., 1983) to acquire knowledge about the manufacturing system. Intelligent decisions are then made in real time, based on this knowledge (see for example, Nakasuka and Yoshida, 1992; Shaw et al., 1992; Kim et al., 1998; Min et al., 1998). A set of control attributes need to be established which identify the manufacturing system's state at each particular moment in time in order to generate training examples. Aytug et al. (1994) and Priore et al. (2001) provide a review in which machine learning is applied to solving scheduling problems. Finally, it has to be noted that many works take benefit from a combination of two or more approaches (see for example, Glover et al., 1999; Flanagan and Walsh, 2003).

Machine learning solves problems by using knowledge acquired while solving earlier problems in the past similar in nature to the problem at hand. The main algorithm types in the field of machine learning are case-based reasoning (CBR), neural networks and inductive learning. The right training examples and machine-learning algorithm must be selected if this scheduling approach based on machine learning is to work properly. However, there are hardly any studies in the literature dealing with this problem. This paper thus aims to compare three of the above-mentioned machine-learning algorithms. In an attempt to improve the manufacturing system's performance, a new approach is also proposed whereby new control attributes that are arithmetical combinations of the original attributes can be determined.

The rest of this paper is organised as follows. Machine-learning algorithms used in this paper are first described. An approach to scheduling jobs that employs machine learning is then presented. The experimental study describing a new approach to determining new control attributes from the original ones now follows, along with a comparison of the machine-learning algorithms. The proposed scheduling approach is then compared with the alternative of using a combination of dispatching rules constantly. A summary of the results obtained rounds the paper off.

2. Machine learning

According to Quinlan (1993), machine-learning algorithms can be classified into the following categories: CBR, neural networks and inductive learning. A brief description of the machine-learning algorithms applied in this work will be provided next. The most commonly used CBR algorithm is the nearest neighbour algorithm (Aha, 1992). The formulation of this algorithm, called NN, or k-NN in the more sophisticated version, is very simple. The starting point is a metric, d, amongst examples, and a set of training examples, E. When a new case x occurs, its solution or class is determined as follows:

$$\mathrm{Class}(x) = \mathrm{class}(\mathrm{nn}),$$

where nn (nearest neighbour) satisfies:

$$d(x, \mathrm{nn}) = \mathrm{Minimum}\{d(x, e) : e \in E\}.$$

This means that the case's class coincides with that of the nearest example or neighbour. This initial scheme can be fine tuned by including an integer value, k ($k \geqslant 1$), which determines the k nearest neighbours in E for x. The class of x is therefore a function of the class of the majority of the k neighbours.

As regards neural networks, those that will be used in this work are 'backpropagation neural networks (BPNs)', or multilayer perceptron (Rumelhart et al., 1986). These are one of the most well-known and widely used as pattern classifiers and function approximators (Lippman, 1987; Freeman and Skapura, 1991). Fig. 1 gives an overview of a neural network of this type. As can be seen, there is a single hidden layer and there are no connections between neurons in the same layer in this particular case.

The backpropagation training algorithm is the one that is used in this type of neural networks. There are several versions of this algorithm, and the standard one (Rumelhart et al., 1986; Freeman and Skapura, 1991) will next be commented on. We will assume a network with an input layer of n_1 neurons, a hidden layer of n_2 neurons, and an output layer of n_3 neurons. The outputs of the input, hidden and output layers are called x_i, y_j and z_k, respectively. The weights of the connections that connect

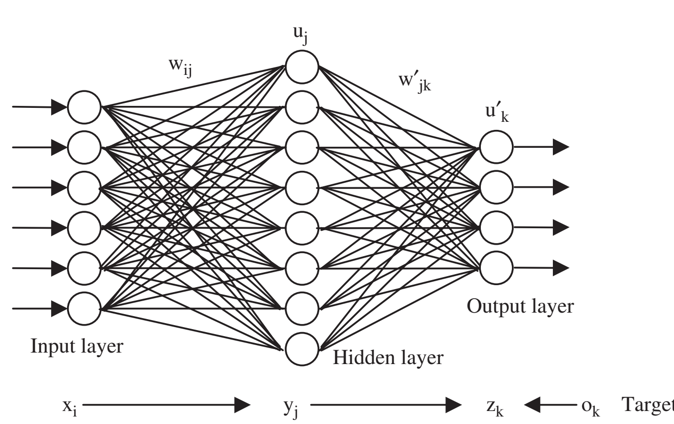

Fig. 1. An overview of a backpropagation neural network.