ELSEVIER

Robotics and Autonomous Systems 33 (2000) 169–178

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Dynamic job-shop scheduling using reinforcement learning agents

M. Emin Aydin [a], Ercan Öztemel [a,b,*]

[a] *Department of Industrial Engineering, Sakarya University, 54040-Esentepe Campus, Adapazari, Turkey*
[b] *Tubitak-Marmara Research Center, BTAE, Artificial Intelligence Group, Gebze, Kocaeli, Turkey*

Received 1 January 1999; received in revised form 1 May 1999; accepted 1 December 1999

Abstract

Static and dynamic scheduling methods have attracted a lot of attention in recent years. Among these, dynamic scheduling techniques handle scheduling problems where the scheduler does not possess detailed information about the jobs, which may arrive at the shop at any time. In this paper, an intelligent agent based dynamic scheduling system is proposed. It consists of two independent components: the agent and the simulated environment. The agent selects the most appropriate priority rule according to the shop conditions in real time, while simulated environment performs scheduling activities using the rule selected by the agent. The agent is trained by an improved reinforcement learning algorithm through the learning stage and then it successively makes decisions to schedule the operations. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Intelligent agents; Reinforcement learning; *Q*-III learning; Dynamic job-shop scheduling

1. Introduction

Scheduling, one of the key problems in manufacturing systems, has been a subject of interest for a long time. However, it is difficult to talk about a method that gives optimal solutions for every problem that emerges. The problem is to schedule a set of jobs subject to a set of constraints where each job consists of a set of operations. The aim is to get an appropriate schedule in terms of a certain criterion.

Since previous studies have considered the set of jobs as having all required information at initial time, and hence most of the methods scheduled the jobs in a static manner. On the other hand, the relation between jobs and shop floor is not so static that the systems proposed in that manner are not suitable in real life. In fact, each job comes into shop over time and the required information is uncertain in most cases. Thus, a dynamic scheduling system is more suitable than a static one. Dynamic systems start with the jobs that come first, and assume that they come according to a stochastic rule over time.

In order to build dynamic scheduling systems, several methods have been proposed so far. Some studies have focused on dynamic scheduling for flexible manufacturing systems. Yih and Thesen [40] considered the real-time scheduling system for an FMS as a semi-Markovian decision process to be optimized. Ishii and Talavage [15] generate short-term schedules for an FMS, while Arzi [1] suggests a two-step dynamic scheduling algorithm for such systems. Similarly, Matsuura et al. [22] proposed a switching technique for dynamic scheduling allowing consideration of machine break-downs and other emergent events. Most of the studies were also performed for generic

* Corresponding author. Present address: Department of Industrial Engineering, Sakarya University, 54040-Esentepe Campus, Adapazari, Turkey.
E-mail address: eoman@mam.gov.tr (E. Öztemel).

0921-8890/00/$ – see front matter
PII: S0921-8890(00)00087-7

systems. For example, Sun and Lin [34] viewed the scheduling system as an optimal control problem of discrete events and scheduled the jobs using a backward scheduling algorithm.

On the other hand, there are some approaches developed based on artificial intelligence techniques such as neural networks, expert systems, fuzzy logic and genetic algorithms. Chang [10] developed a rule-based system that proposes incremental dispatching rules. Sim et al. [30] combined ES and NN for generating the most appropriate schedule in the current state. Both Shaw et al. [29] and Nakasuka and Yoshida [23] used a second generation ES model that acquires its knowledge automatically. In all of these approaches, the most appropriate dispatching rule is proposed. Genetic algorithms (GAs) are also used extensively for JSS. Bierwirth et al. [7] and Lin et al. [17] adapted GA to the Giffler and Thompson algorithm and constructed dynamic schedules.

The literature review indicates that there has been little work on creating intelligent autonomous scheduling systems with a learning ability based on trial and error. In this study, an intelligent agent based scheduling system is proposed aiming at the generation of a more autonomous scheduler where the agent is trained by a new improved reinforcement learning algorithm, Q-III.

In the following sections, first intelligent agents and then the Q-III learning algorithm are presented. Thereafter, details of the intelligent agent based scheduling system are discussed using the simulation results.

2. Intelligent agents and JSS

Intelligent agents are autonomous systems which can perform appropriate intelligent actions using their own knowledge in dynamic environments [12,13,20,33]. They are mainly composed of three parts; perception, cognition and action. An intelligent agent receives messages from the environment via its *perception* mechanism. These messages are then evaluated by the *cognition* system and appropriate actions are produced and implemented by the *action* module. Since the aim of this paper is not to discuss these aspects of the agents in detail, the relevant information is omitted. The readers are referred to [28,39] which are extremely well organized for more information.

Intelligent agent based JSS systems have been the subject of research for a number of years now. Most of the studies deal with static JSS using multiple agents. In this case the aim is to explore an applicable solution using various search and distributed computing techniques [9,11]. On the other hand, Zhang and Dietterich [41] proposed a reinforcement learning based search approach for specific scheduling problems that are also static. This paper presents a novel approach, which deals with dynamic scheduling using a reinforcement learning algorithm.

3. A reinforcement learning algorithm: Q-III

Learning is one of the most important topics in research on intelligent agents [8]. In particular, reinforcement learning techniques are widely employed [6,14,19,35–37]. With these techniques, the agent has to take into account a reinforcement signal, which is produced against its actions. Well known reinforcement learning algorithms are TD(λ) and Q-learning. There are a lot of successful implementations of these algorithms in different domains [3,16,18,21,26,31]. Among these, Q-learning algorithm works in such a way that the agent gains experience by trial and error throughout the execution of the actions. During this process, the agent figures out how to assign credit or blame to each of its actions in order to improve its behavior. This is called *temporal credit assignment*. Once the agent learns how to behave, it may generalize the knowledge it possesses and recall the knowledge when required. The ability of the agent to use its past experience (generalization) is called *structural credit assignment*. However, Q-learning is only able to handle temporal credit assignment. Furthermore, the Q-learning procedure suffers from several problems. For example, as discussed by Whitehead and Lin [38], traditional Q-learning was developed utilizing a Markovian decision process. Therefore, it may not be implemented as successfully as expected in non-Markovian domains. There may be two problems: the learning process may end up with a local optimum solution or may take too long to succeed. Some attempts have already been made to solve these problems. Öztemel and Aydin [24] developed an algorithm called Q-I to solve the “local optimum” problem. However, the learning time still remained too