PERGAMON

Available online at www.sciencedirect.com
SCIENCE DIRECT

Computers & Industrial Engineering 45 (2003) 215–229

computers & industrial engineering

www.elsevier.com/locate/dsw

Scheduling for on-time completion in job shops using feasibility function[☆]

Attila Lengyel[a,*], Itsuo Hatono[b], Kanji Ueda[c]

[a]*MI-2 Laboratory, Faculty of Engineering, Kobe University 1-1, Rokkodai, Nada, Kobe 657-8501, Japan*
[b]*Information Processing Center, Kobe University1-1, Rokkodai, Nada, Kobe 657- 8501, Japan*
[c]*RACE (Research into Artifacts, Center for Engineering), The University of Tokyo, Komaba, 4-6-1, Meguro-ku, Tokyo 153-8904, Japan*

Accepted 20 March 2003

Abstract

This article focuses on job shop type manufacturing systems where a high variety of products of different volumes are requested to be produce on a tight due date. The feasibility function (FF) is addressed in this article to schedule jobs in multi-machine random job shop, where the purpose is to minimize unit penalty by achieving a balance between the number of tardy and early jobs, and reducing the difference between the maximum and the minimum lateness of jobs. A job shop simulation model based on multi-agent architecture developed by the authors provides an environment for comparing the FF to commonly used dispatching rules. The results show the benefit of using the FF. Discussions reveal that this concept is more reliable in case of due dates with different tightness level.

Keywords: Job shop; Due date based scheduling; On-time completion; Feasibility function; Multi-agent architecture

1. Introduction

Reliable delivery dates have been of capital importance in make-to-order manufacturing environment since the beginning of production. Before the industrial revolution a craftsman, a member of a guild created a unique product by himself fully satisfying the costumer's requests, planning and scheduling were based on intuitions. Quality and reliable delivery dates represented the trademark of the craftsman.

☆ This manuscript was processed by Area Editor Maged Dessouky
* Corresponding author. Fax: +81-78-803-6155.
E-mail address: lengyel@mi-2.mech.kobe-u.ac.jp (A. Lengyel).

0360-8352/03/$ - see front matter
doi:10.1016/S0360-8352(03)00042-1

Present manufacturing companies with make-to-order environment yearn for flexibility, quality and ability of customizing these according to the customer's requirements. The modern market of high standard requires high variety of qualitative products, on-time delivery, minimal work in process, short costumer lead times, and maximum utilization of resources. Unfortunately, these goals are conflicting. The goal of the modern production scheduling is to strike a profitable balance among these conflicting objectives (Hopp, Wallace, Spearman, & Mark 1996). The more complex the product and the manufacturing system are the more complex planning and control techniques are needed, intuitions are not effective enough anymore.

One of the basic objectives of production scheduling in make-to-order environment is to meet the due dates. Due-date based scheduling of manufacturing systems has been widely examined in the related literature. There are several empirical and mathematical models to schedule jobs in shop floor to obtain better due date performance. Dynamic priority rules are the most used scheduling algorithms in case of due date related real-time scheduling. Scheduling problems can be defined by three separate elements: the machine environment, the optimal criterion, and a set of side constraints and characteristics. In single machine environment earliest due date (EDD) is an exact algorithm for minimizing the maximum lateness (Jackson, 1955). Dertouzos (1974) proved that in a context of queuing system, the EDD algorithm is optimal in a following sense: if any algorithm can schedule a particular task set without missing any deadline, the EDD algorithm can schedule it as well. Mok (1983) proved that the last slack time (LST) algorithm could also complete the above. If we have more than one machine, scheduling problems become more complicated. This article is focused on due-date based dynamic scheduling in job shop type manufacturing systems. Job shops produce small lots with high variety of routings through the plant with different machines. Each job has its own route to follow and unique requirements. Recent research results in the field of job shop scheduling for due date performances point out that the combination of the shortest process time first (SPT) and the critical ratio with allowance (ALL + CR + SPT) scheduling algorithms gives the best performances (Cheng & Jiang, 1997). Due to the complexity of job shop type manufacturing systems benchmark problems are still under development. The first expedient benchmark concept for minimizing maximum lateness was introduced by Demirkol, Mehta, and Uzsoy (1996). Later discussion will point out that in case of dynamic scheduling a realistic simulation model with long-term simulation runs may represent a job shop behavior more genuinely than benchmark problems.

The commonly used due date performance measures are unit penalty, mean tardiness, maximum and minimum lateness. It is known that tardy jobs may incur tardiness costs, such as contractual penalties, depending on how late they are. According to the new production technologies, early jobs may incur earliness costs as well, such as inventory holding costs (Yeunga, Oguz, & Cheng, 2001). Thus, minimizing average earliness of jobs has become more important in production scheduling research (Baker, & Scudder, 1989; Lee, Liman, & Lin, 1991). For this performance the minimizing mean absolute and square lateness measurements are used.

The purpose of the presented research work is to develop a scheduling algorithm that minimizes unit penalty by balancing between the number of tardy and early jobs and reducing the difference between the maximum and the minimum lateness of jobs by its nature. This article first introduces a new due date related performance measurement, and then brings out a new scheduling algorithm, called feasibility function (FF). The environment of comparing the FF with the other real-time scheduling algorithms is a