Computers & Operations Research 39 (2012) 1305–1314

Contents lists available at ScienceDirect

Computers & Operations Research

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

A cooperative dispatching approach for minimizing mean tardiness in a dynamic flowshop

Ahmed El-Bouri

Department of Operations Management & Business Statistics, College of Commerce and Economics, Sultan Qaboos University, PO Box 20, PC 123, Al Khod, Oman

ARTICLE INFO

Available online 13 July 2011

Keywords:
Scheduling
Flowshop
Heuristics
Dispatching rules
Mean tardiness

ABSTRACT

Cooperative Dispatching is a real-time scheduling methodology, which consults downstream machines before making a job dispatching decision on any given machine. This paper proposes such an approach for minimizing the mean tardiness in a dynamic flowshop where new jobs arrive continuously, at random points in time, throughout the production cycle. Cooperative Dispatching is based on the idea that individual machines act self-interestedly, with the objective of optimizing their local performance criteria. A consulted machine attempts to influence upstream dispatching decisions in a manner that promotes its ability to minimize its total local tardiness. A machine's influence in the dispatching decision depends on current congestion and due-date tightness levels in the shop. A multiple regression model is proposed to help determine the weight a consulted machine's preferences will carry in the dispatching decision. Conflicting demands from the different machines are resolved by a minimum regret decision procedure, which aims to minimize the aggregate deviation from the consulted machines' preferences. The winning candidate that ultimately emerges from this procedure is the job that is dispatched. A comparative analysis to evaluate the performance of cooperative dispatching, compared to six other dispatching rules that are commonly favoured for tardiness-based criteria, is performed by means of simulation, using randomly generated test problems. Computational results indicate that Cooperative Dispatching outperforms the other dispatching rules, across a broad range of flowshop congestion and due-date tightness levels.

1. Introduction

The dynamic flowshop is a production scheduling problem where each arriving job requires m operations on m different machines, and in the same order. Specifically, the kth operation for each job will be processed uniquely on machine k. Hence, the flowshop is best described as a production system consisting of m machines arranged in series, through which all jobs follow the same route, beginning at machine 1 and ending at machine m. The flowshop environment under consideration here is a dynamic one, in the sense that new jobs continue to arrive over a rolling horizon, even as existing jobs are either in progress or waiting to be started. Every job has specific processing time requirements on the machines, as well as a due-date by which it must be completed. A job's arrival time is not known in advance, and its processing times and due-date become known only upon its arrival.

The flowshop in this study utilizes input buffers to hold waiting jobs ahead of each machine. From the second machine onwards, the input buffers are known as intermediate buffers, in that they hold in-process jobs. After completion on one machine, a job is transported to join the input queue at the next machine. If that machine is free, the job is loaded immediately, otherwise it waits in the buffer. The scheduling problem is to decide in what order the waiting jobs at each machine should be processed, so that the mean tardiness is minimized. Tardiness is the amount by which a job's completion time exceeds its due-date. Any job completed on or before its due-date accumulates no tardiness. An optimal schedule is one that minimizes the mean tardiness measure for all completed jobs.

A popular and convenient scheduling technique in dynamic flowshops is to use dispatching rules. A dispatching rule prioritizes queued jobs at a machine. Whenever a machine becomes free to service a job, the dispatching rule decides which of the waiting jobs to process next. There are numerous rules that prioritize jobs according to different criteria. Although several particular rules have been developed for minimizing the mean tardiness, there is no single rule that dominates all others. The main advantage in using dispatching rules in dynamic environments, however, is their ability to make their dispatching decisions based on local machine data. This means that new jobs may be scheduled in real-time, locally and without a need to revise or re-establish a new global schedule each time a new job arrives.

E-mail address: aelbouri@squ.edu.om

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.07.004

The trade-off, however, is that schedules resulting from dispatching rules are much weaker in satisfying the performance criteria, when compared to centralized schedules that use global data from the whole flowshop.

This paper presents a cooperative dispatching methodology for minimizing mean tardiness in dynamic flowshops. Cooperative dispatching (CD) aims to improve the quality of the local data on which dispatching decisions are made, by injecting global information collected from other machines that are potentially affected by the present dispatching decision. When a dispatching decision is needed at a given machine, *s*, CD determines which of the waiting jobs to load next, but only after consulting all machines that lie downstream of machine *s*. A consulted machine returns its degree of preference for each candidate job, based on the consequence that immediate dispatching of that candidate has on the consulted machine's performance objective of minimizing the local total tardiness for the jobs it processes. Conflicting preferences received from the consulted machines, if any, are then resolved by a minimum 'regret' procedure before a candidate job is ultimately selected and dispatched on machine *s*.

A review of the current literature in dynamic flowshop scheduling research is presented in the next section, followed in Section 3 by a detailed methodology of the proposed cooperative dispatching approach. Section 4 presents results from a computational study that compares cooperative dispatching and a number of selected dispatching rules for minimizing mean tardiness. Finally, conclusions and recommendations for future research are discussed in Section 5.

2. Literature review

The problem of minimizing mean tardiness in a flowshop is NP-hard in the strong sense [1,2]. As a result, the use of exact solution methods has been limited mostly to branch and bound approaches for small, 2-machine flowshops [3–5]. For larger flowshops, heuristics are usually preferred. Kim [6], for example, used list scheduling techniques, including the modified due date dispatching rule, to create initial solutions that are then improved by local search. Raman [7] adapted several single and two-machine flowshop heuristics to multi-machine flowshops.

With recent advances in computational speed, meta-heuristics like simulated annealing, genetic algorithms, ant colony methods and tabu search have been suggested for the mean tardiness criterion (a comprehensive review is available in Vallada et al. [8]). However, these have been directed primarily at static flowshops, where all of the jobs are present at the beginning of the scheduling horizon. Of more significance in real-world industrial and service applications is the dynamic version of the flowshop, in which the scheduling is done in the presence of real-time events that render pre-established schedules obsolete. In their survey of dynamic scheduling, Ouelhadj and Petrovic [9] covered three categories of dynamic scheduling, namely: completely reactive, reactive-predictive, and robust pro-active scheduling. In completely reactive scheduling, dispatching decisions are made locally at the machines in real-time, and no advance schedule exists. The cooperative dispatching approach presented in this paper belongs to this category.

Research in completely reactive scheduling focuses mainly on applying dispatching rules [9]. Although there is a large body of research in dispatching rules, it is mostly concerned with job shop environments [10–12] and flexible manufacturing systems [13]. With respect to flowshops, the earliest studies of dispatching rules investigated simple rules such as SPT, EDD, MDD and FIFO [14–18]. Rajendran and Holthaus [19] presented a comparative study of some common dispatching rules for several different performance criteria in both flowshop and jobshop manufacturing environments. Their results identified one of a number of rules that they proposed, along with the COVERT dispatching rule, as the most effective in minimizing mean tardiness under different conditions of due-date tightness. In Rajendran and Alicke [20], a number of dispatching rules were developed for minimizing total tardiness in a flowshop that has bottleneck machines.

Dispatching rules have also been considered for other tardiness-based criteria. Swaminathan et al. [21] investigated the apparent tardiness cost (ATC) rule for minimizing total weighted tardiness for flowshops, in which new jobs arrive at the beginning of every shift. Lodree et al. [22] investigated minimizing the number of tardy jobs by applying a variation of the Moore–Hodgson algorithm at individual machines, based on job operation due-dates.

The advantage of dispatching rules is that they are quick and easy to implement in a real-time environment. However, most of the common dispatching rules are myopic, in that their dispatching decisions are based on local machine information, rather than global flowshop data. As a result, their performance in most situations is far from optimal. Various approaches for improving the quality of local dispatching decisions have been researched, including agent-based scheduling [9], neural networks [23] and cooperative dispatching [24].

The study presented in this paper extends the cooperative dispatching approach [24] to the mean tardiness criterion, and compares its performance to a number of alternative dispatching rules. The comparisons are performed by applying cooperative dispatching and the other rules to randomly generated test problems, using computer simulation models of 5- and 10-machine flowshops.

3. Cooperative dispatching

Given a flowshop composed of m machines, and jobs that arrive continuously over the scheduling period, the scheduling problem is to decide in what order to process the jobs on each of the machines, such that the mean tardiness of all completed jobs, $\overline{T}$, is minimized.

Let

D_j=due-date for job j.

N=total number of completed jobs.

C_j=completion time of job j's final operation.

The mean tardiness is:

$$\overline{T} = \frac{\sum_{j=1}^{N} (C_j - D_j)^+}{N} \quad (1)$$

where $(C_j - D_j)^+ = C_j - D_j$ if $C_j > D_j$; 0 otherwise.

The m-machine flowshop under consideration assumes that part overtaking in intermediate buffers is permitted; set-up times, machine loading/unloading times, and transportation times between machines and buffers are all negligible; buffers have unlimited storage capacities; and machines will not remain idle while there are any parts waiting in their input buffers.

The following notation is used in presenting the Cooperative Dispatching (CD) methodology:

s machine where the current dispatching decision is needed (also called the 'dispatching' machine).
m total number of machines in the flowshop.
k index for machine.
j index for job identification number.
i index for queued jobs at machine s.
n number of jobs queued at machine s.
Ω ordered set of jobs currently in queue for processing on machine s.