Int. J. Production Economics 141 (2013) 360–365

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

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

A flexible dispatching rule for minimizing tardiness in job shop scheduling

Binchao Chen, Timothy I. Matis *

Department of Industrial Engineering, Texas Tech University, Box 43061, Lubbock, TX 79409, United States

ARTICLE INFO

Article history:
Received 25 April 2012
Accepted 29 August 2012
Available online 7 September 2012

Keywords:
Job shops
Scheduling
Dispatching rules
Tardiness
Simulation

ABSTRACT

In this paper, a dispatching rule called the Weight Biased Modified RRrule is developed that minimizes the mean tardiness of weighted jobs in an m-machine job shop, i.e. $J_m|r_i,recrc|\sum_i T_{ik}$ where T_{ik} denotes the tardiness of those jobs with weight greater than a specified threshold level k. It is a significant extension of the RRrule in that it has linear complexity and considers weighted jobs. In addition, the WBMR rule allows for biasing of the schedule towards meeting the deadline of high priority jobs through the tuning of a single parameter, where such an effect is quantified by evaluating tardiness at different truncation thresholds. Numerical testing demonstrates the ability of the WBMR to outperform other traditional rules at various congestion and due-date tightness levels.

Published by Elsevier B.V.

1. Introduction

The objective of this paper is to develop a dispatching rule that heuristically minimizes the tardiness of weighed jobs in an m-machine job shop with unequal ready time and recirculation. In shorthand notation, this scheduling problem is represented as $J_m|r_i,recrc|\sum_i T_{ik}$, where T_{ik} denotes the tardiness of those jobs with weight greater than a specified threshold level k. In general, this problem is known to be strongly NP-hard (Zhou et al., 2009) and hence exact algorithms for optimally scheduling such, including integer programming (Manne, 1960), dynamic programming (Held and Karp, 1962) and the branch and bound method (Brucker et al., 1994), are not practical for large-scale problem instances. The use of improvement-based heuristics as an alternative approach to this problem, including evolutionary algorithms (Nakano, 1995; Yamada and Nakano, 1997), tabu search (Dell'Amico and Trubian, 1993; Barnes and Chambers, 1995), simulated annealing (Catoni, 1998; Laarhoven and Aarts, 1992) and cultural algorithm (Rivera et al., 2007), may also suffer from scalability issues for very large problems, and are not practical when the schedule is rolling. As such, a natural alternative for such cases are construction-based heuristics that have linear complexity, which are frequently referred to as dispatching rules. Dispatching rules assign a priority to jobs waiting in a queue according to some combination of the jobs characteristics and common system level parameters, and then picks the job with the highest priority. Popular rules include first in first out (FIFO), shortest process time (SPT), apparent tardiness cost (ATC), etc. Dispatching rules typically do not produce a schedule that finds a global optimum and at times fail miserably, yet they are simple to implement, compute very fast, and typically find a good solution somewhat near the optimum, which make them ideal for rolling schedules in a large scale system.

The development of this dispatching rule was motivated by our work with the tooling operations of a heavy manufacturing firm that could be represented as a large scale make to order job shop. It was not uncommon for these tooling operations to have work in progress on the order of 10^4 jobs with up to 90% of these being unique. Work orders for new and reworked tools were constantly being added to the schedule, where orders varied in relative weight that was statically assigned based on the associated manufactured product. Production priorities, however, dynamically shifted emphasis between meeting deadlines for all jobs and for only high priority jobs, where such was typically determined at a daily operations meeting based on customer demands and the stage of processing of associated products. In response, we developed a dispatching rule called the Weight Biased Modified RRrule (WBMR) that is flexible in targeting tardiness minimization at various specified priority threshold levels, and that is robust to varying degrees of system level congestion and tightness of the due dates. The basis for the WBMR rule lies in the observation that processing time-based rules perform better at minimizing mean tardiness when due dates are tight, and due-date-based rules perform better when due dates are loose. It is an extension of the RRrule developed by Raghu and Rajendran (1993), yet differs in that it considers job weights and is a true construction-based heuristic with linear complexity. In addition, the WBMR rule allows for the biasing of the schedule to meet production priorities through the adjustment of a

* Corresponding author. Tel.: +1 806 742 3543.
E-mail addresses: binchao.chen@ttu.edu (B. Chen), timothy.matis@ttu.edu (T.I. Matis).

0925-5273/$ - see front matter Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.ijpe.2012.08.019

single parameter, which in our case would be by a production manager on a daily basis in response to the operations meeting. In the remainder of this paper, modeling preliminaries are given in Section 2, followed by the literature review in Section 3. The WBMR rule is developed in Section 4, numerical testing in Section 5, and concluding remarks in Section 6.

2. Modeling preliminaries

The following notation and modeling assumptions are used throughout this paper.

2.1. Notation

n	total number of jobs
M	total number of machines
i	job number, $i \in \{1,2,\ldots,n\}$
N_i	number of operations for ith job
j	operation number, $j=1,2,\ldots, N_i$
ω_i	weight of job i
η	utilization of system
d_i	due date of job i
t	current time
p_{ij}	process time of ith job for jth operation
I_{ij}	the priority index of ith job for jth operation
AT_i	arrival time of job i
c_i	completion time of job i
s_i	slack of job i
RPT_i	remaining process time of job i
λ	the arriving rate of a job
A	allowance factor
β	weight biasing parameter

2.2. Modeling assumptions

Assumptions about the characteristics of the M-machine job shop considered in this paper are typical of those commonly defined for such when there are no special cases. In particular, each machine operates independently and can perform only one operation at a time on a job. Once taken up for processing on a machine, each job is processed to completion, i.e. there is no job preemptions or machine vacations. Each job has a specified sequence of machines it should visit. In addition, each job has an associated weight that is assigned as a parameter.

3. Literature review on dispatching rules

Popular dispatching rules and a description of their performance strengths are listed in this section for reference purposes. These will be used for comparative purposes when testing the performance of the WBMR.

3.1. FIFO (first in first out)

This rule is often used as a benchmark. Highest priority is given to the waiting operation that arrived at the queue first.

3.2. SPRT (slack per remaining process time)

The job with the least slack per remaining process time is chosen for loading. This rule is extensively used in industry and often used as bench mark for evaluating tardiness-related measurements (Bulkin et al., 1966; Putnam et al., 1971; Anderson and Nyirenda, 1990). The priority of the jth operation of job i is given as

$$I_{ij} = s_i/RPT_i, \tag{1}$$

where $s_i = d_i - t - \sum_{k=j}^{N_i} p_{ik}$, and $RPT_i = \sum_{k=j}^{N_i} p_{ik}$. The job with the least I_{ij} is loaded on the machine. Notice that slack s_i is usually greater than 0. If the slack is a negative value, then the job has already been tardy regardless of the follow up schedule.

3.3. WSPT (weighted shortest process time)

This rule is one of the most commonly used rules for jobshop scheduling and is found to be very effective in minimizing mean tardiness under heavily load shopfloor condition (Conway, 1965; Blackstone et al., 1982; Haupt, 1989). The priority is given as

$$I_{ij} = \omega_i/p_{ij}, \tag{2}$$

and the job with the largest I_{ij} is loaded on the machine.

3.4. WATC (weighted apparent tardiness cost)

Vepsalainen and Morton (1987) proposed the apparent tardiness cost (ATC) rule using an exponential function of job slack. The rule can be expressed as

$$I_{ij} = \frac{\omega_i}{p_{ij}} \times \exp\left\{ -\max\left[\frac{d_i - t - p_{ij} - \sum_{q=j+1}^{N_i}(W_{iq}+p_{iq})}{k\overline{p}}, 0 \right] \right\}, \tag{3}$$

where k is a lookahead parameter, with a suggested range between 1.5 and 4.5, which scales the slack according to the expected number of competing jobs and $\overline{p}$ is the average process time of the waiting jobs. The job with the maximum value of I_{ij} is chosen for loading. In the simulation study, we fix k at 3, and the estimated waiting time for each operation, W_{iq}, is calculated according to the MOD rule by Anderson and Nyirenda (1990).

3.5. RRrule (Raghu and Rajendran rule)

Previous studies have shown that processing time-based rules perform better under tight due-date conditions where as due-date-based rules are found to perform better under loose due-date conditions (Conway, 1965; Rochette and Sadowski, 1974; Ramasesh, 1990). This observation lays the foundation for the RRrule developed by Raghu and Rajendran (1993), where the priority index (I_{ij}) is calculated as

$$I_{ij} = \frac{s_i}{RPT_i} \exp(-\eta)p_{ij} + \exp(\eta)p_{ij} + W_{nxt}, \tag{4}$$

where the job with the least I_{ij} is loaded on the machine. This rule may be thought of as a linear combination of the SPRT and SPT rules with the respective weights of $\exp(-\eta)$ and $\exp(\eta)$ corresponding to a utilization level of η, added to the estimated waiting time of the job at the next machine (W_{nxt}).

4. Weight biased modified RRrule (WBMR)

The WBMR rule developed in this paper has similar theoretical basis to that of the RRrule (Section 3.5) in that it linearly balances the SPRT and SPT rules based on system utilization, as the former works well in highly congested systems while the latter under low congestion. Though the RRrule performs well in minimizing the mean tardiness in simulated settings, it has two main drawbacks that make it impractical for use as a dispatching rule in practice. The first is in estimating the term W_{nxt} in an accurate and computationally feasible manner. The authors do this by examining only the next queue in the task list of the job and using what they call the “probably relative priority” of jobs in that queue to heuristically calculate W_{nxt}. This in and of itself is extremely computationally difficult as jobs in all queues must be enumerated. Moreover, the accuracy of this approach is questionable due to the dynamic nature