2012 Hawaii University International Conferences
Education, Math & Engineering Technology
July 31st to August 2nd
Waikiki Beach Marriot Resort & Spa
Honolulu, Hawaii

Online Due Date Quotation for Heavily Loaded Flow Shops

Zu-Hsu Lee
Marist College
School of Management, 3399 North Rd., Poughkeepsie, NY 12601

Wenshen Pong
School of Engineering, San Francisco State University

Abstract

The scope of this paper is online algorithm development for flow shop due date setting and job sequencing. Emphasis is placed on heavily loaded flow shops. Finite instances are randomly generated under certain probabilistic assumptions. We focus on the large-scale systems and use probabilistic analysis approach to design an efficient (polynomial) online algorithm that is effective in terms of the average-case performance.

Introduction

Given an *n*-job *m*-machine flow shop instance, each job is associated with an arrival time and a set of processing times on each machine. Each machine can handle at most one job at a time, and a job can be processed on at most one machine. Jobs arrive over time and each job is sequentially scheduled and processed on machines 1 through *m*. When the job is completed at one machine, it immediately moves to the following machine. In the online scenario, on a machine the information used for sequencing a job at any time is limited to that of the jobs that have arrived in the shop by the time. Therefore, for the scheduler the start time to process a job on a machine may be tentative until this job begins processing. This is because the new jobs that arrive when the above job is waiting at a machine may provide new information to help to improve the current tentative processing schedule.

In our problem we focus on immediate due date quotation. That is, the due date of a job has to be determined upon its arrival at the flow shop (and cannot be changed later). The objective is to determine the processing sequences on each machine so that the due dates can be all met and the average lead time is minimized. The problem can be presented as follows. Associated with job *i* (defined to be the i^{th} arrival, $i = 1,2,\ldots n$), there are a release date r_i (or the arrival time of the job at the system) and a set of processing times on all machines, $\{p_{ij},\ j = 1,2,\ldots,m\}$. The decision variables are $\{C_{ij};\ i = 1,2,\ldots n,\ j = 1,2,\ldots m\}$ as well as $\{d_i, i = 1,2,\ldots n\}$, where C_{ij} represents the completion time of job *i* on machine *j* and d_i the due date. The objective is to minimize

$\sum_{i=1}^{n}(d_i - r_i)$ with $C_{im} \le d_i$ for all i. Since no job can begin processing before it arrives at a machine, it implies $C_{i(j-1)} \le C_{ij} - p_{ij}$.

Clearly, the above objective is equivalent to minimizing $\sum_{i=1}^{n} d_i$ because all r_i are constant for a given instance. In addition, as $d_i - r_i$ is the lead time quoted to the customer, the objective and constraints presented above may indicate the ability of a company to quote lead times (or due dates) that are both relatively near in the future and attainable with certainty in a make-to-order (MTO) environment.

The scope of this paper is online algorithm development for heavily loaded flow shop systems. It can be considered as a variation of the problem presented by Van Mieghem (2003), where the single machine system is presented using queuing approach and due date considered as given parameter rather than decision variable.

Online vs. Offline

If all information of the *n*-job instance can be used for scheduling at any time (i.e. offline scenario), $d_i = C_{im}\ \forall i$ (100% reliable due dates) can be easily achieved whichever sequencing rule is applied since completion times of all jobs can be known in advance. In this case, our model is obviously a total completion time minimization problem, an NP-hard problem even for the two-machine flow shop with all jobs available at the start (Garey *et al*. 1976). Under certain probabilistic assumptions, the Shortest Processing Time among Available jobs (SPTA) and permutation scheduling rule, a polynomial sequencing procedure, has been found to be effective for this completion time problem for large-size instances (Xia *et al*. 2000, Liu *et al*. 2005). Using this rule, the available job with the shortest total processing time is selected to be processed as soon as the machine is available on the first machine, and the job sequence used on the first machine is applied to all the other machines. Note that on any machine, idle time is not allowed unless there is not any job in the queue.

First Come First Served (FCFS) can be the most naïve and simplest for our online reliable due date quotation. When FCFS is used on all machines, $d_i = C_{im}\ \forall i$ can be easily achieved as C_{ij} $\forall j$ can be known immediately at r_i based on the sequence of jobs that have arrived by r_i. However, the sequence of job arrivals may not provide a favorable value for $\sum_{i=1}^{n}(d_i - r_i)$. Thus, we are forced to seek other method(s) due to the above drawback of FCFS. The challenge is that, in general $d_i = C_{im}$ $\forall i$ is unlikely to achieve in an online environment since the time job *i* begins processing on each machine may not be known at r_i. A job that arrives after job *i* at a machine can be sequenced before job *i* if job *i* is not processed yet. Thus, when quoting a due date for a new job upon its arrival at the flow shop to allow for future better jobs to be sequenced before it on some machine(s), the estimation of the information of future arrivals is essential for our online algorithm development.

We have learned that the SPTA permutation schedule provides a favorable outcome for the offline version of our problem. Using this sequencing rule, the time job *i* starts processing on the first machine cannot be known at r_i (thus neither can C_{im}) until job *i* is the shortest (in terms of total processing time) available job when the machine is available. This indicates the challenge for online due date setting upon each job's arrival. In the subsequent section, we present the framework of our proposed algorithm. The algorithm build on the rationale behind the SPTA permutation schedule for job sequencing, incorporated with the estimation of the information of future jobs when quoting due dates.

Algorithm Design

Firstly, we specify the probabilistic nature of the system being considered. Instances consisting of *n* jobs are randomly generated based on assumptions that

A1. processing times and inter-arrival times are generated from stationary distributions respectively;

A2. processing times, p_{ij}, are drawn from identical and bounded distributions on interval $[p_{\min}, p_{\max}]$ with non-zero density $\phi(\cdot)$, defined on the interval $(0,1]$;

A3. inter-arrival times $t_i = r_{i+1} - r_i$ for all *i* are non-negative and bounded; and

A4. inter-arrival times are *i.i.d.* and independent of processing times.

Also, there is no preemption on any of the machines. We denote q_i^m the total processing time over all machines ($\sum_{j=1}^{m} p_{ij}$) for job *i*.

Note that we use the permutation scheduling rule where the processing sequence determined on the first machine will be used at the other machines on a FCFS basis. We propose the scheduling procedure on the first machine as follow.

Consider a queue of waiting jobs on machine one. The waiting jobs have each been assigned process priorities (i.e. different positions on the waiting list) and due dates upon their arrivals. The job with the first priority (i.e. position one on the list) will be processed at the time when the currently processing job is completed, and each job moves up one position in the list. When a job *j* arrives, if machine one is idle, it is processed immediately and a due date is quoted to be the completion time based on current shop information. Otherwise, job *j* is first added to the end of the queue. Then, we seek to exchange it with the nearest waiting job that has a total processing time longer than q_j^m (i.e. a longer job than *j*), without violating any due dates. This operation proceeds until job *j* is advanced to a position as close to the beginning of the queue as possible. The above sequencing procedure is the same as that presented by Kaminsky and Lee (2008) for the single machine shop (which is a special case for flow shop as $m = 1$).

Once the position of job *j* in the queue of machine 1 is finalized at time r_j, a tentative completion time may be known according to the processing status of jobs on all machines (being processed or waiting). However, to account for the future better jobs to be sequenced before *j* (if *j* is not processed yet on machine 1 when a shorter job arrives), we may want to quote d_i to be longer than the above tentative completion time (i.e. adding a slack to the above tentative completion time). Here we revise the method presented by Kaminsky and Lee (2008) to determine the slack by estimating the expected waiting time that will be added to job *j* due to the future shorter job arrivals after r_j being sequenced ahead of job *j* on machine 1. This amount of waiting time is mainly determined by the ratio of the expected processing time on machine 1 for these future shorter jobs to the expected inter-arrival time. For example, if this ratio is greater than 1, it implies job *j* may be stuck in the queue of machine 1 for a considerable amount of time.

Discussions

Our algorithm focuses on the job traffic and the processing sequence determined on the first machine to quote the due date. The arrival of future better jobs may also hold up the current jobs on the other machines; however, it appears not to be significant (see Xia *et al*. 2000, Liu *et al*. 2005). In the future research, we will build on this framework to explore the use of our algorithm when the importance of the job (i.e. job weights assigned) is considered.

References

Kaminsky, P. and Z. Lee (2008). Effective on-line algorithms for reliable due date quotation and large-scale scheduling. *Journal of Scheduling*, 11, pp. 187-204.

Liu, H., M. Queyranne and D. Simchi-Levi (2005). On the asymptotic optimality of algorithms for the flow shop problem with release dates. *Naval Research Logistics*, 52, pp. 232-242.

Van Mieghem, J. A. (2003). Due date scheduling: Asymptotic optimality of generalized longest-queue and generalized largest delay rules. *Operations Research*, 51, pp. 113-122.

Xia, C. H., J. G. Shanthikumar and P. W. Glynn (2000). On the asymptotic optimality of the SPT rule for the flow shop average completion time problem. *Operations Research*, 48, pp. 615-622.