A Distributed Learning Method for Due Date Assignment in Flexible Job Shops

Wei Weng[1], Gang Rong[2] and Shigeru Fujimura[1]

[1] Graduate School of Information, Production and Systems, Waseda University, Fukuoka 8080135, Japan

[2] State Key Laboratory of Industrial Control Technology, Institute of Cyber-systems and Control, Zhejiang University, Hangzhou, China

wengwei@toki.waseda.jp

Abstract. This study intends to help manufacturers that use flexible job shops improve performance of due date assignment, that is, setting delivery times to jobs that arrive dynamically. High performing due date assignment enables achieving on-time delivery and quick response of delivery time to customer orders. Traditional methods for due date assignment are predefined equations that estimate the duration of making a product in the production system. Such equations are sufficient for relatively simple systems such as single machine shops, but are not very high in accuracy for complex systems such as flexible job shops. To improve due date assignment for such systems, we propose a more flexible method that uses distributed learning to learn the remaining time of a job inside the system. We let each workstation in the production shop be a distributed unit that updates its local queuing time and interacts with other units to provide the total remaining time of a job. We carry out extensive computational experiments to evaluate performance of the proposed method, and the results show that it outperforms two advanced equational methods in terms of both accuracy of estimation and stability in performance.

Keywords: Due date assignment, distributed system, artificial intelligence in production

1 Introduction

In the manufacturing industry, a due date refers to the time that a product can be delivered. It is important for manufacturers to set good due dates to their products when customers give orders. The due dates should be neither too short for the production line to finish making the products nor too long for customers to wait. Good due date assignment enables achieving reliable on-time delivery, thereby improving level of customer service [1–3].

High-performing due date assignment requires a good mechanism for estimating the duration of a job inside the production system. Traditional methods for such estimation are predetermined equations that either sum up the time that is required for

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

the processing of a job and the time that is needed for waiting before the processing could start or use averaged historical durations of some recently completed jobs. Some representative methods include Constant (CON) [1, 2], Total Workload (TWK) [2, 6–8], Slack (SLK) [1], Job In Queue (JIQ), Job In Bottleneck Queue (JIBQ), and Exponential Smoothed Flowtime (ESF) [3]. Some improved methods include Dynamic TWK (DTWK), Dynamic Processing Plus Waiting (DPPW), Dynamic Feedback Processing Plus Waiting (DFPPW) [4], and Estimation Method for First-Come-First-Served (EMF) [3]. A recent review by Gordon et al. gives the details of most of the methods [5].

Such equation-based methods provide almost accurate due dates for products that are made in simple production shops such as single machine shops. For complex production shops such as flexible job shops, the methods are not very high-performing. This is because a flexible job shop contains many workstations, each having multiple parallel, alternative machines, and each job has its own predetermined route to visit some of the workstations for processing. For such a process, it is difficult to use one equation to accurately express the waiting time of a specific job. Therefore, more flexible methods are needed and this study intends to address this need.

We propose a method that is based on distributed learning for this problem. Since there is no common route for jobs to visit the workstations in a flexible job shop, information that is needed for estimating waiting times of different jobs differs. If each workstation stores some information about local waiting time, then the remaining time after a specific workstation for a specific job could be learnt by collecting information stored in the downstream workstations of the job. Therefore, we think distributed learning might be a good method for the problem and would have high flexibility for general application due to its working mechanism.

The remainder of this paper is organized as follows: Section 2 gives the detailed problem description. Section 3 describes the proposed distributed learning method. Section 4 details the computational experiments we carry out to evaluate performance of the proposed method. Finally, Section 5 concludes the paper with outlook.

2 Problem Description

We consider the problem of assigning a due date to a job that arrives dynamically in a flexible job shop system. As shown in Fig. 1, a flexible job shop is composed of multiple workstations, each having multiple machines and a buffer for waiting jobs. When the number of machines in each workstation is one, a flexible job shop becomes a job shop. The multiple machines are nonidentical, in other words, the processing time of a job differs on the machines. Jobs arrive dynamically in the shop and a job's product type is known upon the job's arrival. The product type must be one of the predefined types. A job must visit some of the workstations in a specific sequence that depends on the job's product type to become a finished product. Each workstation processes one operation of the job (denoted by O_x in Fig.1). The processing can be performed by any of the multiple machines, and which machine is selected to do it depends on the rule used for allocating jobs to machines. In this study, we use the rule First-Come-First-Served.

The objective is to set to each job upon its arrival a due date that is as close as possible to the real completion time of the job in the shop. For performance evaluation, we use relative error ratio between the assigned due date and the completion time of a job as the objective function (Eq.(1)), since relative error ratio is widely used for evaluating performance of due date assignment [3, 5].

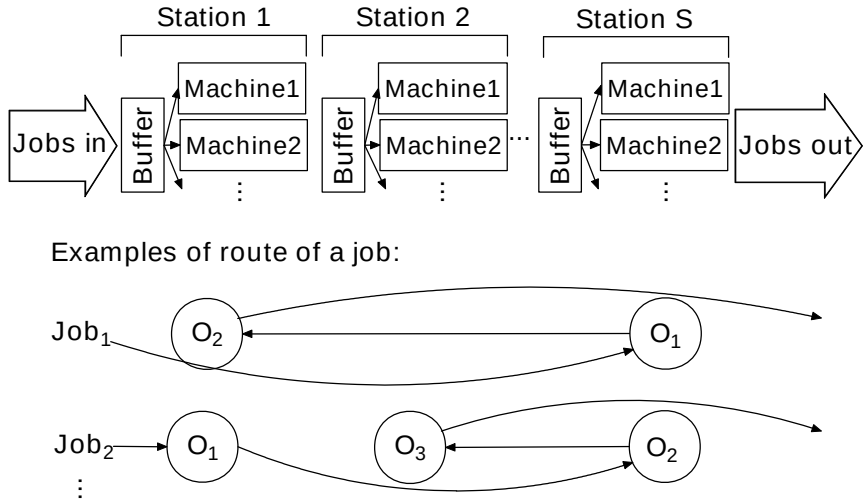

Fig. 1. The flexible job shop process.

$$Min \sum_{j=1}^{J} \frac{1}{J} \cdot \frac{|c_j - d_j|}{c_j - r_j} \cdot 100\%, \tag{1}$$

in which j is index of job, J is the number of dynamically arrived jobs, and d_j, c_j, and r_j are the due date, the completion time, and the release time (time that a job arrives in the system) of job j, respectively. The following assumptions are made in this study:

Assumptions
(1) One machine can only process one job at a time.
(2) The processing of a job is non-preemptive.
(3) The buffer in each workstation is unlimited.

3 Proposed Distributed Learning Method

We let each workstation be a distributed unit where an estimated remaining time of each product type is stored and dynamically updated. The update is based on feedback from the immediate downstream workstation of each product type at the time when

the local waiting time in the downstream workstation has changed due to arrival of new jobs. As new jobs keep arriving at the workstations, the estimated remaining time in every upstream workstation keeps being updated. When a new job arrives in the system, the estimated remaining time that is stored in the first workstation the job would visit is used for assigning a due date to the job. More details are given following the notation.

Notation

j, k	index of job, $j, k = 1, 2, ..., J$
o, i	index of operation, $o, i = 1, 2, ..., O$
$p(j)$	product type of job j, $p = 1, 2, ..., P$
$s(j, o)$	workstation that processes operation o of job j
$o(j, s)$	operation of job j that is processed in workstation s
$RT(p(j), o)$	estimated remaining time of product type $p(j)$ after operation o
$WT(s)$	average waiting time in workstation s
$Q(s)$	set of jobs waiting in workstation s
$M(s)$	set of machines in workstation s
m	index of machine in a workstation, $m = 1, 2, ..., \|M(s)\|$
$PT(j, m)$	processing time of job j on machine m
$RT(m)$	remaining processing time of the job being processed on machine m
$OT(j, o)$	average processing time of operation o of job j, $= \sum_{m \in M(s(j,o))} PT(j, m)/\|M(s(j, o))\|$

Initially, $RT(p(j), o)$ is set to be the average total processing time of the remaining operations after operation o in job j, i.e. the case in which there is no waiting in any workstations:

$$RT(p(j), o) = \sum_{i=o+1}^{O} OT(j, i). \tag{2}$$

As new jobs arrive dynamically in the system, each job is released into the system immediately after it arrives. At the time, due date of job j is assigned to be

$$d_j = OT(j, 1) + WT(s(j, 1)) + RT(p(j), 1). \tag{3}$$

In Eq.(3), $OT(j, 1)$ is known whereas $WT(s(j, 1))$ and $RT(p(j), 1)$ are collected from workstation $s(j, 1)$. The process to update $WT(s)$ and $RT(p(j), o)$ is as follows:

Every time a job, say job j, arrives in workstation $s(j, o)$, it joins queue $Q(s(j, o))$ first, and then $WT(s(j, o))$ is updated to be

$$WT(s(j, o)) = \frac{\sum_{k \in Q(s(j,o))} OT(k, o(k, s(j, o))) + \sum_{m \in M(s(j,o))} RT(m)}{|M(s(j, o))|}. \tag{4}$$

The upper part is the total average processing time in the workstation of waiting jobs plus the total remaining processing time on the machines. Dividing it by the number of machines in the workstation leads to the average waiting time in the workstation. The change in $WT(s(j, o))$ due to the arrival of job j would affect remaining times of

the upstream operations of jobs that go through $s(j,o)$, so estimated remaining times in the workstations where the immediate upstream operation of a job k whose product type requires the job to visit $s(j,o)$ are updated to be

$$RT(p(k),o(k,s(j,o))-1) = (1-\alpha)\cdot RT(p(k),o(k,s(j,o))-1)+ \\ \alpha\cdot[WT(s(j,o))+RT(p(k),o(k,s(j,o)))], \tag{5}$$

in which α is a parameter in the range [0,1].

This is like a learning process whose learning rate is α. When α is at the higher end of range, change in $WT(s(j,o))$ leads to fast change in the remaining times of the immediate upstream workstations. In contrast, a small α results in slower change in them.

4 Computational Experiments

We evaluate the proposed method by generating random problem instances using factors given in Table 1. For each problem instance, 1200 jobs arrive dynamically in the system, the interarrival time between them being exponential distribution whose mean is given by

$$I = \frac{\overline{PT}}{M\cdot U}, \tag{6}$$

in which $\overline{PT}$ is the average total processing time of a job, M is the total number of machines in the system, and U is machine utilization rate. We use the first 200 jobs to warm up the system to a relatively stable status and record due dates and completion times of the remaining 1000 jobs for obtaining the reported results. We set the parameter α by carrying out pilot tests for each problem instance.

Table 1. Factors for generating random problem instances

Factor	Levels	Number of levels
S	5, 10	2
P	5, 10	2
M	5, 10, 20	3
U	0.8, 0.85, 0.9, 0.95	4
	Number of combinations	48
	Problems generated for each combination	5
	Total number of problem instances	240

Tables 2-6 show the simulation results. In the tables, the proposed method is denoted by DL. We compare with DFPPW [4] and EMF [3]. DFPPW is an improved version of DPPW and DPPW is an improved version of PPW. DFPPW is reported to

Table 2. Simulation results

DFPPW	DL	EMF
25.00[a]	10.40	23.42
12.70[b]	7.18	21.48

[a] Objective function value: relative error ratio
[b] Standard deviation

Table 3. Simulation results under different levels of S

	S=5	S=10
DFPPW	25.44	24.33
	13.43	11.48
DL	8.81	12.77
	6.40	7.63
EMF	25.53	20.25
	23.95	16.64

outperform DTWK and DPPW, which outperform TWK and PPW respectively. EMF is reported to outperform many traditional methods such as TWK, JIQ, JIBQ, and ESF.

Since the objective aims for minimization, values in the tables are the smaller the better. Table 2 gives the overall results, in which DL shows a relative error ratio that is less than the other two methods by over 50% and a standard deviation that is less than the others by nearly 50%. This indicates significant improvement in both accuracy of due date forecast and stability in performance.

To check impact from the levels of factors, we further divide the results according to levels of S, P, M, and U, respectively, and show the results in Tables 3-6.

From Table 3, it could be found that DL works better for systems in which the number of workstations is relatively small. This is probably due to the mechanism of DL that only when a job arrives in a workstation will the estimated remaining times of the immediate upstream workstations be updated. When there are many workstations in the system, the update of remaining times from the most downstream workstation to the most upstream workstation of a job might take time, causing a delay in using

Table 4. Simulation results under different levels of P

	P=5	P=10
DFPPW	24.79	23.38
	13.19	9.35
DL	13.76	14.34
	5.18	6.46
EMF	31.96	36.65
	16.28	19.83

Table 5. Simulation results under different levels of M

	M=5	M=10	M=20
DFPPW	15.79	23.44	31.16
	6.78	9.48	14.40
DL	10.00	11.27	9.72
	6.67	7.23	7.29
EMF	18.09	18.68	30.81
	13.35	16.38	26.55

Table 6. Simulation results under different levels of U

	U=0.8	U=0.85	U=0.9	U=0.95
DFPPW	21.99	24.08	25.91	28.00
	11.59	12.40	12.68	13.28
DL	9.86	10.29	10.78	10.66
	6.74	7.07	7.40	7.45
EMF	23.76	23.49	23.12	23.29
	22.62	21.02	20.77	21.46

the newest waiting time information when estimating the durations of new jobs arrive in the system.

From Table 4, it could be found that as the number of product types increases, performance of DL slightly decreases. This might be understood as the result of using the average processing time of a large number of products, which is rougher than the average processing time of a small number of products.

Tables 5-6 show that neither M nor U have a noticeable impact on performance of DL. In summary, DL shows better performance than the other methods in all examined cases, indicating it to be a general better choice than traditional methods for due date assignment in flexible job shops.

5 Conclusions and outlook

For the problem of due date assignment in flexible job shops, we propose a distributed learning method that updates the remaining time of a job by using changed local waiting time in immediate downstream workstation. The method differs from traditional equation-based methods and has the advantage of having no restriction on applicable shop types.

By carrying out extensive computational experiments, we evaluated performance of the proposed method and found it to significantly outperform two comparatively high-performing methods in terms of both performance and stability. Thus, the proposed method would be expected to improve a manufacturer's performance in both due date promising and reliability of on-time delivery.

As stated in the discussion of results, the proposed method might have the problem of having a delay in updating the remaining times of upstream workstations if

there are many workstations in the system, so future work might include fixing this problem, thereby enabling the method to achieve higher performance in general system environments.

6 Acknowledgement

The authors acknowledge the support of JSPS Grants-in-Aid for Scientific Research (KAKENHI) Grant Number 15K16296 and the National High Technology R&D Program of China (2014AA041805).

References

1. Sridharan, V., Li, X.: Improving delivery reliability by a new due-date setting rule. European Journal of Operational Research 186: 1201-1211 (2008)
2. Moses, S., Grant, H., Gruenwald, L., Pulat, P.: Real-time due-date promising by build-to-order environments. International Journal of Production Research 42: 4353-4375 (2004)
3. Lee, G. -C.: Estimating order lead times in hybrid flowshops with different scheduling rules. Computers and Industrial Engineering 56: 1668-1674 (2009)
4. Weng, M. X., Wu, Z., Qi, G., Zheng, L.: Multi-agent-based workload control for make-to-order manufacturing. International Journal of Production Research 46: 2197-2213 (2008)
5. Gordon, V., Strusevich, V., Dolgui, A.: Scheduling with due date assignment under special conditions on job processing. Journal of Scheduling 15: 447-456. DOI: 10.1007/s10951-011-0240-2 (2012)
6. Sha, D. T., Hsu, S. Y.: Due-date assignment in wafer fabrication using artificial neural networks. International Journal of Advanced Manufacturing Technology 23: 768-775 (2004)
7. Vinod, V., Sridharan, R.: Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. International Journal of Production Economics 129: 127-146 (2011)
8. Baykasoglu, A., Gocken, M.: Gene expression programming based due date assignment in a simulated job shop. Expert Systems with Applications 36: 12143-12150 (2009)