Computers & Operations Research 39 (2012) 3316–3324

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

Dynamic routing strategies for JIT production in hybrid flow shops

Wei Weng *, Xin Wei, Shigeru Fujimura

Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0135, Japan

ARTICLE INFO

Available online 27 April 2012

Keywords:
Dynamic routing
Hybrid flow shop
Dispatching rules
Just-in-time production

ABSTRACT

In production processes, just-in-time (JIT) completion of jobs helps reduce both the inventory and late delivery of finished products. Previous research which aims to achieve JIT job completion mainly worked on static scheduling problems, in which all jobs are available from time zero or the available time of each job is known beforehand. In contrast, dynamic scheduling problems which involve continual arrival of new jobs are not much researched and dispatching rules remain the most frequently used method for such problems. However, dispatching rules are not high-performing for the JIT objective. This study proposes several routing strategies which can help dispatching rules realize JIT completion for jobs arriving dynamically in hybrid flow shops. The routing strategies are based on distributed computing which makes realtime forecast of completion times of unfinished jobs. The advantages include short computing time, quick response and robustness against disturbance. Computer simulations show that the performance of dispatching rules combined with the proposed routing strategies is significantly higher than that of dispatching rules only and that of dispatching rules combined with the previous routing methods.

1. Introduction

In the manufacturing industry, production processes are usually conducted in shops which are linked as a production chain. For each shop along the chain, temporary inventory of finished products will be produced if jobs are completed earlier than the time when they can be delivered to the successor shop, and late delivery of finished products will be caused if jobs are completed later than the time mentioned above. This indicates that only when each job is completed on time can the inventory and late delivery of finished products be minimized simultaneously. The same applies to the flow of products between the parties along a supply chain, as JIT job completion can lead to zero inventory of finished products for the manufacturer as well as timely delivery of products to the downstream customer.

To achieve JIT completion for a job, a number of factors should be taken into account, including the due date (i.e. the time that a job should be finished), earliness and tardiness. Perhaps due to this reason, most existing research for the JIT objective work on problems with relatively simple conditions such as the single machine common due date problem [13] and identical parallel machine problem [20,1,19]. Only a small number of studies have tried to solve more complex problems such as the hybrid flow shop (HFS) problem [6,10].

The HFS problem is one of the typically complex problems which are widely seen in industries [16]. It is the same as the flow shop problem except that there are multiple machines available at each stage. As a job must go through all the stages in sequence and be processed by one of the machines at each stage, there exist a lot of different routes for a job to go through an HFS. Because of this, the HFS problem is proven to be strongly NP-hard and even solving a small-sized problem may require considerable time [17].

Therefore, researchers have been trying to develop efficient models which can find the optimal or near-optimal solutions for the HFS problem within reasonable time. Commonly used methodology includes exact methods such as branch and bound [12] and mixed integer programming [4,10] as well as heuristic methods such as genetic algorithm and simulated annealing [4,5,9].

These efforts have surely upgraded the searching efficiency and helped reduce the difficulty in dealing with large-scale HFS problems. However, there are still limitations in applying the methods to daily production, in that these methods are developed based on the assumption that all jobs are available from time zero or the available time of each job is known beforehand. In other words, these are static scheduling methods which plan a given number of received jobs. For jobs which are unknown beforehand or arrive dynamically at the HFS production shop, rescheduling must be made, which may require long computing time and, in the case that previously received jobs have already started undergoing processing, may reduce the quality of solutions.

According to these limitations of static scheduling and the complex nature of HFS, this paper presents some routing strategies

* Corresponding author.
E-mail address: wengwei@toki.waseda.jp (W. Weng).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.04.022

which use dynamic scheduling to solve the HFS problem for JIT objective. The reason to choose dynamic scheduling lies in that dynamic scheduling has the advantage of being much less restricted by the size and complexity of a problem than static scheduling. Instead of seeking for the optimal solution for a given number of received jobs, dynamic scheduling focuses more on the provision of quick solutions to jobs which are received continually over time.

Existing research on dynamic HFS scheduling is still small in number and dispatching rules and their combinations remain the most widely used method [7,18,2]. Since most dispatching rules are designed for the objective of minimizing the makespan, mean flow time or tardiness [14,3], and almost no dispatching rule considers both the due date of a job and the speed of a machine, it is difficult to use these rules to deliver good JIT performance.

To make up for this void, this study proposes routing strategies which can help dispatching rules realize the function of JIT job completion in HFS's. The strategies work based on realtime distributed computing, which requires only small amount of calculation to make a routing decision. Such autonomous control based methods are believed to be suitable for dynamic HFS problems [18]. By using the strategies, a job can be delivered to the most suitable machine at each stage for processing and finally be completed at a time close to its due date.

As the proposed strategies preserve all the advantages of dynamic scheduling, the available time of a job is not required to be known before the job becomes available, and no rescheduling is necessary after new jobs are received. No matter how large the HFS environment is, no matter how many jobs are involved, and whether there exists dynamic arrival of new jobs or not, the proposed strategies is able to work unaffected.

This paper is organized as follows: Section 2 gives the problem statement; Section 3 presents the proposed routing strategies; Section 4 gives the detailed computer simulations and discussions; and Section 5 concludes the paper with some suggestions on future topics.

2. Problem statement

Notation

C_j	completion time of job j
D_j	due date of job j
E_j	earliness of job j
J	number of jobs received dynamically
M_s	set of stage-s machines
P	number of predefined product types
S	number of stages
T_j	tardiness of job j

We consider an HFS production system (Fig. 1) which is consisted of S stages ($s=1, 2, \ldots, S$), with $|M_s|$ machines ($m=1, 2, \ldots, |M_s|$) which are different in processing speed at stage s. Each machine has a buffer where jobs to be processed by it wait in a queue.

Jobs arrive dynamically over time and each job must be made into one of the predefined P types of products ($p=1, 2, \ldots, P$). A job must go through the S stages in sequence for processing so as to become a final product. At each stage, a job can be processed by any one of the machines, and the processing time differs depending on the machine. For jobs of the same product type, their processing times are identical on the same machine. Waiting jobs at each machine are processed in the order of a dispatching rule, and one dispatching rule is used throughout the system. No delivery time exists between the stages and a job is considered to arrive at the next stage as soon as it is completed by the predecessor stage.

Each job has a due date D_j, which is the time when the job should be completed as the product. D_j must be longer than the total processing time of job j excluding the waiting time, and is given at the time when job j arrives in the HFS system. If job j is completed before D_j, its earliness is expressed by $E_j=\max(0,D_j-C_j)$, in which C_j represents the completion time of job j; otherwise, its tardiness is expressed by $T_j=\max(C_j-D_j,0)$. The maximum function is used in E_j and T_j, accordingly an early job produces zero tardiness; and a tardy job, zero earliness.

The objective of this study is to minimize the total earliness and tardiness of each job. For evaluation, the average earliness and tardiness of all the jobs that are received dynamically is used

$$\text{Minimize} \sum_{j=1}^{J}(E_j+T_j)\cdot\frac{1}{J}. \tag{1}$$

Besides, some assumptions are made in this study.

Assumptions:

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

3. Proposed routing strategies

Notation

$DM_{j,s}$	destination machine of job j at stage s
$FC_{j,s,m}$	forecast completion time of job j if machine m processes the job at stage s
$L_{m,T}$	time before machine m finishes the job in which it is processing at time T
$PT_{j,s,m}$	processing time of job j on machine m of stage s

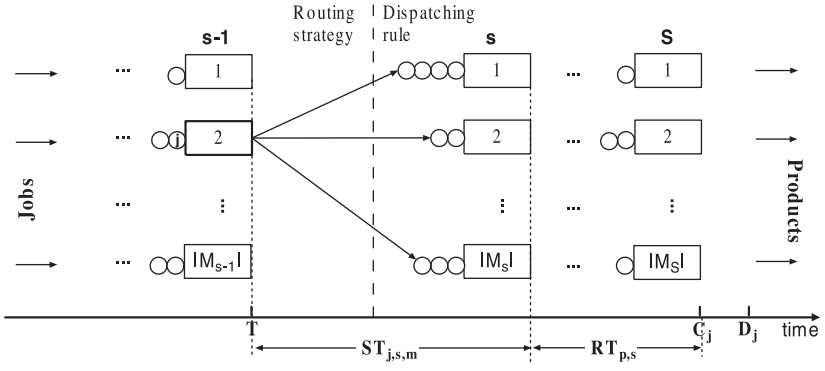

Fig. 1. The production environment of HFS and illustration of notation of time.

$PT_{p,s}$	average processing time of jobs of product type p at stage s, obtained by $PT_{p,s}=\sum_{m=1}^{\lvert M_s\rvert} PT_{j,s,m}/\lvert M_s\rvert$ in which p represents the product type of job j
$Q_{j,m,d}$	number of jobs which will be processed earlier than j by machine m according to dispatching rule d
$QT_{j,s}$	average queue time of job j at stage s if the waiting jobs are processed in the FIFO order
$RT_{p,s}$	approximate remaining time of jobs of product type p after stage s
$ST_{j,s,m}$	time that job j will spend at stage s if it is processed by machine m
T	time at present
$WT_{j,s,m}$	waiting time of job j at machine m of stage s

The routing strategies proposed in this study determine the processing machine for each job. More exactly, after a job arrives at stage s, it is first routed to one of the $|M_s|$ machines, and then wait to be processed by the machine in the order of a dispatching rule. The major function of the proposed strategies is to judge whether it is fit to allocate a certain job to a certain machine according to the job's due date and the machine's processing time.

3.1. Basic routing strategy (BRS)

A basic routing strategy (BRS) is proposed, which forms the basis for the routing strategies developed in this study. The BRS is implemented at each stage, where it works in the following way.

When job j arrives at stage s, a forecast completion time, $FC_{j,s,m}$, will be calculated for it according to each machine of the stage

$$FC_{j,s,m}=\underbrace{T+ST_{j,s,m}}_{\text{known}}+\underbrace{RT_{p,s}}_{\text{to be learnt}},\quad m=1,2,\ldots,|M_s|. \tag{2}$$

In the calculation, $ST_{j,s,m}$, the time that job j will spend at stage s if it is processed by machine m, is obtained by

$$ST_{j,s,m}=WT_{j,s,m}+PT_{j,s,m}. \tag{3}$$

The waiting time $WT_{j,s,m}$ can be expressed as

$$WT_{j,s,m}=L_{m,T}+\sum_{i=1}^{Q_{j,m,d}} PT_{i,s,m} \tag{4}$$

in which $L_{m,T}$ represents the time before machine m finishes its currently processing job and $Q_{j,m,d}$ denotes the number of jobs which will be processed earlier than j by machine m according to dispatching rule d. Fig. 1 gives an illustration of T, $ST_{j,s,m}$ and $RT_{p,s}$, which are shown based on the position of job j inside the HFS system.

According to BRS, the machine whose $FC_{j,s,m}$ is the closest to D_j among the $|M_s|$ machines will be selected to process job j

$$DM_{j,s}=\arg\min_{m\in M_s}|FC_{j,s,m}-D_j|. \tag{5}$$

After job j is routed to $DM_{j,s}$, it will wait to be processed by the machine in the order of a dispatching rule.

After job j is routed, the approximate remaining time after the predecessor stage $s-1$ will be updated for each product type. The updating procedure is like a simplified Q-learning process and can be expressed by

$$RT_{p,s-1}\leftarrow\alpha\cdot RT_{p,s-1}+(1-\alpha)\cdot(QT_{j,s}+PT_{p,s}+RT_{p,s}). \tag{6}$$

In the equation

$$QT_{j,s}=\sum_{m=1}^{|M_s|}\left(L_{m,T}+\sum_{i=1}^{Q_{j,m,FIFO}} PT_{i,s,m}\right)\Big/|M_s|. \tag{7}$$

BRS represents the core idea which enables JIT job completion to be achieved in dynamic HFS problems. Taking into account that changes in the arrival rate of new jobs may affect the routing performance, some supportive strategies are proposed to enhance the performance under disturbances and conditions.

3.2. Feedforward of demand change (FDC)

As jobs are received dynamically from customer orders, the speed at which new jobs arrive is not constant. More jobs might be received in a period when customer demand is strong than in another period when the demand is weak. To enhance the routing performance under such changes in demand, a feedforward strategy is proposed to adjust the remaining times of products accordingly.

Since the first stage is the place where changes in demand are directly and firstly received, it is designed that remaining times at successor stages are dynamically adjusted according to the changes in queue time at the first stage. The queue time represent the demand level, because increase in queue time implies that new jobs arrive at a speed which is higher than the processing speed of the first stage, whereas decrease in queue time indicates that new jobs arrive at a speed lower than that.

It is designed that when a new job j arrives at the first stage, the stage queue time is checked and compared with that at the time when the last job j^* arrived

$$\Delta QT_{j,1}=QT_{j,1}-QT_{j^*,1}. \tag{8}$$

If $\Delta QT_{j,1}$ is positive, the remaining times at successor stages will be adjusted longer because a similar increase in queue time may occur some time later at some successor stages. Similarly, if $\Delta QT_{j,1}$ is negative, the remaining times at successor stages will be adjusted shorter. The adjustment is made by

$$RT_{p,s}\leftarrow RT_{p,s}+\Delta QT_{j,1},\quad s=1,2,\ldots,b,\ p=1,2,\ldots,P. \tag{9}$$

In the equation, b is the index of bottleneck stage, i.e. the stage whose average processing time for all product types $\sum_{p=1}^{P} PT_{p,s}/P$ is the longest among all the stages. Such a stage, if not being the first stage, is likely to suffer an increase or decrease in queue time similar to that at the first stage some time later than the first stage.

By making such adjustment, the shortcoming of a feedback-driven system which may take a longer time to respond to changes is overcome.

3.3. Inclination to early machines (IEM)

Since the completion time of a job is determined at the final stage, it will be helpful if one of the machines there is suitable in processing time for each job. When a job arrives at the final stage, there are totally three possible cases: (1) some machines will finish the job early and others late, (2) all the machines will finish the job early, and (3) all the machines will finish the job late. The first case is the most desirable one, in which the job will be routed to the most JIT machine. In the second case, the job will be routed to the machine where it will be completed with the smallest earliness. But after being routed, the job's earliness may further be reduced if any subsequently arrived job gets processed earlier than the job under the dispatching rule. In the third case, however, after the job is routed to the machine where it will be completed with the smallest tardiness, there are no chances to reduce the job's tardiness any more. Accordingly, if a job encounters the third case, even if the smallest tardiness is big, its completion time cannot be adjusted earlier.

In order to prevent serious tardiness which may be produced in the third case, it is proposed that a higher priority is given to routing a job to an early machine than to a tardy machine at an

early stage. Such inclination to early machines is realized by the following decision making process.

When job j arrives at stage s, an early-machine candidate $eM_{j,s}$ and a tardy-machine candidate $tM_{j,s}$ are identified by

$$eM_{j,s} = \arg\min_{m \in M_s} |FC_{j,s,m} - D_j|, \quad FC_{j,s,m} < D_j, \tag{10}$$

$$tM_{j,s} = \arg\min_{m \in M_s} |FC_{j,s,m} - D_j|, \quad FC_{j,s,m} \geq D_j. \tag{11}$$

Job j will be routed to one of the candidates by using the following:

$$DM_{j,s} = \begin{cases} eM_{j,s}, & \dfrac{D_j - FC_{j,s,eM_{j,s}}}{FC_{j,s,tM_{j,s}} - D_j} \leq 1 + \dfrac{S-s}{S}, \\ tM_{j,s} & \text{otherwise.} \end{cases} \tag{12}$$

In the equation, the part $(S-s)/S$ adjusts the probability of selecting the early machine according to the position of job j. As job j moves toward the final stage, the probability of selecting the early machine becomes smaller. At the final stage, $(S-s)/S$ becomes zero, and as a result, job j will be routed to the most JIT machine for completion.

To apply this strategy, replace Eq. (5) in the basic strategy with Eq. (12).

3.4. Consideration of time flexibility (CTF)

The realtime updated $RT_{p,s}$ has an important role to play in the routing of jobs, but inevitably there may exist some difference between the actual remaining time and $RT_{p,s}$. If the difference is big, the proposed IEM strategy may not be enough to make up for such a gap. In such a case, allowance of time flexibility will be helpful.

In light of this, a strategy is proposed that the due date of job j, which is compared with $FC_{j,s,m}$ in the routing process, is tightened to be $D_{j,s}$ at early stages so that some time flexibility can be reserved for late stages

$$D_{j,s} = D_j - \beta \cdot (S-s). \tag{13}$$

The parameter β is used to determine the allowed time flexibility for each stage.

To apply this strategy, replace Eq. (5) in the basic strategy with the following equation:

$$DM_{j,s} = \arg\min_{m \in M_s} |FC_{j,s,m} - D_{j,s}|. \tag{14}$$

To apply the strategies IEM and CTF simultaneously, replace D_j in Eq. (12) with $D_{j,s}$ and then replace Eq. (5) with Eq. (12).

4. Computer simulations and analysis

The proposed routing strategies are simulated upon HFS problem instances which are generated according to the factors given in Table 1. In the table, $|M_s|$ and $PT_{j,s,m}$ are both generated by using uniform distribution. D_p represents the due date of jobs of product type p. In the case that D_p is fixed, the due date of a job of product type p will be a fixed time after the time when the job arrives. The fixed time is obtained by $FT_p = \gamma_p \cdot \sum_{s=1}^{S} PT_{p,s}$, in which γ_p is predetermined randomly for each product type within the range [1,2]. In the case that D_p is variable, the due date of a product-p job will be $\sigma_j \cdot FT_p$ after the time when the job arrives, in which σ_j is determined randomly at the job's arrival within the range [0.8,1.8] and FT_p is the fixed time predefined by using the method described above. λ, which represents the arrival rate of new jobs, is the number of new jobs which arrive per unit time according to the Poisson distribution. λ_l and λ_h are both obtained through pilot tests for each problem instance. λ_l represents the arrival rate at which about 3/4 jobs will be completed early and the other 1/4 late if jobs are dispatched to machines by using the dispatching rule FIFO. λ_h represents the arrival rate at which about 1/4 jobs will be completed early and the other 3/4 late under the described dispatching method. Twenty problem instances are generated randomly for each combination of factors, and totally 720 problem instances are generated.

Table 1
Design for generating problem instances.

Factor	Levels	Number of levels
S	10, 20, 40	3
P	5, 15, 30	3
$\lvert M_s \rvert$	[2,10]	1
$PT_{j,s,m}$	[2,60]	1
D_p	Fixed, variable	2
λ	λ_l, λ_h	2
	Number of combinations	36
	Problems generated for each combination	20
	Total number of problem instances	720

Table 2
Equations to select the next processing job by the dispatching rules.

Rule	Next job
MST	$\arg\min_{j \in WJ_s}(D_j - T - \sum_{k=s}^{S} PT_{p,k})$
SRPT	$\arg\min_{j \in WJ_s}[(D_j - T - \sum_{k=s}^{S} PT_{p,k}) / \sum_{k=s}^{S} PT_{p,k}]$
EDD	$\arg\min_{j \in WJ_s} D_j$
MDD	$\arg\min_{j \in WJ_s} \max(D_j, T + \sum_{k=s}^{S} PT_{p,k})$
CR	$\arg\min_{j \in WJ_s}[(D_j - T) / \sum_{k=s}^{S} PT_{p,k}]$
CRSPT	$\arg\min_{j \in WJ_s} \max[PT_{p,s}, (D_j - T) \cdot PT_{p,s} / \sum_{k=s}^{S} PT_{p,k}]$

To evaluate the proposed routing strategies, some typical dispatching rules are selected for comparison. Besides the basic rule FIFO, two time-related rules: Minimum-Slack-Time (MST) and Slack/Remaining-Processing-Time (SRPT), two duedate-related rules: Earliest-Due-Date (EDD) and Modified-Due-Date (MDD), and two time- and duedate-related rules: Critical-Ratio (CR) and CR+Shortest-Processing-Time (CRSPT) are selected. These rules are often used for comparison in dynamic HFS problems [14,3], and some are reported to be competitive for the JIT objective [11]. In the case that no routing strategy is adopted, the dispatching rules collect all the waiting jobs at stage s in one buffer and use the equations in Table 2 to select the job which will be delivered to the machine that becomes idle first among the $|M_s|$ machines. If more than one machine is idle, the selected job will be delivered to the machine whose processing time is the shortest. In Table 2, WJ_s represents the waiting jobs at stage s, and p is the product type of job j. In the case that a routing strategy is adopted, the dispatching rules only determine the sequence of jobs which wait in the buffer of a machine.

In addition, two realtime feedback based routing methods: Multi-Agent-Learning (MAL) and Distributed-Arrival-Time-Control (DATC) proposed in the previous research [2,15,8] are also introduced for comparison. Both the methods were designed based on the assumption that jobs were processed in the FIFO order, so in the simulations FIFO is applied for these two methods. The working mechanisms of these two methods are given in the Appendix.

In the simulations, $RT_{p,s}$ is initialized to be $RT_{p,s} = \sum_{k=s+1}^{S} PT_{p,k}$ and the parameters α and β are set to be 0.2 and $5(\lambda_l/2 + \lambda_h/2)/\lambda$, respectively, since pilot tests show that such setting produces relatively high performance.

The program is written in JAVA and run on a PC platform of Pentium IV 3.00 GHz with 1 GB memory. For each problem instance, the program is executed for 100 times, and for each time 500 jobs dynamically arrive at the HFS system. The results reported are the averages of the 720 instances, each of which includes totally $100 \times 500 = 50{,}000$ jobs.

The simulations include two parts: in the first part, new jobs arrive at relatively stable speed, i.e. λ does not change during the simulation; in the second part, the speed at which new jobs arrive changes between λ_l and λ_h once during the simulation. The purpose is to reveal the system performance under relatively stable demand as well as to check the system robustness under demand changes.

4.1. Simulation under stable demand

Table 3 gives the comparison of the overall performance under stable demand. In addition to the objective function value which is denoted by ET, the relative error ratio (RER) between ET and the lead time of a job is also traced. To show the improvement which is obtained by each of the proposed strategies and to compare with MAL and DATC, ET and RER are measured for each strategy respectively when jobs are processed in the FIFO order. The comparison in the aim of finding the best performing combination of a dispatching rule and the proposed routing strategies is conducted by combining each of the dispatching rules with all the proposed strategies (denoted by ALL).

From Table 3, it can be seen that compared with the traditional dispatching rules, the performance of dispatching rules combined with routing strategies is noticeably better. By using previous routing strategies, ET's of FIFO+MAL and FIFO+DATC are 26% and 49% lower than that of the best-performing dispatching rule CR, respectively. Compared with the best-performing previous strategy DATC, the proposed BRS shows a reduction of 21% in ET. Combined with the supportive strategies FDC, IEM and CTF, ET of BRS is further reduced by 13%, 27% and 42%, respectively. These demonstrate that the distributed feedback design of BRS is more effective and efficient than those of MAL and DATC. Effectiveness of the supportive strategies indicates that the disturbances and conditions, for which the strategies are specifically designed, are included in most general cases.

ET of FIFO+BRS+ALL is only a little lower than the smallest ET of FIFO+BRS+FDC, FIFO+BRS+IEM and FIFO+BRS+CTF, which indicates that the application of one or two of the supportive strategies may be enough in some cases.

The best- and second best-performing combinations are SRPT+BRS+ALL and FIFO+BRS+ALL, whose ET's are 57% and 54% lower than that of FIFO+DATC, respectively.

Standard deviations after using all the proposed strategies become smaller than the others, which prove that the strategies can enhance the stability in achieving good performance.

Table 4 provides the result of analysis of variance (ANOVA) for the performance measure ET. ANOVA is widely used to check if a factor (or combination of several factors) has a significant impact on the performance measure. The closer to 0 the p-value is, the larger the significance will be; and the closer to 1 it is, the less the significance will be. Generally, if the p-value is less than 0.05, it is believed that the corresponding factor has a significant impact on the performance measure.

We conducted ANOVA among all the methods including dispatching rules and combinations of them with routing strategies (Rule+Routing), among dispatching rules only (Rule), and among combinations of dispatching rules with routing strategies only (Routing). Such an analysis can reflect the impact of a factor under different methods. For example, the p-values of D_p are less than 0.05 under both Rule and Routing, but more than 0.05 under Rule+Routing. This implies that D_p has significant impacts on the performance of Rule and the performance of Routing, respectively. However, its impact on the performance of Rule might be opposite to its impact on the performance of Routing, because its p-value becomes insignificant in the analysis of the combined data of Rule and Routing. Such discussions will be made in detail based on the results given by Tables 5–8, which provide more detailed performance comparison according to S, P, D_p and λ.

Table 5 shows that no matter which method is used, ET increases, whereas RER decreases as the number of stages increases. Compared with each other, ET's of dispatching rules combined with the proposed strategies are about 1/3 of those of dispatching rules only and 1/2 of the dispatching rules combined with the previous routing methods, regardless of the number of stages. In HFS systems of no more than 10 stages and no less than 40 stages, the proposed routing strategies can generally control RER to be within 11% and 4%, respectively.

In Table 6, it is noticed that when $P=15$, ET's and RER's of the proposed strategies are the smallest among the three P levels. Though the ANOVA result in Table 4 shows that the difference is not significant, it is possible that there exists a certain range of P for which the proposed strategies are highest in performance. In other words, good JIT performance might be expected if the

Table 3
Comparison of overall performance under stable demand.

Rule+Routing	ET	RER	σ_{ET}	σ_{RER}
FIFO	251	0.30	18.63	0.03
MST	199	0.25	28.93	0.04
SRPT	204	0.25	27.90	0.04
EDD	200	0.25	29.07	0.04
MDD	206	0.26	27.93	0.04
CR	198	0.25	29.07	0.04
CRSPT	247	0.30	22.15	0.03
FIFO+MAL	147	0.21	30.53	0.05
FIFO+DATC	100	0.15	26.77	0.05
FIFO+BRS	79	0.12	25.39	0.04
FIFO+BRS+FDC	69	0.11	24.22	0.04
FIFO+BRS+IEM	58	0.09	21.02	0.04
FIFO+BRS+CTF	46	0.08	18.47	0.03
FIFO+BRS+ALL	46	0.08	17.17	0.03
MST+BRS+ALL	50	0.08	21.49	0.04
SRPT+BRS+ALL	**43**	**0.07**	**18.21**	**0.03**
EDD+BRS+ALL	49	0.08	21.16	0.04
MDD+BRS+ALL	54	0.09	21.72	0.04
CR+BRS+ALL	51	0.09	21.62	0.04
CRSPT+BRS+ALL	59	0.09	17.89	0.03

ET: objective function value.
RER: relative error ratio between ET and the lead time of a job, obtained by $[\sum_{j=1}^{J}(E_j+T_j)/(D_j-AT_j)]/J$ in which AT_j represents the time when job j arrives in the HFS system.
σ_{ET}: standard deviation of ET.
σ_{RER}: standard deviation of RER.

Table 4
The p-value of analysis of variance (ANOVA) for the performance measure ET.

Source	Rule+Routing	Rule	Routing
Method (M)	0.0000*	0.3194	0.0000*
S	0.0006*	0.0000*	0.0008*
P	0.9381	0.1105	0.1358
D_p	0.3651	0.0000*	0.0005*
λ	0.5226	0.1301	0.0000*
$M \times S$	0.0000*	0.0000*	0.0000*
$M \times P$	0.0075*	0.9972	0.1686
$M \times D_p$	0.0000*	0.7339	0.9955
$M \times \lambda$	0.0010*	0.8221	0.2535

* Marks statistical significance at the 0.05 level.

Table 5
Performance comparison under stable demand according to the number of stages.

Rule+Routing	ET	RER	σ_{ET}	σ_{RER}
$S=10$				
FIFO	120	0.33	17.50	0.05
MST	103	0.29	23.85	0.07
SRPT	104	0.29	23.37	0.06
EDD	103	0.29	23.87	0.07
MDD	107	0.30	23.42	0.06
CR	102	0.28	23.98	0.07
CRSPT	122	0.34	17.59	0.05
FIFO+MAL	82	0.25	26.38	0.08
FIFO+DATC	63	0.19	22.46	0.07
FIFO+BRS	49	0.15	21.38	0.07
FIFO+BRS+FDC	45	0.14	20.24	0.06
FIFO+BRS+IEM	37	0.12	17.62	0.06
FIFO+BRS+CTF	33	0.11	16.77	0.05
FIFO+BRS+ALL	**31**	**0.11**	**15.03**	**0.05**
MST+BRS+ALL	42	0.13	20.30	0.06
SRPT+BRS+ALL	32	0.10	16.71	0.05
EDD+BRS+ALL	42	0.13	20.17	0.06
MDD+BRS+ALL	47	0.14	21.33	0.06
CR+BRS+ALL	43	0.13	20.23	0.06
CRSPT+BRS+ALL	45	0.13	16.54	0.05
$S=20$				
FIFO	210	0.28	16.63	0.02
MST	164	0.23	28.36	0.04
SRPT	167	0.23	27.10	0.03
EDD	164	0.23	28.34	0.04
MDD	168	0.23	26.95	0.03
CR	163	0.23	28.18	0.04
CRSPT	211	0.29	19.32	0.02
FIFO+MAL	137	0.21	32.22	0.05
FIFO+DATC	96	0.15	29.27	0.05
FIFO+BRS	72	0.12	27.75	0.04
FIFO+BRS+FDC	66	0.11	26.70	0.04
FIFO+BRS+IEM	54	0.09	22.67	0.04
FIFO+BRS+CTF	42	0.08	19.83	0.03
FIFO+BRS+ALL	41	0.08	17.66	0.03
MST+BRS+ALL	53	0.08	24.77	0.04
SRPT+BRS+ALL	**39**	**0.07**	**19.16**	**0.03**
EDD+BRS+ALL	52	0.08	24.09	0.04
MDD+BRS+ALL	55	0.08	23.95	0.04
CR+BRS+ALL	54	0.08	24.86	0.04
CRSPT+BRS+ALL	59	0.09	18.54	0.03
$S=40$				
FIFO	419	0.27	21.65	0.01
MST	329	0.23	34.53	0.02
SRPT	339	0.23	33.16	0.02
EDD	330	0.23	34.95	0.02
MDD	339	0.24	33.37	0.02
CR	327	0.23	34.99	0.02
CRSPT	407	0.28	29.35	0.02
FIFO+MAL	222	0.16	33.10	0.02
FIFO+DATC	142	0.10	28.76	0.02
FIFO+BRS	115	0.09	27.19	0.02
FIFO+BRS+FDC	97	0.07	25.88	0.02
FIFO+BRS+IEM	83	0.07	22.87	0.02
FIFO+BRS+CTF	64	0.06	18.89	0.02
FIFO+BRS+ALL	69	0.06	18.85	0.02
MST+BRS+ALL	55	0.04	19.63	0.01
SRPT+BRS+ALL	56	0.05	18.83	0.01
EDD+BRS+ALL	**54**	**0.04**	**19.43**	**0.01**
MDD+BRS+ALL	60	0.05	20.02	0.02
CR+BRS+ALL	56	0.04	19.98	0.02
CRSPT+BRS+ALL	74	0.06	18.63	0.01

Table 6
Performance comparison under stable demand according to the number of product types.

Rule+Routing	ET	RER	σ_{ET}	σ_{RER}
$P=5$				
FIFO	236	0.29	20.25	0.03
MST	186	0.24	29.53	0.05
SRPT	193	0.25	28.64	0.04
EDD	186	0.24	29.63	0.05
MDD	198	0.26	29.21	0.04
CR	186	0.24	29.67	0.05
CRSPT	240	0.31	23.56	0.04
FIFO+MAL	166	0.25	34.46	0.06
FIFO+DATC	123	0.19	31.20	0.06
FIFO+BRS	100	0.16	30.35	0.05
FIFO+BRS+FDC	89	0.15	28.78	0.05
FIFO+BRS+IEM	69	0.12	24.58	0.05
FIFO+BRS+CTF	57	0.11	22.29	0.04
FIFO+BRS+ALL	53	0.10	19.96	0.04
MST+BRS+ALL	60	0.11	24.66	0.05
SRPT+BRS+ALL	**51**	**0.09**	**21.61**	**0.04**
EDD+BRS+ALL	59	0.11	24.42	0.05
MDD+BRS+ALL	68	0.12	26.77	0.05
CR+BRS+ALL	61	0.11	24.94	0.05
CRSPT+BRS+ALL	69	0.11	21.28	0.04
$P=15$				
FIFO	263	0.30	18.52	0.03
MST	217	0.26	29.00	0.04
SRPT	219	0.26	28.06	0.04
EDD	217	0.26	29.47	0.04
MDD	221	0.26	28.02	0.04
CR	215	0.25	29.37	0.04
CRSPT	264	0.31	22.27	0.03
FIFO+MAL	143	0.20	29.28	0.05
FIFO+DATC	86	0.12	24.93	0.04
FIFO+BRS	64	0.09	22.75	0.04
FIFO+BRS+FDC	56	0.08	21.58	0.04
FIFO+BRS+IEM	47	0.07	18.37	0.03
FIFO+BRS+CTF	36	0.06	15.72	0.03
FIFO+BRS+ALL	37	0.06	14.64	0.03
MST+BRS+ALL	43	0.07	19.82	0.03
SRPT+BRS+ALL	**34**	**0.06**	**15.84**	**0.03**
EDD+BRS+ALL	42	0.07	19.53	0.03
MDD+BRS+ALL	46	0.07	19.16	0.03
CR+BRS+ALL	44	0.07	19.81	0.03
CRSPT+BRS+ALL	52	0.08	15.89	0.03
$P=30$				
FIFO	254	0.30	17.12	0.03
MST	196	0.24	28.25	0.04
SRPT	201	0.25	27.00	0.04
EDD	196	0.24	28.14	0.04
MDD	198	0.25	26.57	0.04
CR	195	0.24	28.18	0.04
CRSPT	240	0.29	20.62	0.03
FIFO+MAL	131	0.18	27.76	0.05
FIFO+DATC	92	0.13	24.06	0.04
FIFO+BRS	71	0.10	22.88	0.04
FIFO+BRS+FDC	62	0.09	22.12	0.04
FIFO+BRS+IEM	58	0.09	19.93	0.03
FIFO+BRS+CTF	45	0.07	17.21	0.03
FIFO+BRS+ALL	50	0.08	16.74	0.03
MST+BRS+ALL	46	0.07	19.88	0.03
SRPT+BRS+ALL	**42**	**0.07**	**17.02**	**0.03**
EDD+BRS+ALL	45	0.07	19.44	0.03
MDD+BRS+ALL	48	0.08	19.05	0.03
CR+BRS+ALL	47	0.07	19.98	0.03
CRSPT+BRS+ALL	56	0.09	16.36	0.03

number of product types is neither too small nor too big for the proposed strategies to treat jobs of different product types differently. However, this is not yet proved by the simulation results in this study.

Table 7 reveals a contrastive difference between the dispatching rules and combinations of them with routing strategies. The dispatching rules work better in the fixed duedate case than in the variable duedate case, whereas combinations of them with routing strategies work better in the variable duedate case than in the fixed duedate case. Such difference was also noticed in a previous study [18] which concluded that methods using realtime distributed computing were more suitable than scheduling rules for problems involving dynamics and variability.

Table 7
Performance comparison under stable demand according to whether the due date of each product type is fixed or not.

Rule+Routing	ET	RER	σ_{ET}	σ_{RER}
D_p is fixed				
FIFO	180	0.26	16.69	0.03
MST	136	0.21	23.17	0.04
SRPT	144	0.22	21.69	0.04
EDD	136	0.21	23.20	0.04
MDD	146	0.22	21.62	0.04
CR	135	0.21	23.19	0.04
CRSPT	195	0.28	16.93	0.03
FIFO+MAL	162	0.24	31.77	0.06
FIFO+DATC	115	0.17	29.15	0.05
FIFO+BRS	95	0.14	28.40	0.05
FIFO+BRS+FDC	86	0.13	27.75	0.05
FIFO+BRS+IEM	72	0.11	24.66	0.04
FIFO+BRS+CTF	58	0.10	21.68	0.04
FIFO+BRS+ALL	58	0.10	20.79	0.04
MST+BRS+ALL	64	0.11	25.37	0.05
SRPT+BRS+ALL	**57**	**0.09**	**22.54**	**0.04**
EDD+BRS+ALL	63	0.10	25.08	0.05
MDD+BRS+ALL	69	0.11	25.49	0.04
CR+BRS+ALL	65	0.11	25.58	0.05
CRSPT+BRS+ALL	73	0.12	20.43	0.04
D_p is variable				
FIFO	321	0.33	20.58	0.02
MST	263	0.29	34.69	0.04
SRPT	264	0.28	34.10	0.04
EDD	264	0.29	34.94	0.04
MDD	265	0.29	34.25	0.04
CR	261	0.28	34.95	0.04
CRSPT	300	0.33	27.37	0.03
FIFO+MAL	132	0.18	29.28	0.05
FIFO+DATC	86	0.12	24.40	0.04
FIFO+BRS	63	0.09	22.38	0.04
FIFO+BRS+FDC	52	0.08	20.68	0.03
FIFO+BRS+IEM	44	0.07	17.38	0.03
FIFO+BRS+CTF	35	0.06	15.25	0.03
FIFO+BRS+ALL	35	0.07	13.54	0.03
MST+BRS+ALL	36	0.06	17.62	0.03
SRPT+BRS+ALL	**28**	**0.05**	**13.88**	**0.02**
EDD+BRS+ALL	35	0.06	17.24	0.03
MDD+BRS+ALL	39	0.07	17.94	0.03
CR+BRS+ALL	37	0.06	17.65	0.03
CRSPT+BRS+ALL	46	0.07	15.36	0.03

Table 8
Performance comparison under stable demand according to the arrival rate of new jobs.

Rule+Routing	ET	RER	σ_{ET}	σ_{RER}
$\lambda=\lambda_l$				
FIFO	254	0.29	18.99	0.02
MST	212	0.26	30.41	0.04
SRPT	215	0.26	29.20	0.04
EDD	212	0.26	30.42	0.04
MDD	216	0.26	29.22	0.04
CR	211	0.26	30.58	0.04
CRSPT	256	0.31	22.96	0.03
FIFO+MAL	123	0.16	28.09	0.05
FIFO+DATC	79	0.10	23.97	0.04
FIFO+BRS	58	0.08	21.87	0.03
FIFO+BRS+FDC	50	0.07	20.62	0.03
FIFO+BRS+IEM	40	0.06	17.06	0.03
FIFO+BRS+CTF	31	0.05	14.29	0.02
FIFO+BRS+ALL	31	0.05	13.14	0.02
MST+BRS+ALL	32	0.05	17.30	0.03
SRPT+BRS+ALL	**28**	**0.04**	**13.73**	**0.02**
EDD+BRS+ALL	32	0.05	16.90	0.03
MDD+BRS+ALL	36	0.05	18.12	0.03
CR+BRS+ALL	33	0.05	17.42	0.03
CRSPT+BRS+ALL	44	0.06	15.20	0.02
$\lambda=\lambda_h$				
FIFO	247	0.31	18.28	0.03
MST	187	0.24	27.45	0.04
SRPT	192	0.24	26.60	0.04
EDD	187	0.24	27.72	0.04
MDD	195	0.25	26.65	0.04
CR	186	0.24	27.56	0.04
CRSPT	239	0.30	21.33	0.03
FIFO+MAL	171	0.26	32.96	0.06
FIFO+DATC	122	0.19	29.58	0.05
FIFO+BRS	99	0.16	28.90	0.05
FIFO+BRS+FDC	89	0.15	27.81	0.05
FIFO+BRS+IEM	76	0.13	24.98	0.05
FIFO+BRS+CTF	61	0.11	22.64	0.04
FIFO+BRS+ALL	61	0.11	21.20	0.04
MST+BRS+ALL	68	0.12	25.69	0.05
SRPT+BRS+ALL	**57**	**0.10**	**22.68**	**0.04**
EDD+BRS+ALL	66	0.12	25.43	0.05
MDD+BRS+ALL	72	0.13	25.31	0.05
CR+BRS+ALL	69	0.12	25.82	0.05
CRSPT+BRS+ALL	74	0.12	20.58	0.04

Table 8 indicates that combinations of dispatching rules and routing strategies work better under λ_l than under λ_h. This is due to the fact that λ_h is higher than the system processing speed and time is required for the system to respond to fast increase in waiting queue lengths. In contrast, the performance of dispatching rules is a little bit better under λ_h than under λ_l, which is probably due to reduction in earliness.

4.2. Simulation under changing demand

The second part of simulation aims to check the system robustness under disturbances in demand, i.e. the speed at which customer orders arrive. To create such a disturbance, it is designed that λ changes from λ_l to λ_h or vice versa once in the simulation. Since most patterns of stochastic change can be formed by combining a number of step changes, this simulation may reflect to some extent the impact on performance in general changing-demand cases.

According to the results of the first part of simulation, FIFO, CR, FIFO+MAL, FIFO+DATC, FIFO+BRS+ALL and SRPT+BRS+ALL are selected for comparison. Among them, CR represents the best-performing dispatching rule; FIFO+MAL and FIFO+DATC, combinations of dispatching rule with the previous routing strategies; FIFO+BRS+ALL, combination of the same dispatching rule with the proposed strategies; and SRPT+BRS+ALL, the best-performing combination of dispatching rule with routing strategies under stable demand.

The simulation is conducted under each level of S, because S is the factor which is the mostly likely to influence the system performance against disturbances among all the factors in Table 1. The time when the demand changes is set to be the moment at which the 200th new job arrives in the system, and the impact is traced by tracking ET during the simulation process.

Figs. 2-7 show the simulation results, in which the y-axis represents the average ET of the latest 50 jobs among which the last job's index is given by the x-axis. The average ET of 50 jobs is used in the purpose of smoothening the curves.

From Figs. 2–4, it is noticed that ET's of combinations of dispatching rules with routing strategies tend to increase when $\lambda=\lambda_h$ but turn to decrease after λ changes to λ_l. The reason has been discussed in the analysis of Table 8. The turning point at which ET's turn from increase to decrease occurs immediately after λ drops in 10-stage HFSs, but much later in 20- and 40-stage HFSs. This is due to the design of feedback in which information passes stage by stage and accordingly, the more the stages there are in an HFS system, the more the time it takes to respond to a disturbance. But compared with the dispatching rules, combinations of dispatching rules with the proposed routing strategies are much faster in responding speed and much higher in performance.

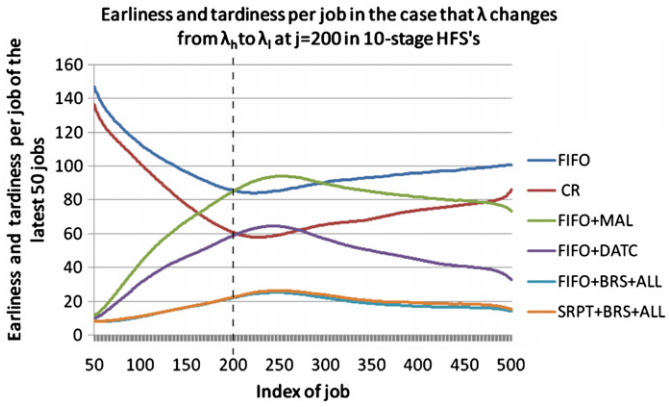

Fig. 2. Trace of earliness and tardiness per job in the simulation for 10-stage HFS problem instances in which λ changes from λ_h to λ_l when the 200th new job arrives.

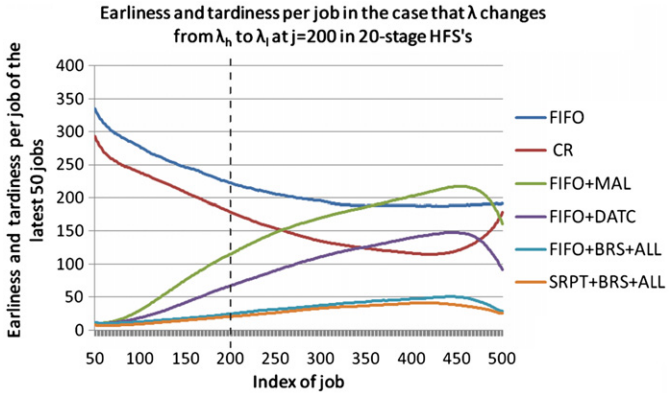

Fig. 3. Trace of earliness and tardiness per job in the simulation for 20-stage HFS problem instances in which λ changes from λ_h to λ_l when the 200th new job arrives.

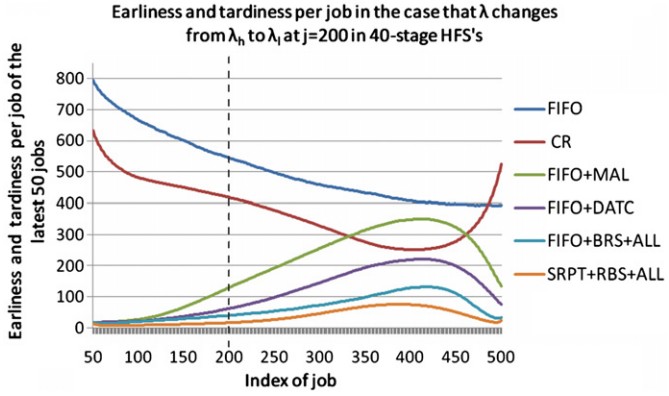

Fig. 4. Trace of earliness and tardiness per job in the simulation for 40-stage HFS problem instances in which λ changes from λ_h to λ_l when the 200th new job arrives.

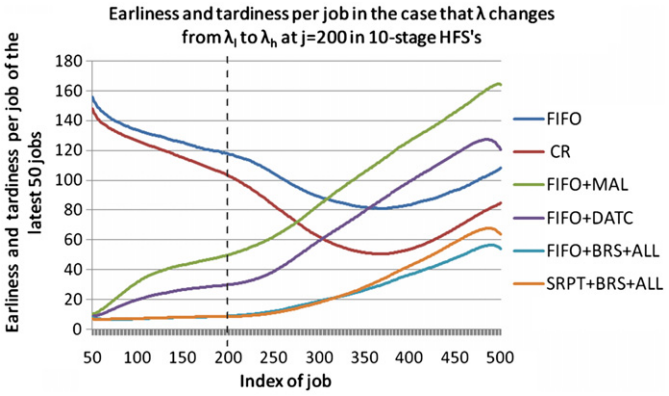

Fig. 5. Trace of earliness and tardiness per job in the simulation for 10-stage HFS problem instances in which λ changes from λ_l to λ_h when the 200th new job arrives.

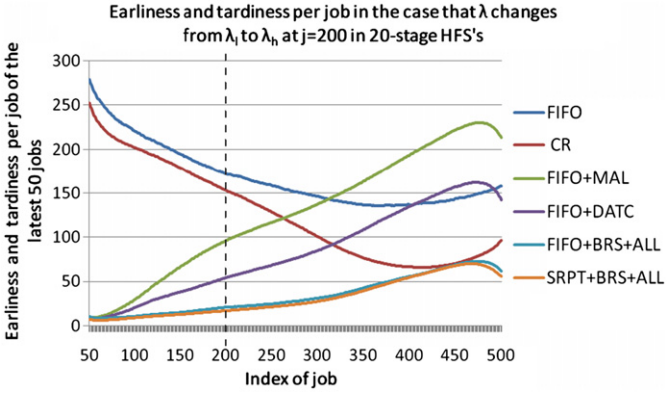

Fig. 6. Trace of earliness and tardiness per job in the simulation for 20-stage HFS problem instances in which λ changes from λ_l to λ_h when the 200th new job arrives.

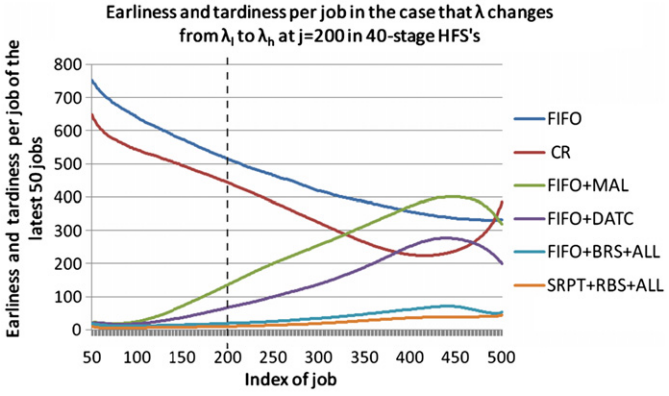

Fig. 7. Trace of earliness and tardiness per job in the simulation for 40-stage HFS problem instances in which λ changes from λ_l to λ_h when the 200th new job arrives.

From Figs. 5–7, it is observed that dispatching rules may not lose much to their partners combined with routing strategies during a sudden rise of λ. Especially, the previous routing methods MAL and DATC are not high-performing after λ changes from λ_l to λ_h. ET's of dispatching rules decrease for a while after λ rises but start to increase again shortly afterwards. This is probably caused by a decrease in earliness at first and an increase in tardiness after that. Comparatively, the performance of dispatching rules combined with the proposed routing strategies is much less disturbed.

5. Conclusions

This paper aims to solve the HFS problem for the objective of JIT job completion by using dynamic scheduling. Routing

strategies are proposed to support dispatching rules to achieve this objective. The advantages include fast computing, quick response and robustness under disturbance. Computer simulations show that with the help of the proposed routing strategies, the performance of dispatching rules is greatly improved and becomes much better than that of the previous routing methods.

The basis of the proposed routing strategies is the forecast of a job's completion time. This provides useful information to both the customer and production manager. The customer can be told an approximate time when the product will become ready, and the production manager can make schedules according to the forecast completion times of jobs being processed.

In addition, the proposed design of feedback and realtime information updating may provide insights to other dynamic problems. The proposed supportive strategies may help create measures which can be taken to reduce the negative impact of disturbance on the performance of production systems. Moreover, the ideas behind and the disturbing phenomena themselves may also be taken into account when solving other scheduling problems and developing robust methods.

Future topics may include extending and developing routing strategies for other multi-stage complex problems such as the flexible job shop problem as well as for other objectives such as those considering weights.

Appendix A

In the simulations, MAL and DATC developed by Brauer and Weiß [2], Prabhu [15] are used for comparison. MAL was originally designed for the objective of minimizing the makespan in HFS. It selected machines by using Eq. (15) and fed back remaining times by using Eq. (16). In the equations, $RT_{p,m}$ represents the remaining time of jobs of product type p after machine m, and n represents the machine which processes job j at stage $s-1$. α is a parameter within the range (0,1). For convenient comparison, the values of the α in Eq. (6) and the α in Eq. (16) are set to be the same in our simulations

$$DM_{j,s} = \arg\min_{m \in M_s} (ST_{j,s,m} + RT_{p,m}), \tag{15}$$

$$RT_{p,n} \leftarrow \alpha \cdot RT_{p,n} + (1-\alpha) \cdot (ST_{j,s,DM_{j,s}} + RT_{p,DM_{j,s}}). \tag{16}$$

To make MAL applicable to the JIT objective, we slightly modified Eq. (15) into Eq. (17) in the simulations

$$DM_{j,s} = \arg\min_{m \in M_s} |T + ST_{j,s,m} + RT_{p,m} - D_j|. \tag{17}$$

DATC was developed for the JIT objective and used Eq. (18) to select machines. In the equation, $G_{p,T}$ is obtained in real time by Eq. (19) in which $J_{p,T}$ represents the number of jobs of product type p which have been completed by time T

$$DM_{j,s} = \arg\min_{m \in M_s} \left| T + ST_{j,s,m} + \sum_{k=s+1}^{S} PT_{p,k} + G_{p,T} - D_j \right|, \tag{18}$$

$$G_{p,T} = \sum_{j=1}^{J_{p,T}} (C_j - D_j)/J_{p,T}. \tag{19}$$

References

[1] Balakrishnan N, Kanet JJ, Sridharan SV. Early/tardy scheduling with sequence dependent setups on uniform parallel machines. Computers and Operations Research 1999;26:127–141.

[2] Brauer W, Weiß G. Multi-machine scheduling—a multi-agent learning approach. In: Proceedings of the third international conference on multi-agent systems; 1998. p. 42–48.

[3] Chiang TC, Fu LC. Using dispatching rules for job shop scheduling with due date-based objectives. International Journal of Production Research 2007;45:3245–3262.

[4] Jabbarizadeh F, Zandieh M, Talebi D. Hybrid flexible flowshops with sequence-dependent setup times and machine availability constraints. Computers and Industrial Engineering 2009;57:949–957.

[5] Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F. A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Computers and Operations Research 2009;36:358–378.

[6] Khalouli S, Ghedjati F, Hamzaoui A. A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop. Engineering Applications of Artificial Intelligence 2010;23:765–771.

[7] Kia HR, Davoudpour H, Zandieh M. Scheduling a dynamic flexible flow line with sequence-dependent setup times: a simulation analysis. International Journal of Production Research 2010;48:4019–4042.

[8] Kim J, Ok C. Distributed feedback control algorithm for dynamic truck loading scheduling problem. Applied Mathematics and Computation 2008;199:275–284.

[9] Liu C-Y, Chang S-C. Scheduling flexible flow shops with sequence-dependent setup effects. IEEE Transactions on Robotics and Automation 2000;16:408–419.

[10] Liu Y, Karimi IA. Scheduling multistage batch plants with parallel units and no intermediate storage. Computers and Chemical Engineering 2008;32:671–693.

[11] Moon DH, Christy DP. A simulation study for dynamic scheduling in a hybrid assembly/job shop considering the JIT context. Production Planning and Control 1998;9:532–541.

[12] Moursli O, Pochet Y. A branch-and-bound algorithm for the hybrid flowshop. International Journal of Production Economics 2000;64:113–125.

[13] Nearchou AC. A differential evolution approach for the common due date early/tardy job scheduling problem. Computers and Operations Research 2008;35:1329–1343.

[14] Parthanadeea P, Buddhakulsomsiri J. Simulation modeling and analysis for production scheduling using real-time dispatching rules: a case study in canned fruit industry. Computers and Electronics in Agriculture 2010;70:245–255.

[15] Prabhu VV. Performance of real-time distributed arrival time control in heterarchical manufacturing systems. IIE Transactions 2000;32:323–331.

[16] Quadt D, Kuhn H. A taxonomy of flexible flow line scheduling procedures. European Journal of Operational Research 2007;178:686–698.

[17] Ruiz R, Vazquez-Rodriguez JA. The hybrid flow shop scheduling problem. European Journal of Operational Research 2010;205:1–18.

[18] Scholz-Reiter B, Rekersbrink H, Gourges M. Dynamic flexible flow shop problems—scheduling heuristics vs. autonomous control. CIRP Annals—Manufacturing Technology 2010;59:465–468.

[19] Sun H, Wang G. Parallel machine earliness and tardiness scheduling with proportional weights. Computers and Operations Research 2003;30:801–808.

[20] Toksarl MD, Guner E. Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach. International Journal of Advanced Manufacturing Technology 2007;38:801–808.