African Journal of Business Management Vol.6 (27), pp. 7995-8009, 11 July, 2012
Available online at http://www.academicjournals.org/AJBM
DOI: 10.5897/AJBM11.477
ISSN 1993-8233©2012 Academic Journals

Full Length Research Paper

Reactive scheduling to minimize makespan of parallel-machine problem with job arrival in uncertainty

Shu-Hsing Chung[1], Ming-Hsien Yang[2], and Ching-Kuei Kao[1]*

[1]Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.
[2]Department of Business Management, National United University, Miao-Li, Taiwan, Republic of China.

Accepted 16 May, 2011

Unpredictable events such as uncertain job arrivals might change the system status or affect the system negatively. Proper actions, such as rescheduling, should be triggered to keep the performance of the system at a specific level. The adoption of the event-driven rescheduling policy counters the impacts of dynamic arrival of jobs, and the parallel insertion algorithm with adjusting procedure is designed to minimize makespan of parallel-machine problem with sequence-dependent setup time. To estimate makespan, probabilistic model is developed with exponentially distributed inter-arrival time and sequence-dependent setup time for identical parallel-machine under First-in First-out (FIFO) rule. The estimated makespan under FIFO can be regarded as a lower level of standard in performance comparison because FIFO is a simple and widely usage dispatching rule, which can be used to evaluate the superiority of the proposed scheduling algorithm. The larger the difference between makespans, respectively determined by the probabilistic model under FIFO and the proposed algorithm, the more superior algorithm can be concluded. Comparative computations are provided to demonstrate the effectiveness of the proposed algorithm and the accuracy of the probabilistic model in estimating makespan and setup time.

Key words: Parallel machine, dynamic events, rescheduling, makespan, probabilistic model.

INTRODUCTION

The identical parallel-machine scheduling problem is the common system in many real-world applications, especially in the integrated circuit (IC) manufacturing, and packaging industries. Makespan is the maximum completion time for all jobs to be processed and is often used to measure of schedule efficiency when generating a production schedule, which is an objective usually being taken to avoid extreme uneven utilization of machine capacity. Consider the identical parallel-machine problem with minimizing makespan, which has been proved to be a NP problem (Sethi, 1977; Garey and Johnson, 1979). The longest processing time (LPT) rule (Graham, 1969)), a popular dispatching rules, has been applied to this problem and has been shown to perform well among several popular rules. Furthermore, this problem can been solved by operational research methods such as integer programming, branch and bound, dynamic programming (Potts, 1983; Bernstein et al., 1985; Luh et al., 1988; Cheng and Sin, 1990). In this paper, the parallel insertion algorithm with adjusting procedure (PIAAP) is proposed to solve this problem.

In the dynamic manufacturing environment, unpredictable events exist such as jobs arriving at the production system dynamically, which cause uncertainties on the arrival time of jobs. Proper actions, such as rescheduling, should be triggered to prevent the significant reduction of the performance of the system (Vieira et al., 2003). Rescheduling is the process of updating an existing production schedule in response to unpredictable events to minimize its impact on system performance. The three types of rescheduling policies that have been studies are periodic, event-driven, and hybrid. Vieira et al. (2000a) developed the mathematical model to predict the performance of a single machine system under the former two rescheduling policies. Vieira et al. (2000b)

*Corresponding author. E-mail: ckkao.iem91g@nctu.edu.tw Tel: 886-5731638. Fax: 886-5722392.

studied all rescheduling policies for parallel machine systems to predict the system performance. To respond to the impacts of dynamic arrival of jobs, PIAAP adopts event-driven rescheduling policy when the number of job arriving at the system reaches a threshold.

To evaluate the superiority of the proposed reactive PIAAP, a probabilistic model is developed to estimate the makespan in this paper. Regarding makespan estimation, Lee and Pinedo (1997) suggested that the estimated makespan equals to the multiplication of the average time to process a job, including setup time, by the average number of jobs to be processed on one machine. Chakravarty and Balakrishnan (1996) proposed that the estimated makespan equals the sum of all job-processing times plus the expected machine down-time for job scheduling on flexible machines with machine breakdown and negligible setup time. Raaymakers et al. (2001) developed the regression model to estimate the makespan in batch process industries based on the aggregate resource and the job set characteristics. In addition to the regression model, Raaymakers and Weijters (2003) used the neural network model to estimate the makespan, and the performance of neural network model is significantly better than the performance of regression model. However, the makespan may include setup time, especially being sequence-dependent, which was seldom considered in the model of estimating makespan. In this paper, the estimated makespan is defined by the summation of the arrival time of the last job, the average waiting time of jobs in queue, and the average service time of jobs on the machine. The service time of a job is defined as the sum of its processing time plus its setup time. The sequence-dependent setups are considered for scheduling two consecutive jobs of different product types on one machine and First-in first-out (FIFO) rule is applied to dispatch jobs.

To evaluate the accuracy of the probabilistic model on estimating makespan and the performance of the proposed reactive PIAAP, several simulation models are built to generate the makespan, in which jobs are dispatched according to the corresponding rules or scheduled by the proposed algorithm and jobs are carrying with exponentially distributed inter-arrival time. The simulation results (for example, makespan) are collected for jobs arriving in a time horizon (called run time in the following sections) with various machine utilization rates, total arrival rates, and the number of machines. Since FIFO is a simple rule without considering the enhancement of scheduling-related performances, the estimated makespan generated by the probabilistic model can be regarded as a lower level of standard in performance comparison, which can be used to evaluate the superiority of the proposed scheduling algorithm for reducing the makespan of dynamic parallel-machine problem. With the objective of minimizing makespan, the larger the difference between the makespans respectively determined by the probabilistic model under FIFO rule and the algorithm, the more superior algorithm can be concluded. Comparative computations are provided to demonstrate the effectiveness of the proposed algorithm. Figure 1 illustrates the architecture of this paper.

REACTIVE ALGORITHM

To find a schedule with lower makespan for the considered problem with uncertain job arrivals, PIAAP with reactive mechanism is developed. Reactive mechanism is triggered when the number of arriving job in the buffer of unscheduled jobs reaches a specific level. Then, PIAAP is activated to regenerate a new schedule by inserting unscheduled jobs into the feasible positions in the current schedules of parallel machines with the help of a regret measure to detect insertion problems instead of solely considering of lowest cost insertion.

Parallel insertion algorithm with adjusting procedure

To solve jobs scheduling on parallel machines with minimum total workload by reducing setup time, several types of parallel insertion algorithm (PIA) were developed by Pearn et al. (2008), which was the application of the approach of Potvin and Rousseau (1993) for vehicle routing problem. However, as a schedule solution with lowest total workload for parallel machines may imply unbalanced workload on all machines and then increase the makespan, this paper develops an adjusting procedure to be implemented iteratively with PIA to reduce makespan.

Parallel insertion algorithm (PIA)

PIA considers several product types of jobs to be processed on the identical parallel machines with capacity constraint. Initially, PIA generates a set of K machine schedules (each includes a single job and K indicates the total number of machines). Then, to augment the current set of schedules, PIA evaluates the regret value of not inserting a job into the lowest insertion cost position immediately. That is, a job with largest regret value should be considered first to be inserted into the schedule. Let the schedule of machine m_k be represented by $(u_{0k}, u_{1k}, ..., u_{(p-1)k}, u_{pk}, u_{(p+1)k}, ...)$, where u_{0k} represents the idle status of machine m_k, and u_{pk} indicates the job be scheduled at the pth position on machine m_k. The cost formula $c_1(u,k,p)$ in Equation (1) evaluates the additional setup time (cost) by inserting a job u between two adjacent jobs $u_{(p-1)k}$ and u_{pk} feasibly on the schedule of machine m_k, in which $s_{u_{(p-1)}u}$ indicates the required setup time for two consecutive jobs $u_{(p-1)k}$ and u_k on machine m_k. Let $c_1'(u,k,p^*)$ and $c_1''(u,k^*,p^*)$

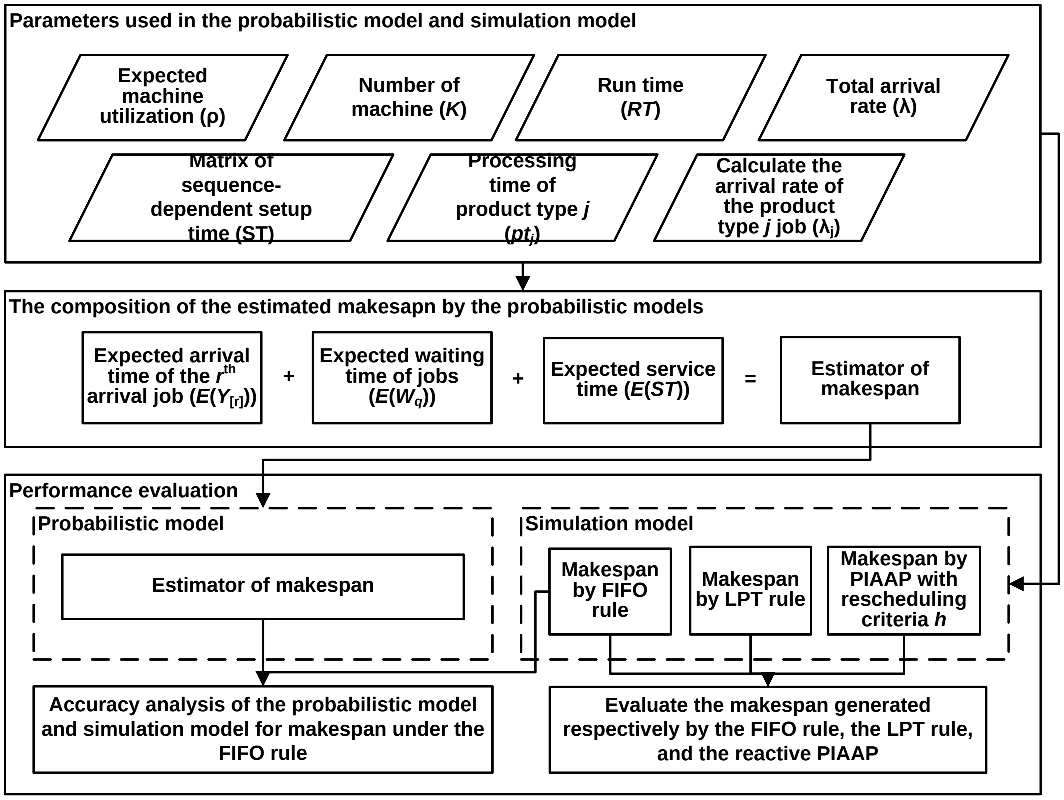

Figure 1. Paper architecture.

in Equations (2) and (3), respectively, represent the minimum additional setup time position p^* in all positions on machine m_k and the minimum additional setup time position p^* at machine k^* among all schedules. The regret measure $c_2(u)$ in Equation (4) denotes the regret value of job u by summarizing the differences of cost between the best and the second best alternative insertion position. Therefore, the PIA selects the job u^* with largest regret value $c_2(u^*)$, as stated in Equation (5), to be inserted into its best insertion position:

$$c_1(u,k,p) = c_1(u_{(p-1)k}, u_k, u_{pk}) = s_{u_{(p-1)}u} + s_{uu_p} - s_{u_{p-1}u_p} \quad (1)$$

$$c_1'(u,k,p^*) = \min_p [c_1(u,k,p)] \quad , \quad k = 1,2,...,K \quad (2)$$

$$c_1''(u,k^*,p^*) = \min_k [c_1'(u,k,p^*)] \quad (3)$$

$$c_2(u) = \sum_{k \neq k^*} \left[c_1'(u,k,p^*) - c_1''(u,k^*,p^*) \right] \quad (4)$$

$$c_2(u^*) = \max_u [c_2(u)] \quad (5)$$

The procedure of PIA is described further.

Step 1: Initialize the schedule on each machine by selecting K jobs with the first K largest values for the sum of initial setup time and job processing time.
Step 2: Find the best feasible insertion position by computing $c_1''(u,k^*,p^*)$ for each unscheduled job.
Step 3: Calculate the regret value $c_2(u)$ for each unscheduled job. Select the job u^* with the largest regret measure $c_2(u^*)$ among all unscheduled jobs. Insert it into the p^*th position of machine k^* without violating the machine capacity and its due date restrictions.
Step 4: Repeat Steps 2 and 3 until all jobs are scheduled.

Adjusting procedure of PIAAP

Since PIA tries to minimize the total machine workload, an adjusting procedure to be executed with PIA is developed to make PIA be applicable for reducing the makespan of the considered problem. The adjusting procedure is implemented iteratively basing on the three point equal-interval search to decrease the expected machine load, which can be used as the capacity constraint of machines to reduce makespan.

Let EL be the expected machine load and EL^{δ} be the expected machine load at the δ^{th} adjusting of EL. Let ML_k^{δ} be the actual workload on machine m_k, which is

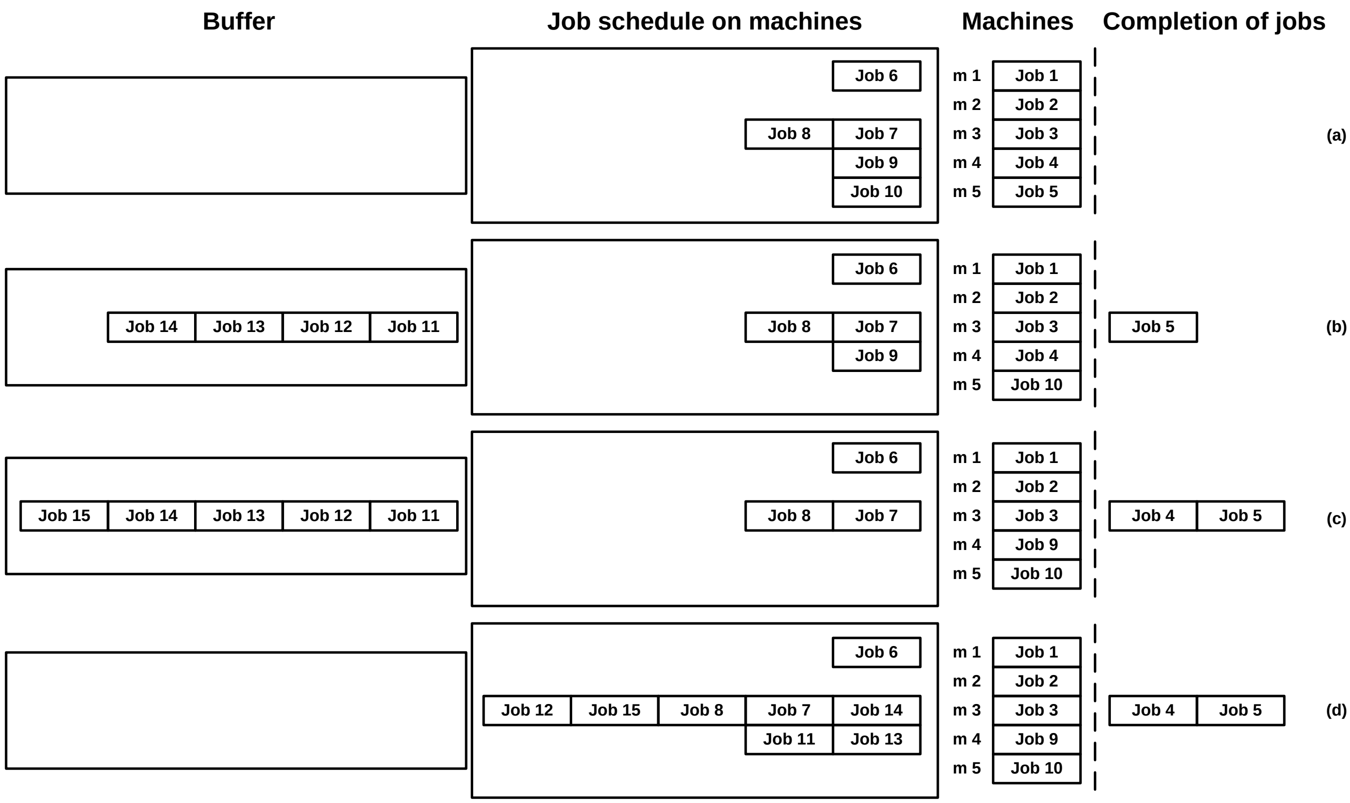

Figure 2. Illustrative example of rescheduling

determined by the machine schedules solved by PIA at the δ^{th} adjusting. LB^{δ} and UB^{δ} respectively represent the lowest and the highest machine workload among the current generated parallel-machine schedule at the δ^{th} adjusting of EL, in which $LB^{\delta} = \min_{k \in K}\{ML_k^{\delta}\}$ and $UB^{\delta} = \max_{k \in K}\{ML_k^{\delta}\}$. Predefine the stopping criterion of the adjusting procedure as the condition when the absolute difference of LB^{δ} and UB^{δ} is less than a specific magnitude. The steps of the adjusting procedure are described thus.

Step 1: Let δ be zero and set EL^0 be equal to EL, which is given initially as the capacity constraints for all machines. Use PIA to generate a solution for the parallel-machine problem by adding EL^0 as the capacity constraint.
Step 2: Define LB^0 as the initial lower bound and UB^0 as the initial upper bound for the machine workload in the solution generated by PIA.
Step 3: If $\left|UB^0 - LB^0\right|$ is less than the stopping criterion and the scheduling solution is feasible, stop the procedure. Otherwise, go to next step.
Step 4: Update δ ($\delta = \delta + 1$) and replace the expected machine load EL^{δ} with $(LB^{\delta'} + UB^{\delta'})/2$, where $\delta' = \delta - 1$.
Step 5: Repeat the algorithm PIA for the parallel-machine problem with EL^{δ} as the adjusted capacity constraint. Consider the following two cases.
(1) If there is feasible solution generated, set UB^{δ} be equal to EL^{δ}.
(2) If it is not possible to find feasible solution, set LB^{δ} be equal to EL^{δ}.
Step 6: If $\left|UB^{\delta} - LB^{\delta}\right|$ is less than the stopping criterion and the δ^{th} scheduling solution is feasible, stop the procedure. Otherwise go to step 4.

Reactive mechanism in PIAAP

To respond to the impacts of dynamic arrival of jobs, PIAAP adopts an event-driven rescheduling policy, which is activated whenever the number of arriving jobs in the buffer reaches a specific level, for example, *h*. Figure 2 presents an illustrative example to show how the rescheduling policy cooperates with PIAAP. Twenty jobs required to be scheduled on five parallel machines, and the rescheduling trigger point is set to five jobs (h = 5).

Figure 2a indicates the initial status of the rescheduling example, in which Job 1 ~ Job 5 are processing on the machines m_1 ~ m_5 and Job 6 ~ Job 10 are dispatched to machines m_1 ~ m_5 according to the job schedule

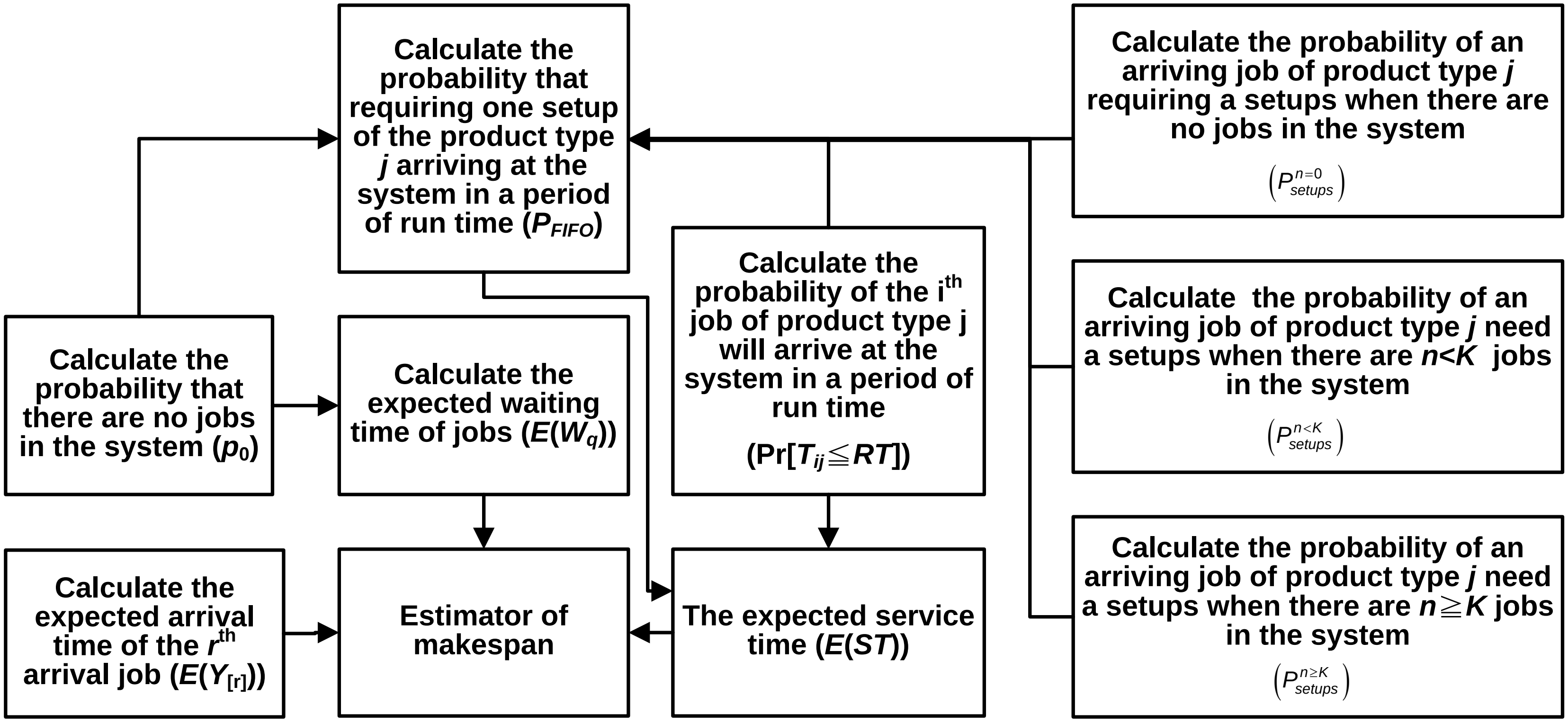

Figure 3. Procedure of calculating estimator of makesapn.

determined by PIAAP.

In Figure 2b, Job 5 is processed and leaves machine m_5. Once the machine m_5 is idle, Job 10 is dispatched subsequently to machine m_5 according to the job schedule on machines. Meanwhile, another four jobs (Jobs 11 to 14) have arrived at the system but PIAAP is not activated yet due to the number of jobs in the buffer is less than h (= 5).

Figure 2c indicates the condition to satisfy the rescheduling criterion when the number of jobs in the buffer is equal to h (= 5). However, before satisfying the rescheduling criterion, Job 4 is processed and leaves machine m_4 and Job 9 is subsequently dispatched to machine m_4. Figure 2d illustrates the updated schedule on machines, in which the unscheduled jobs (Jobs 11 to 15) in the buffer are inserted into the machine schedule as shown in Figure 2c by applying PIAAP.

ESTIMATOR OF MAKESPAN

Here, the probabilistic model is derived to estimate makespan for identical parallel-machine problem with dynamic jobs arrivals, in which the inter-arrival time of jobs is exponentially distributed and FIFO is applied to dispatch jobs. Figure 3 shows the calculating procedure of the estimated makesapn and the calculations in Figure 3 are presented further. In describing the estimator of makespan the following notation is used:

RT: Period of run time
K: Number of parallel machines
Λ_j: Arrival rate of type j
Λ: Total arrival rate
pt_j: Processing time of product type j job
s_{jr}: Setup time for product type j job after product type r job
S: Setup time of jobs

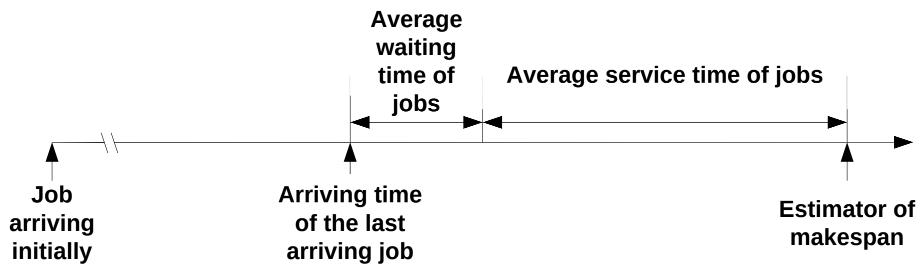

Figure 4. Diagram of estimator of makespan.

ST_{ij}: Service time of the ith job of product type j
ST: Service time of jobs
W_q: Waiting time in queue of jobs
Yi: Arrival time of jobs
$Y_{[r]}$: Arrival time of the rth arrival job
n: Number of jobs in the system
n_j: Number of type j jobs arrived in the time interval [0, RT]
N: Number of all jobs arrived in the time interval [0, RT]
N'(t): Number of arriving jobs different from product type j by time t
p_n: Probability that there are n jobs in the system
P_{FIFO}: Probability that requires one setups of the product type j under FIFO
P^n_{setups} : Probability of an arriving job of product type j need a setups when there are n jobs in the system
f(x; a, b): Gamma distribution with parameters a and b
Y: Binomial random variable with parameters (n, λj/λ)
Y': Binomial random variable with parameters (K − 1, λj/λ)
ρ: Machine utilization rate

Consider identical parallel machines in which jobs with different product types dynamically arrive in a period of run time (*RT*) for processing according to FIFO, and sequence-dependent setups occur when job changes from one product type to another. The inter-arrival time for jobs of different product type is independent and exponentially distributed with the parameter λ_j, where $j = 1, 2, \ldots, J$ and J is the number of product type. Total arrival rate is $\lambda = \sum_{j=1}^{J} \lambda_j$.

According to FIFO, in dynamic parallel-machine problem, makespan can be estimated with the summation of the arrival time of the last arriving job, the average waiting time for jobs in queue, and the average service time for jobs, as illustrated as Figure 4. However, the completion time of the last arriving job is not necessarily equal to the makespan for the dynamic parallel-machine problem. Therefore, the estimated makespan can be defined as Equation (6):

$$\hat{C}_{\max} = \left(\sum_{z=0}^{K-1} E\left[Y_{[N-z]} \right] \right) K^{-1} + E\left(W_q\right) + E\left(ST\right) \quad (6)$$

where N indicates that there are N jobs arrived in the time interval [0, *RT*], $E[Y_{[N\text{-}z]}]$ is the expected arrival time of the $(N-z)^{\text{th}}$ arrival job, $E(W_q)$ is the expected waiting time in queue of jobs, and $E(ST)$ is the expected service time of jobs. The calculation of $E[Y_{[N\text{-}z]}]$, $E(W_q)$ and $E(ST)$ are presented in the following sub-sections.

Expected arrival time of jobs

Suppose that the inter-arrival time of the product type *j* job is independent and exponentially distributed with the parameter λ_j and n_j jobs would arrive in *RT* for product type *j*, where $j = 1, 2, \ldots, J$. Therefore, $N = \sum_{j=1}^{J} n_j$ jobs would arrive in *RT* for all product type. Let $Y_1, Y_2, \ldots, Y_N$ be the arrival time of jobs under the condition that N jobs have arrived in the time interval [0, *RT*] having an uniformly distributed over [0, *RT*]. The probability density function of Y_i is shown as $f(y_i) = 1/RT$, where $i = 1, 2, \ldots, N$.

Let $Y_{[r]}$ be the arrival time of the r^{th} arrival job under the condition that there are N jobs arriving in *RT*. Therefore, the probability density function of $Y_{[r]}$ can be shown as Equation (7):

$$g\left(y_{[r]}\right) = \frac{N!}{(r-1)!(N-r)!} \left(\frac{y_{[r]}}{RT} \right)^{r-1} \frac{1}{RT} \left(1 - \frac{y_{[r]}}{RT} \right)^{N-r} \quad (7)$$

According to Equation (7), the expected value of $Y_{[r]}$ can be calculated as $E[Y_{[r]}] = \int_0^{RT} y_{[r]} g(y_{[r]}) dy_{[r]} = (r/(N+1))RT$, which implies that there are N jobs arriving in *RT* and the time interval [0, *RT*] is divided into $(N+1)$ equal subintervals. The expected elapsed times between any two consecutive jobs arrive at the system equals the range of subinterval.

Expected waiting time of jobs

Consider jobs arriving at a Poisson rate $\lambda = \sum_{j=1}^{J} \lambda_j$ and are served by any of K machines, each of which has the general service time distribution.

Therefore, the queuing system is the model *M*/*G*/*K*. Hokstad (1978) proposed an approximation for the queue length distribution in queue for *M*/*G*/*K*, showed as Equations (8) to (10):

$$p_0 = \left[\sum_{n=0}^{K-1} \frac{\left[\lambda E\left(ST\right)\right]^n}{n!} + \frac{\left[\lambda E\left(ST\right)\right]^K}{(1-\rho)K!} \right]^{-1} \quad (8)$$

$$p_n = \frac{\left[\lambda E\left(ST\right)\right]^n}{n!} p_0 \quad , n = 1, 2, \ldots, K-1 \quad (9)$$

$$P(z)=\sum_{n=K}^{\infty}p_n z^{n-K}=\left[\frac{1-B^*\left(\dfrac{\lambda(1-z)}{K}\right)}{B^*\left(\dfrac{\lambda(1-z)}{K}\right)-z}\right]p_{K-1} \quad (10)$$

According to Equations (8) to (10), the expected number of waiting jobs in queue ($E(L_q)$) is equal to $(\rho C_s^2 \prod)/[2(1-\rho)]$. By Little's law, an approximation for the expected waiting time of jobs in queue for $M/G/K$ ($E(W_q)$) can be calculated as Equation (11):

$$E(W_q)=\frac{E(L_q)}{\lambda}=\frac{\lambda^K E(ST^2)\left[E(ST)\right]^{K-1}}{2\left[K-\lambda E(ST)\right]^2(K-1)!}p_0 \quad (11)$$

where $E(ST^t)$ is the t^{th} moment of the service time of jobs, $B^*(\lambda(1-z)/K)$ is the Laplace transform of the probability function of the service time of jobs, $\rho=\lambda E(ST)/K$, $C_s^2=E(ST^2)/[E(ST)]^2$ and $\Pi=[[\lambda E(ST)]^K/[(1-\rho)K!]]p_0$. For further details about the mathematical proof in Equation (11) (Hokstad, 1978; Nokaki and Ross, 1978).

Expected service time of jobs

Consider identical parallel machines in which jobs with different product types arrive dynamically in *RT*. Jobs are processed according to FIFO on machines in the presence of sequence-dependent setup time. Suppose that the service time of jobs equals its processing time plus its setup time. Therefore, the calculation of this time can be divided into three situations. First, the service time of jobs would be zero under the condition that no jobs arrive in *RT*. Second, the service time of jobs would be equal its processing time with the condition that one job arrives within a period of *RT* and the new arrival job needs no setup. Third, the service time of jobs would be equal its processing time and its setup time with the condition that one job arrives in *RT* and this job requires a setup.

Suppose that the inter-arrival time of product type *j* job is independent and exponentially distributed with the parameter λ_j, and then the time until the the i^{th} job of product type *j* is arrived at the system (T_{ij}) is the gamma distribution with parameters λ_j and *i*. The probability of the i^{th} job of product type *j* will arrive at the system in *RT* can be calculated as $\Pr[T_{ij}\le RT]=\int_0^{RT}[\lambda_j^i/\Gamma(i)]t_{ij}^{i-1}e^{\lambda_j t_{ij}}dt_{ij}$. Let ST_{ij} be the random variable of the service time of the i^{th} job of product type *j*, where $j=1, 2,\ldots, J$. The probability mass function of ST_{ij} can be shown as Equation (12):

$$P(ST_{ij}=st_{ij})=\begin{cases}\Pr\left[T_{ij}\le RT\right](1-P_{FIFO}) & \text{, if } st_{ij}=pt_j\\ \Pr\left[T_{ij}\le RT\right]P_{FIFO}\dfrac{\lambda_r}{\lambda'} & \text{, if } st_{ij}=pt_j+s_{jr},\ r=1,2,\ldots,J,\ r\ne j\\ 1-\Pr\left[T_{ij}\le RT\right] & \text{, if } st_{ij}=0\end{cases} \quad (12)$$

where $r=1, 2,\ldots, J$, $r\neq j$, pt_j represents the processing time of product type *j*, and s_{jr} represents the setup time of product type *j* job, in which the previous job belongs to product type *r*. P_{FIFO} represents the probability that requires one setups of the product type *j*, where jobs are dispatched by FIFO, λ_r/λ' represents the probability that the previous job processed on the machine belong to product type *r*, and $\lambda'=\sum_{r=1,r\neq j}^{J}\lambda_r$. Suppose that there would arrive n_j independent jobs of product type *j* in *RT* and the mean of service time of jobs is defined as $ST=\sum_{j=1}^{J}\sum_{i=1}^{n_j}ST_{ij}\Big/\sum_{j=1}^{J}n_j$. The expected service time of jobs $E(ST)$ can be calculated by Equation (13), based on Equation (12):

$$E(ST)=\left[\sum_{j=1}^{J}\sum_{i=1}^{n_j}\Pr\left[X_{ij}\le RT\right]\left(pt_j+P_{FIFO}\sum_{\substack{r=1\\ r\neq j}}^{J}\frac{\lambda_r}{\lambda'}s_{jr}\right)\right]\left(\sum_{j=1}^{J}n_j\right)^{-1} \quad (13)$$

The probability P_{FIFO} has to be calculated in advance when calculating the expected service time of jobs in Equation (13). The calculation of the probability P_{FIFO} is presented further.

Probability P_{FIFO}

The probability P_{FIFO} represents the probability that requires one setup of the product type *j* arriving at the system in *RT* under FIFO and can be written as Equation (14) based on the number of jobs in the system:

$$P_{FIFO}=p_0P_{setups}^{n=0}+\sum_{n=1}^{K-1}p_nP_{setups}^{n<K}+\sum_{n=K}^{\infty}p_nP_{setups}^{n\ge K} \quad (14)$$

where p_0 and p_n are the probabilities of no jobs and *n* ($n\geqq1$) jobs in the system and are given by Equations (8) to (10). In Equation (14), $P_{setups}^{n=0}$, $P_{setups}^{n<K}$, and $P_{setups}^{n\ge K}$ are the probabilities of an arriving job of product type *j* need a setups when there are no jobs, there are *n* ($n<K$) jobs, and there are *n* ($n\ge K$) jobs in the system. The calculations of these three probabilities are presented thus:

i. suppose that there are no jobs in the system and a job of product type *j* is arriving. A setup is required if there are no jobs of product type *j* among the last *K* jobs being processed on *K* parallel machines, where *K* is the number of machines.

This means that there are at least *K* jobs different from type *j* arrived at the system between this arriving jobs of type *j* and its predecessor, which can be shown as $N'(t+t_j)-N'(t)\geqq K$, where $N'(t)$ is the number of arriving jobs different from product type *j* by time *t* and is a Poisson process having rate λ', and t_j is the value of the inter-arrival time of type *j* (T_j). Therefore, the probability of an arriving job of product type *j* requires a setup given by $P_{setups}^{n=0}=\Pr[N'(t+t_j)-N'(t)\ge K\,|\,T_j\le RT]$.

The probability of an arriving job of product type *j* which does not need setups is calculated by Equation (15):

ii. suppose that there are *n* ($n<K$) jobs in the system and a job of product type *j* is arriving at the system, in which there are *n* jobs being processed on machines and ($K-n$) machines currently being idle. A setup is required if there are no jobs of product type *j* in the last jobs being processed on the ($K-n$) machines currently being idle. Suppose that *Y* is the random variable representing the number

$$P_{not\ need\ setups}^{n=0} = 1 - P_{setups}^{n=0} = \frac{1}{1-e^{-\lambda_j RT}} \sum_{\alpha=0}^{K-1} \frac{\lambda_j}{\lambda}\left(1-\frac{\lambda_j}{\lambda}\right)^{\alpha} \int_0^{RT} f\left(\tau;\alpha+1,\lambda\right)d\tau \qquad (15)$$

of type j jobs among the n jobs in the system, and then Y is the binomial random variable with parameters (n, λ_j/λ). For the situation that one type j job is arriving when there are n jobs already in the system with y jobs of type j, where y is the value of the random variable Y, setup would be necessary for this job if there are at least ($K-n$) jobs different from product type j arrived during the time until the $(y+1)^{th}$ job of product type j arrives, which is given by $N'(t+t_{(y+1)j}) - N'(t) \geqq K-n$. Therefore, the probability of a type j job requiring a setup is calculated by Equation (16):

$$P_{setups}^{n<K} = \sum_{y=0}^{n} P\left(Y=y\right)P\left[N'\left(t+t_{(y+1)j}\right)-N'\left(t\right)\geq K-n \middle| T_{(y+1)j} \leq RT\right] \qquad (16)$$

By Equation (16), the probability of not requiring setups for this job is computed by Equation (17):

$$\begin{aligned} P_{not\ need\ setups}^{n<K} &= \sum_{y=0}^{n} \Pr\left(Y=y\right)\Pr\left[N'\left(t+t_{(y+1)j}\right)-N'\left(t\right)\leq K-n-1 \middle| T_{(y+1)j} \leq RT\right] \\ &= \frac{\lambda_j}{\lambda}\sum_{y=0}^{n} P\left(Y=y\right) \sum_{\alpha'=0}^{K-n-1} \binom{\alpha'+y}{y}\left(\frac{\lambda'}{\lambda}\right)^{\alpha'}\left(\frac{\lambda_j}{\lambda}\right)^{y} \frac{\int_0^{RT} f\left(\tau';\alpha'+y+1,\lambda\right)d\tau'}{\Pr\left(T_{(y+1)j}\leq RT\right)} \end{aligned} \qquad (17)$$

iii. suppose that there are n ($n \geqq K$) jobs in the system and a job of product type j is arriving at the system. let Y' be a Binomial random variable with parameters ($K-1$, λ_j/λ). If a type j job arrives at the system under the condition that there are y' jobs of type j being processed on ($K-1$) machines, there would be no setup required for this job while the last job being processed on the currently idle machine belongs to product type j. We can get ($y'+1$) jobs of type j and ($K-y'-1$) jobs different from type j. Thus it can be seen that there are ($K-y'-1$) jobs different from type j arrived at the system during the time until ($y'+1$) jobs of type j arrived at the system, which can be expressed as $N'(t+t_{(y'+1)j}) - N'(t) = K-y'-1$. Therefore, the probability of not requiring setups for this job is calculated by Equation (18):

$$\begin{aligned} P_{not\ need\ setups}^{n\geq K} &= \sum_{y'=0}^{K-1} \Pr\left(Y'=y'\right)\Pr\left[N'\left(t+t_{(y'+1)j}\right)-N'\left(t\right)= K-y'-1 \middle| T_{(y'+1)j} \leq RT\right] \\ &= \frac{\lambda_j}{\lambda}\sum_{y'=0}^{K-1} P\left(Y'=y'\right) \binom{K-1}{y'}\left(\frac{\lambda_j}{\lambda}\right)^{y'}\left(\frac{\lambda'}{\lambda}\right)^{(K-1)-y'} \frac{\int_0^{RT} f\left(\tau'';K,\lambda\right)d\tau''}{\Pr\left(T_{(y'+1)j}\leq RT\right)} \end{aligned} \qquad (18)$$

The probability of requiring a setup for this job is calculated by $P_{setups}^{n\geq K} = 1 - P_{not\ need\ setups}^{n\geq K}$. In Equations (15), (17), and (18), $f(x; a, b)$ is the gamma distribution with parameters a and b and $\lambda' = \sum_{r=1, r\neq j}^{J} \lambda_r$.

Then the probability P_{FIFO} can be easy to reformulate as the Equation (19) by using Equation (10):

$$P_{FIFO} = p_0 P_{setups}^{n=0} + p_0 \sum_{n=1}^{K-1} \frac{\left[\lambda E\left(ST\right)\right]^n}{n!} P_{setups}^{n<K} + P_{setups}^{n\geq K} \frac{\left[\lambda E\left(ST\right)\right]^K}{\left(1-\rho\right)K!} p_0 \qquad (19)$$

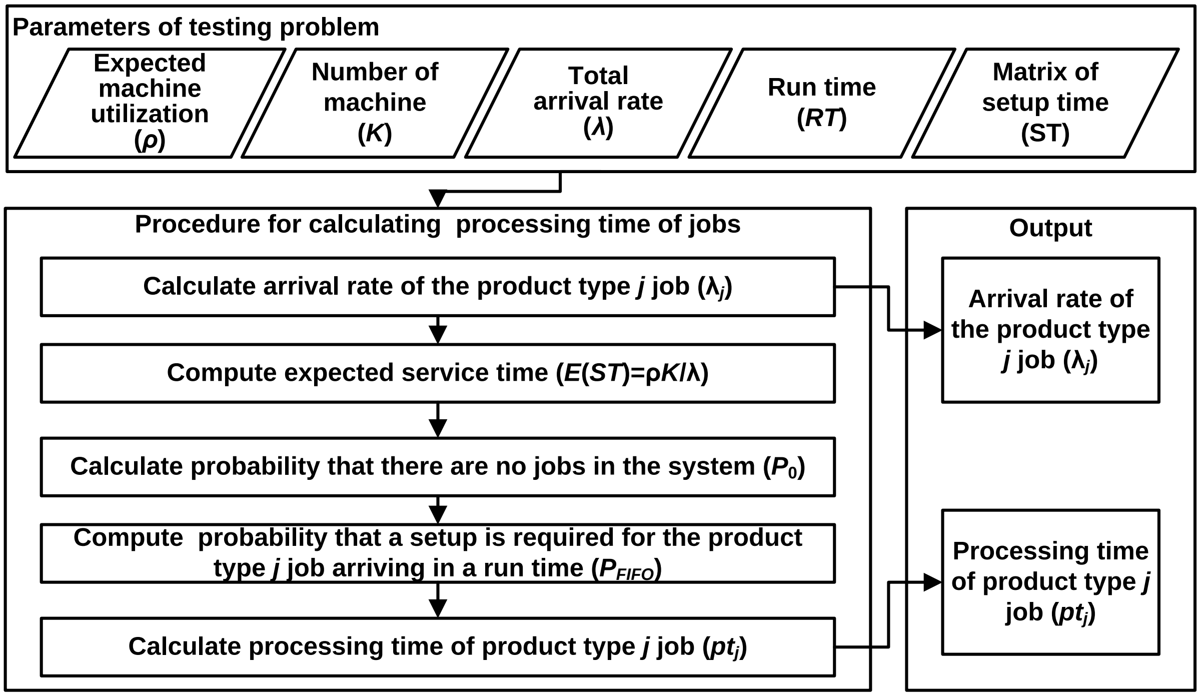

Figure 5. Procedure for calculating jobs processing time.

COMPUTATIONAL RESULT

To verify the superiority of the proposed algorithm (PIAAP) by comparing with FIFO and LPT in reducing makespan and the accuracy of the proposed probabilistic model on estimating makespan, an experimental design is conducted by varying five control parameters. Three simulation models are built, in which the first adopts FIFO and the second adopts LPT to dispatch jobs to machines, and the third dispatches jobs according to the parallel-machine schedule generated by PIAAP. In the simulation models, the inter-arrival time of jobs is independently and exponentially distributed. The estimated makespan calculated by the probabilistic model under FIFO is compared with the makespan determined by the simulation model under FIFO to evaluate the accuracy of the probabilistic model on estimating makespan.

An experiment includes five factors, six levels of the expected machine utilization rate (ρ): 0.95, 0.90, 0.85, 0.80, 0.75, and 0.70; two levels of rum time (RT): 14400 and 21600 (seconds); two levels of the number of machine (K): 2 and 5; two levels of the total arrival rate (λ): 1 (1 jobs in 60 seconds) and 3 (3 jobs in 60 seconds); four levels of the rescheduling criterion (h): 5, 10 15, and 20 (jobs), in which there would be 192 combinations in the experiment. Note that jobs of eight different product types ($J=8$) will arrive dynamically for each combination and each combination would be collected after 10,000 independent simulation runs. For each combination, the simulation models include the matrix of the sequence-dependent setup time for switching product types on a machine as shown as ST:

$$\mathbf{ST}=\begin{bmatrix} 0 & 5 & 10 & 15 & 20 & 15 & 20 & 5 \\ 30 & 0 & 20 & 15 & 25 & 10 & 15 & 10 \\ 20 & 25 & 0 & 30 & 15 & 10 & 20 & 5 \\ 5 & 10 & 15 & 0 & 15 & 10 & 5 & 10 \\ 5 & 15 & 30 & 15 & 0 & 25 & 15 & 20 \\ 10 & 15 & 30 & 10 & 15 & 0 & 5 & 10 \\ 15 & 30 & 25 & 20 & 25 & 20 & 0 & 15 \\ 10 & 15 & 20 & 15 & 5 & 25 & 15 & 0 \end{bmatrix}$$

Furthermore, to be consistent with the specific utilization rate of machine (ρ) in an experiment, the processing time of product type j job (pt_j) is calculated as the procedure shown in Figure 5. Based on ρ, RT, K, and λ, the arrival rate of type j job (λ_j) can be derived as $\lambda_j=(u_j/\sum_{j=1}^{J}u_j)\lambda$, where $j=1, 2,\ldots, J$, and u_j is the random number chosen from the beta distribution with parameters $\alpha=0.65$ and $\beta=0.35$. In Equation (8), the expected service time of jobs ($E(ST)$) is made up of the expected processing time of jobs ($E(PT)$) and the expected setup time of jobs ($E(S)$). Accordingly, the expected processing time of jobs ($E(PT)$) can easily be seen as Equation (20):

$$
\begin{aligned}
E\left(PT\right) &= \left(\sum_{j=1}^{J} n_j\right)^{-1} \sum_{j=1}^{J}\sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right] pt_j \\
&= E\left(ST\right) - E\left(S\right) \\
&= E\left(ST\right) - \left(\sum_{j=1}^{J} n_j\right)^{-1} \sum_{j=1}^{J}\sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right] P_{FIFO} \sum_{\substack{r=1 \\ r\ne j}}^{J} \frac{\lambda_r}{\lambda'} s_{jr}
\end{aligned}
\tag{20}
$$

where the expected service time of jobs ($E(ST)$) is given by $E(ST) = (\rho K)/\lambda$. Consequently, the processing time of product type j job (pt_j) can be computed and can be shown as Equation (21) and Equation (22):

$$
pt = \left[E\left(ST\right)\sum_{j=1}^{J} n_j - \sum_{j=1}^{J} P_{FIFO} \sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right] \sum_{\substack{r=1 \\ r\ne j}}^{J} \frac{\lambda_r}{\lambda} s_{jr} \right] \left[\sum_{j=1}^{J} v_j \sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right] \right]^{-1} \tag{21}
$$

$$
pt_j = \frac{v_j}{\sum_{j=1}^{J} v_j \sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right]} \left[E\left(ST\right)\sum_{j=1}^{J} n_j - \sum_{j=1}^{J} P_{FIFO} \sum_{i=1}^{n_j} \Pr\left[T_{ij} \le RT\right] \sum_{\substack{r=1 \\ r\ne j}}^{J} \frac{\lambda_r}{\lambda} s_{jr} \right] \tag{22}
$$

where $v_j = 1 - u_j$ and $pt_j = v_j pt$. Note "second" is the time unit for processing and setup time.

Once the procedure in Figure 5 has been completed, the following job information is obtained: arrival rate of jobs, processing time of jobs, and setup time of jobs. The following machine information is also obtained: expected machine utilization, number of machines, and run time. Thus, the estimated makespan defined in the previous section can be calculated. Regarding the simulation results, the makespans are also collected from running simulation models by modeling the independently and exponentially distributed inter-arrival time of jobs according to the control parameters (ρ, K, λ, and RT), to be dispatched or scheduled, respectively by FIFO,LPT, and PIAAP on the parallel machines. Only for the simulation model by dispatching jobs according to the machine schedule generated by PIAAP, experiment control parameters h, rescheduling criterion, should be included to make PIAAP regenerate the machines schedule to react to the impacts of dynamic arrival of jobs. However, to make the makespan reducing comparisons conduct in the same initial jobs condition, the simulation models under FIFO or LPT can not start the dispatching of jobs on machines until the number of job arriving in the buffer is equal to the rescheduling triggered threshold.

Analysis on reducing the makespan via FIFO, LPT, and PIAAP

To evaluate the performances in various approaches on reducing makespan, we compare the makespans which are respectively generated in the simulation models by dispatching jobs according to FIFO, LPT, or the schedule determined by PIAAP. The simulation results of the makespans for the dispatching rules with RT=21600 are shown in Figure 6. The makespan under PIAAP is always smaller than the makespans by FIFO and LPT. Thus, PIAAP is better than FIFO and LPT in decreasing the makespan, especially at high level of utilization. Moreover, the makespan by dispatching rules increases as rescheduling condition (h) becomes larger because the waiting time of jobs for reaching the rescheduling triggered threshold increases. To test whether there is significant differences between any two dispatching rules or scheduling algorithm, the statistical paired t-test is used. Let $C_{\max,FIFO}$, $C_{\max,LPT}$, and $C_{\max,PIAAP}$ be the makespans generated in simulation model under FIFO, LPT, and PIAAP.

Two tests are conducted to determine if PIAAP is effective to reduce makespan, which implies that difference in makespan between FIFO and PIAAP ($\delta_{FIFO\text{-}PIAAP}$) and difference in makespan between LPT and PIAAP ($\delta_{LPT\text{-}PIAAP}$) are greater than zero. Table 1 presents the results of the paired t-test. Considering the null hypothesis H_0: $\delta_{FIFO\text{-}PIAA} = 0$ and the alternative hypothesis H_1: $\delta_{FIFO\text{-}PIAA} > 0$, the average and the standard deviation of $d_{FIFO\text{-}PIAAP}$ are equal to 134.2982 and 222.3616. The value of the test statistic under H_0: $\delta_{FIFO\text{-}PIAA} = 0$ is shown as Equation (23):

$$
t = \frac{\bar{d}_{FIFO-PIAAD} - 0}{sd\left(d_{FIFO-PIAAD}\right)\big/\sqrt{n}} = \frac{134.2982 - 0}{222.3616\big/\sqrt{192}} = 8.3688 \tag{23}
$$

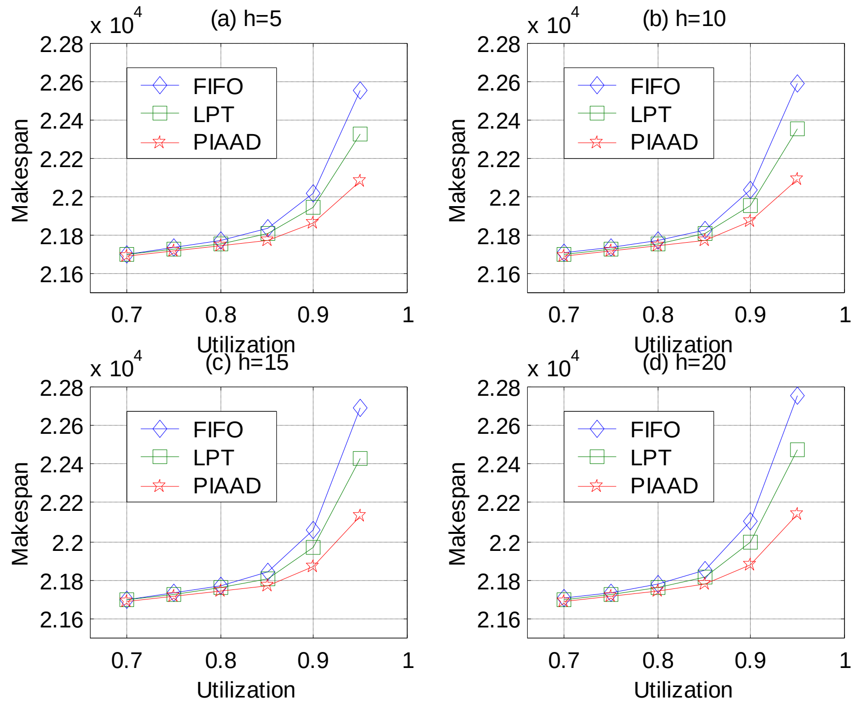

Figure 6. Makespans generated by the simulation models which the jobs are dispatched according to the FIFO rule, the LPT rule, and PIAAP when RT = 21600.

Table 1. Results of the paired t-test.

Null hypothesis	Alterative hypothesis	Sample size	Sample mean	Standard deviation	Test statistic	P-value
H_0: $\delta_{FIFO\text{-}PIAA} = 0$	H_1: $\delta_{FIFO\text{-}PIAA} > 0$	192	134.2982	222.3616	8.3688	< 0.0001
$\boldsymbol{H_0}$: $\delta_{LPT\text{-}PIAA} = 0$	H_1: $\delta_{LPT\text{-}PIAA} > 0$	192	72.0679	135.1171	7.3906	< 0.0001

where $d_{FIFO\text{-}PIAAP} = C_{\max,FIFO} - C_{\max,PIAAP}$.

With degrees of freedom 191 and the obtained P-value lower than 0.0001. When the P-value is smaller than the significance level α = 0.025, the null hypothesis is rejected. Therefore, we conclude that the makespan generated by the simulation model with the job dispatching rule FIFO is larger than that generated by PIAAP. Consider the test of H_0: $\delta_{LPT\text{-}PIAA} = 0$ versus H_1: $\delta_{LPT\text{-}PIAA} > 0$, the P-value obtained is also smaller than 0.0001. Therefore, the makespan generated by the simulation model with the job dispatching rule LPT is larger than that generated by PIAAP. Therefore, PIAAP has a significant effect on reducing the makespan in comparing with the dispatching rules FIFO and LPT.

Analysis on reducing the setup time via FIFO, LPT, and PIAAP

To demonstrate the superiority of PIAAP in reducing set up time by comparing with the setup time determined by FIFO and LPT, we conduct the following analysis. Both Figures 7 and 8 show the average setup time per job under FIFO, LPT, and PIAAP with λ = 1 and λ = 3, respectively. In these two figures, for LPT and PIAAP excluding FIFO, the average setup time per job decreases as the expected machine utilization rate increases. PIAAP contributes to the reduction of setup time per job as the machine utilization rate (ρ) increasing, which indicates that more jobs would wait in the buffer and cause higher opportunities of inserting jobs without causing additional setup time. Furthermore, larger rescheduling condition (h, relating to the number of unscheduled jobs in the buffer waiting to be scheduled before rescheduled mechanism is activated) contributes to the setup time saving of jobs, which implies that PIAAP can not regenerate job schedules until h reaching a specific higher level and then causes a larger possibility of finding an inserted position with small additional setup time. However, even for the same level of utilization rate of machine, PIAAP does not lead to smaller average setup time per job for the more

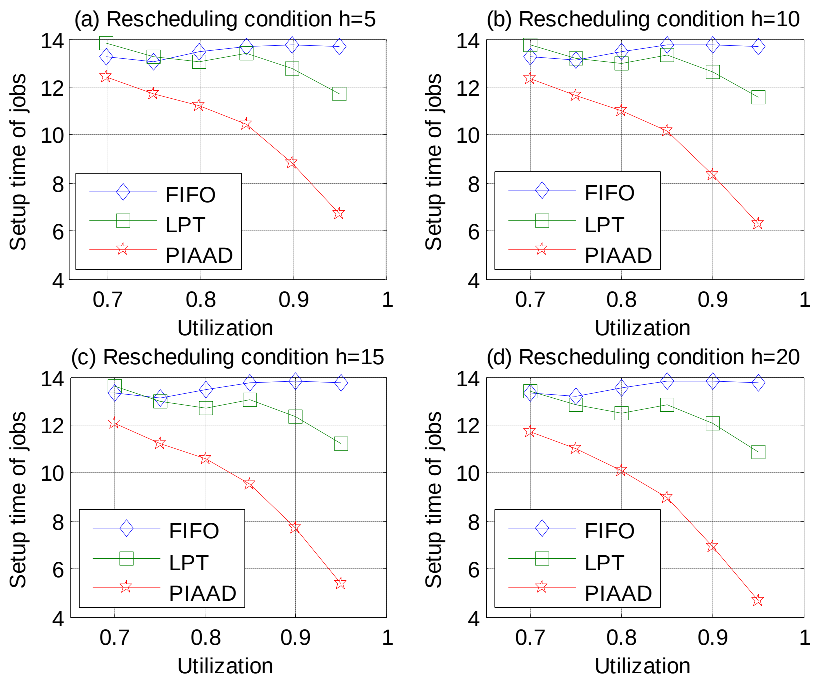

Figure 7. Setup time per job generated by the simulation model which the jobs are dispatched according to the FIFO rule, the LPT rule, and the PIAAP when λ = 1.

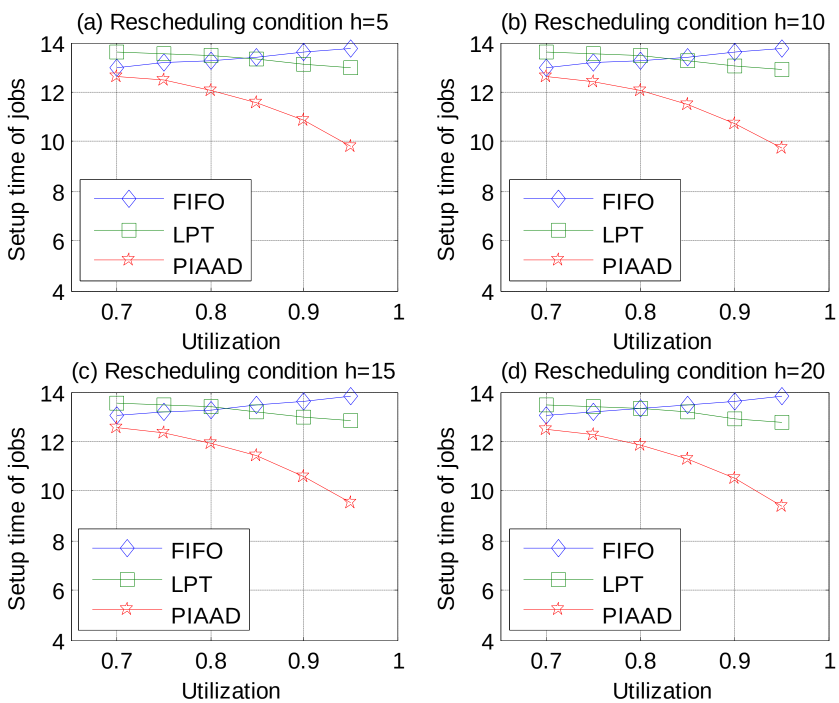

Figure 8. Setup time per job generated by the simulation model which the jobs are dispatched according to the FIFO rule, the LPT rule, and the PIAAP when λ = 3.

arriving jobs condition λ = 3 compared with λ = 1. Let IP_1 be the improved percentage indicating the difference between the simulated setup times by FIFO and PIAAP and IP_2 be the improved percentage indicating

Table 2. Average improved percentage of simulated setup time for various expected machine utilization rates, rescheduling conditions, and total arrival rates.

Total arrival rate	h	Improved percentage	Expected machine utilization rate					
			0.70	0.75	0.80	0.85	0.90	0.95
1	5	IP_1	5.9906	10.1648	16.2676	23.9217	35.9010	50.7113
		IP_2	10.1972	11.5900	13.9548	22.1583	30.7455	42.4829
	10	IP_1	6.8787	11.0524	18.1607	25.9759	39.7288	54.2472
		IP_2	10.5593	11.7797	14.9286	23.6126	34.1157	45.7847
	15	IP_1	9.3594	14.1541	21.6292	30.5316	44.0877	60.6554
		IP_2	11.5784	13.4848	16.8994	26.9230	37.4326	51.7410
	20	IP_1	12.0360	16.3990	25.6165	35.1232	49.9960	65.9528
		IP_2	12.4295	14.1310	19.0648	30.0198	42.6389	56.8310
3	5	IP_1	2.6805	5.2762	8.7881	13.7058	19.7960	28.7786
		IP_2	7.1353	7.8376	10.1159	13.0051	16.5940	24.1187
	10	IP_1	2.9142	5.6733	9.1986	14.1217	21.0265	29.2475
		IP_2	7.1444	8.0710	10.2559	13.2076	17.6669	24.4162
	15	IP_1	3.6373	6.5768	10.1137	15.1594	22.2240	30.7721
		IP_2	7.3875	8.4219	10.6628	13.6834	18.4784	25.4705
	20	IP_1	4.2411	7.0866	10.7322	15.8576	22.9547	22.9547
		IP_2	7.5184	8.3914	10.6668	13.9836	18.6117	18.6117

the difference between the simulated setup times by LPT and PIAAP. IP_1 and IP_2 are defined as Equations (24) and (25):

$$IP_1 = \frac{\text{SetupTime}_{FIFO} - \text{SetupTime}_{PIAAD}}{\text{SetupTime}_{FIFO}} \times 100\% \quad (24)$$

$$IP_2 = \frac{\text{SetupTime}_{LPT} - \text{SetupTime}_{PIAAD}}{\text{SetupTime}_{LPT}} \times 100\% \quad (25)$$

Table 2 displays the average value of IP_1 and IP_2 for various expected machine utilization rates, rescheduling conditions, and total arrival rates. For the total arrival rate (λ) equal to 1, the reduction of setup time by PIAAP increases as the expected machine utilization rate increases, in which IP_1 ranges from 5.9906 to 65.9528%, and IP_2 ranges from 10.1972 to 56.8310%. When the total arrival rate (λ) equals to 3, the range of IP_1 increases from 2.6805 to 30.7721%, and IP_2 ranges from, 7.1353, to 25.4705%.

Overall, IP_1 and IP_2 increase as the expected machine utilization rate increases. PIAAP is better than FIFO and LPT in decreasing the setup time of jobs, especially at high level of machine utilization rate, which also explains why PIAAP has superiority over FIFO and LPT in minimizing makespan.

Analysis on the effect of rescheduling criterion

To analyze the effect of the rescheduling policy to reduce the impacts of dynamic arrival of jobs, let TS_5, TS_{10}, TS_{15}, and TS_{20} respectively represent the total setup time determined by the simulation models, in which jobs are dispatched according to the schedules generated by the PIAAP and the rescheduling criteria (h) are set to 5, 10, 15, and 20 correspondingly. In Figure 9, the label "5-10" indicates the total setup time saving (difference) produced for the increment of rescheduling criterion from 5 to 10, which is calculated with the expression TS_5 to TS_{10}. Similarly, "10 to 15" and "15 to 20" both indicate the total setup time savings (differences) determined by TS_{10} to TS_{15} and TS_{15} to TS_{20}. The solid line in Figure 9 represents the total arrival rate (λ) equal to 1, and the dotted line in Figure 9 represents the total arrival rate (λ) equal to 3. Overall, the increase in the rescheduling condition from 10 to 15 leads to the largest savings of the total setup time. This shows that the rescheduling criterion to be executed along with the PIAAP can be set to a value in the interval (10, 15) to achieve the most benefits of setup time saving.

Accuracy analysis of estimating makespan by the probabilistic model

To analyze the accuracy of the probabilistic model on estimating makespan, the error percentage of estimated makespan is defined by Equation (26):

$$E_{makespan} = \frac{|\text{Makespan\_E} - \text{Makespan\_S}|}{\text{Makespan\_S}} \times 100\% \quad (26)$$

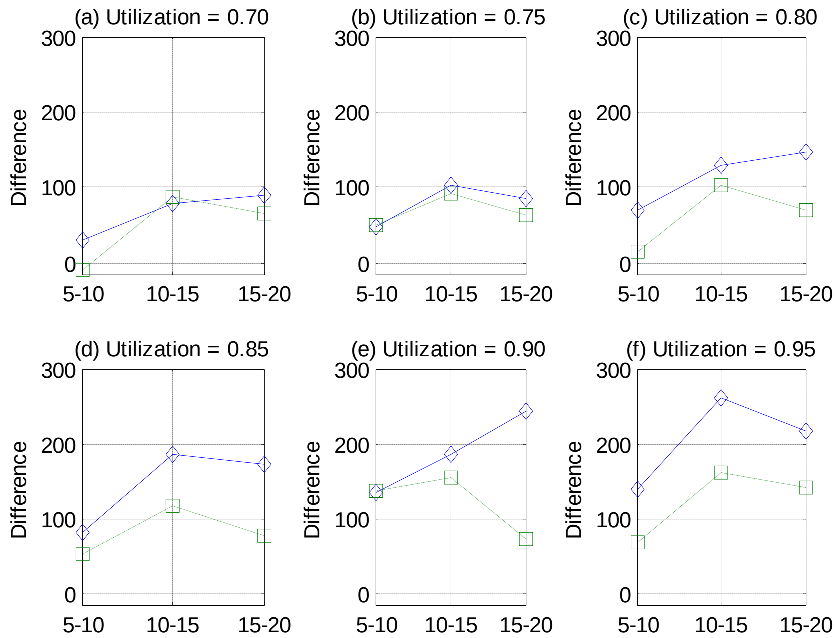

Figure 9. Difference of the total setup time for two rescheduling conditions in the simulation models with the jobs dispatched according to the PIAAP.

Table 3. Error percentage of estimated makespan for two levels of run times with a specific expected machine utilization rate.

Run time	Utilization	0.70	0.75	0.80	0.85	0.90	0.95
	Makespan _S	14509.0307	14533.8355	14599.6973	14661.6875	14853.9324	15201.6698
14400	Makespan _E	14503.2500	14547.0000	14616.2500	14714.7500	14932.5000	15543.0000
	$E_{makespan\ (\%)}$	0.0398	0.0906	0.1134	0.3619	0.5289	2.2453
	Makespan _S	21706.9494	21731.9236	21774.8093	21833.6962	22021.0928	22558.8004
21600	Makespan _E	21703.5000	21746.2500	21817.0000	21917.5000	22129.7500	22770.0000
	$E_{makespan\ (\%)}$	0.0159	0.0659	0.1938	0.3838	0.4934	0.9362

where Makespan_E and Makespan_S are the average makespan estimated by the probabilistic model and simulation model, respectively. The values of Makespan_E, Makespan_S, and $E_{makespan}$ are shown in Table 3. When the utilization rate of machine is increasing, the error percentage of estimated makespan also increases correspondingly. The highest $E_{makespan}$ is 2.2453% at the machine utilization rate of 0.95 for $RT = 14400$, the lowest $E_{makespan}$ is 0.0159% at the machine utilization rate of 0.70 for $RT = 21600$. The overall mean of $E_{makespan}$ is 0.4557%.

Generally, the probabilistic model can accurately estimate the makespan. Since probabilistic model is under FIFO, which is a common and simple dispatching rule, the estimated makespan can be regarded as a lower standard in performance comparison, which can be used to evaluate the performances of proposed PIAAP efficiently.

Accuracy analysis of estimating setup time by the probabilistic model

To analyze the accuracy of the probabilistic model on estimating setup time, the error percentage of estimated setup time is defined by Equation (27):

$$E_{setup\ time} = \frac{|\text{SetupTime\_E} - \text{SetupTime\_S}|}{\text{SetupTime\_S}} \times 100\% \qquad (27)$$

where SetupTime_E and SetupTime_S are the average setup time estimated by the probabilistic model and simulation model, respectively. Table 4 shows the values

Table 4. Error percentage of estimated setup time for various utilization rates of machine.

Utilization	0.70	0.75	0.80	0.85	0.90	0.95
SetupTime_E	13.113	13.393	13.644	13.653	13.785	14.134
SetupTime_S	13.124	13.130	13.358	13.572	13.674	13.731
$E_{setup\ time}$ (%)	0.0838	2.0030	2.1410	0.5968	0.8118	2.9350

of SetupTime_E, SetupTime_S, and $E_{setup\ time}$ for various utilization rates of machine. While the utilization rate of machine increases, SetupTime_E and SetupTime_S display the increasing trend and the error percentage of estimated setup time ranges from 0.0838 to 2.9350%. The overall mean of $E_{setup\ time}$ is 1.4286%. In the foregoing discussion, the setup time can be accurately estimated by this probabilistic model.

Conclusions

This paper provided PIAAP to solve the identical parallel-machine problem in the dynamic manufacturing environment with minimizing the makespan. The event-driven rescheduling policy is adopted to counter the impacts of job arriving dynamically. To evaluate the performance of reactive PIAAP, three simulation models with FIFO, LPT, and PIAAP executed along with a rescheduling criterion are built. The computational results show that PIAAP has an advantage over FIFO and LPT on reducing makespan due to the ability of PIAAP in the setup time saving of jobs, especially at high utilization rate of machine. Moreover, the statistical paired *t*-tests show that PIAAP significantly reduces makespan comparison with dispatching rules FIFO and LPT. Regarding the effect of rescheduling criterion (*h*), the *h* that increases from 10 to 15 has stronger effect on reducing the total setup time in various rescheduling criteria. Furthermore, the probabilistic model developed to estimate makespan or the setup time for identical parallel-machine problem with dynamic job arrivals have been shown to be effective. Generally, the probabilistic model under FIFO can accurately estimate the makespan and the setup time.

Since the probabilistic model is developed under FIFO which is a simple and widely usage rule, the estimated makespan generated by the probabilistic model can be regarded as a lower level of standard in performance comparison, which can be used to evaluate the superiority of the proposed scheduling algorithm. The computational test has proved the probabilistic model’s capability in accurately estimating makespan. Therefore, the superiority of the proposed scheduling algorithm, such as reactive PIAAP, on reducing the makespan of parallel machine problem with dynamic job arrivals can be evaluated efficiently by the developed probabilistic model.

REFERENCES

Bernstein D, Pinter RY, Rodeh M (1985). Optimal scheduling of arithmetic operations in parallel with memory. Conference Record of Annual ACM Symposium on Principles of Programming Langiages, New York.

Chakravarty AK, Balakrishnan N (1996). Job sequencing rules for minimizing the expected makespan in flexible machines. Eur. J. Oper. Res. 96(2):274-288.

Cheng TCE, Sin CCS (1990). A state-of-the-art review of parallel-machine scheduling research. Eur. J. Oper. Res. 47(3):271-292.

Garey MR, Johnson DS (1979). Computer and intractability: A guide to the theory of NP-completeness, W H Freeman, San-Francisco.

Graham RL (1969). Bounds on multiprocessing timing anomalies. SIAM. J. Appl. Math. 17: 263-269.

Lee YH, Pinedo M (1997). Scheduling jobs on parallel machines with sequence-dependent setup times. Eur. J. Oper. Res. 100(3):464-474.

Luh PB, Hoitomt DJ, Max E (1988). Parallel machine scheduling using lagrangian relaxation. IEEE International Conference on Computer Integrated Manufacturing.

Nokaki S, Ross S (1978). Approximations in finite capacity multi-server queues with poisson arrivals. J. Appl. Prob. 15(4):826-83.

Pearn WL, Chung SH, Yang MH, Shiao KP (2008). Solution Strategies for Multi-Stage Wafer Probing Scheduling Problem with Reentry. J. Oper. Res. Soc. 59(5): 637-651.

Potts CN (1983). Analysis of a linear programming heuristic for scheduling unrelated parallel machines. Discrete Appl. Math.10(2): 155-164.

Potvin Y, Rousseau JM (1993). A Parallel route building algorithm for the vehicle routing and scheduling problem with time windows. Eur. J. Oper. Res. 66(3):19-26.

Raaymakers WHM, Bertrand JWM, Fransoo JC (2001). Makespan estimation in batch process industries using aggregate resource and job set characteristics. Int. J. Prod. Econ. 70(2):145-161.

Raaymakers WHM, Weijters AJMM (2003). Makespan estimation in batch process industries: A comparison between regression analysis and neural networks. Eur. J. Oper. Res. 145(1):14-30.

Sethi R (1977). On the complexity of mean flow time scheduling. Math. Oper. Res. 2(4):320-330.

Vieira GME, Herrmann JW, Lin E (2000a). Analytical models to predict the performance of a single machine system under periodic and event-driven rescheduling strategies. Int. J. Prod. Res. 38(8): 1899-1915.

Vieira GME, Herrmann JW, Lin E (2000b). Predicting the performance of rescheduling strategies for parallel machine systems. J. Manuf. Sys. 19(4):256-266.

Vieira GME, Herrmann JW, Lin E (2003), Rescheduling manufacturing systems: A framework of strategies, policies, and methods. J. Scheduling. 6(1):39-62.