Taxonomy of batch queueing models in manufacturing systems

Kan Wu

School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Ave., Singapore 639798, Singapore

A R T I C L E   I N F O

Article history:
Received 6 November 2012
Accepted 3 February 2014
Available online 10 February 2014

Keywords:
Productivity and competitiveness
Queueing
Manufacturing
Batching

A B S T R A C T

Batching plays an important role in performance evaluation of manufacturing systems. Three types of batching are commonly seen: transfer batches, parallel batches and serial batches. To model the batching behavior correctly, a comprehensive classification of batching is proposed. Eight types of batching behavior are classified and corresponding queueing models are given. The newly proposed models are validated by simulation.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Batching is commonly seen in manufacturing systems, such as pharmaceutical plants or semiconductor wafer fabrication facilities (fabs). Modeling batching behavior correctly is crucial to the performance evaluation, and thus to the productivity and competitiveness of a manufacturing system. There are mainly two types of batches, process batches and transfer batches. A process batch can be further classified as a serial batch or a parallel batch. A process batch is defined as a pre-determined group of jobs processed in a certain period of time without being interrupted by other product groups. The job group has no specific composition and can be composed of a single product or multiple products, as long as the jobs use the same recipe (i.e., a parallel batch) or do not induce setup (i.e., a serial batch).

We name a process batch as serial if products are processed one by one consecutively without being interrupted by other product groups. According to Hopp and Spearman (1996), the serial batch size is “the number of jobs of a common family processed before the workstation is changed over to another family.” A serial batch can be composed of a single product or multiple products, as long as the products do not induce setups. A setup induced by a serial batch is called a product-induced setup. It generally occurs when jobs with different recipes are loaded on a machine, and some preparations are needed to set up the recipe. Larger serial batch size reduces setup frequency, but is potentially disadvantageous considering job due dates, since urgent jobs with different recipes may be processed after normal jobs with the same recipe as the current processing jobs.

We name a process batch as parallel if all jobs are processed together at the same time. Furnaces and ovens are typical examples of parallel batches. The parallel batch size is usually determined by the physical capacity of a machine, but it could also be determined by the process constraints. For example, the common physical capacity of a furnace in a 300mm semiconductor fab is four lots. However, the parallel batch size of a furnace is sometimes reduced to three due to process quality concerns. A parallel batch can be composed of a single product or multiple products, as long as the products use the same recipe at the batching machine.

In literature, the parallel batch size sometimes is denoted by the pair \((a, b)\), where \(a\) is the minimal batch size and \(b\) is the maximum batch size. A machine cannot process a batch unless there are at least \(a\) jobs in one batch, and the batch cannot contain more than \(b\) jobs. However, the batching rules can be even more complex in practice. For example, a machine can process a batch when there are at least \(a\) jobs in the batch, and the next job will not arrive within \(x\) minutes. Otherwise, the machine has to wait until \(b\) jobs are batched. Since it involves non-renewal arrival processes, the exact analysis is intricate. To serve the purpose of classification, in this paper we simply focus on the cases where \(a\) equals \(b\).

In addition to process batches, transfer batches are also commonly seen in production lines. The transfer batch size is “the number of parts that accumulate before being transferred to the next station” (Hopp & Spearman, 1996). In general, transfer batches are caused by mismatch between transfer units and process units. Therefore, the definition of transfer batch is highly dependent on the process unit of destination machine. If the process of a machine is lot-based, the transfer batch size is the number

---

* Tel.: +65 67905584.
  E-mail address: wukan@ntu.edu.sg

http://dx.doi.org/10.1016/j.ejor.2014.02.004
0377-2217/© 2014 Elsevier B.V. All rights reserved.
of lots. If the process of a machine is wafer-based, the transfer batch size is the number of wafers. For example, if each lot consists of 24 wafers, when lots arrive one by one at a wafer-based chemical vapor deposition tool, the transfer batch size is 24 (wafers). However, since the process of a wet bench is lot-based, if lots arrive one by one, here the transfer batch size is one for a wet bench. Since the transfer unit in a semiconductor fab is “lot” in general, all wafer-based process machines see transfer batches with wafer-based sizes. The relationship among different types of batching is illustrated in Fig. 1.

Because batching process is a potential cause of long queue times, modeling batch processing correctly is critical in evaluating the performance of manufacturing systems. However, since there are parallel, serial and transfer batches, the analysis can be complicated. Different combinations of the batching behavior can lead to different queueing models. To approximate system performance accurately, choosing the right model should be the first step before proceeding to further analyses. A comprehensive taxonomy of batching is crucial. We hope this paper can be treated as a guidance proceeding to further analyses. A comprehensive taxonomy of batching models is needed. To evaluate the performance of parallel batches in manufacturing systems. By studying the dependence between wait-to-batch time and queue time in the previous approaches, Wu, McGinnis, & Zwart, 2011a introduced a more accurate parallel batch model based on the M/M/c queue.

For batch arrivals with parallel batch processing, Chiamsiri and Leonard (1981) provided the most general single-server model, $G^N/G/1$ through diffusion approximation, where arrivals occur in bulks of random size X and the parallel batch is of the random size Y. However, because Chiamsiri and Leonard’s model allows a job to join a processing batch up to the parallel batch size without affecting the service time, the model may not capture a real situation in practical manufacturing systems. Zisgen (2009) fixed the flaw and gave the diffusion approximation for the $G^N/G/1$ queue, where the parallel batch size is a fixed constant k. Hanschke (2006) provided an approximate model for $G^N/G/c$ queue based on the independence assumption used by Bitran and Tirupati (1989).

Hopp and Spearman (1996) introduced models for parallel process batches, serial process batches and transfer batches. However, their serial process batch models are limited to processes with batch arrivals where the transfer batch size is the same as the serial batch size. This situation is not very general in practice. In order to have better understanding of different types of batching, a comprehensive classification of batching models is needed.

2. Literature review

Batch processing has captured researchers’ attention for decades and been rigorously studied. The study of this topic may be traced back to Bailey (1954), who modeled a simple queueing process in which customers arrive at a single queue at random, and are served in a batch with a fixed maximum batch size.

Chaudhry and Templeton (1983) summarized the state of the art till then in their book, A First Course in Bulk Queues. Two types of batch processing were addressed: bulk-arrival queues and bulk-service queues: bulk arrivals correspond to transfer batches and bulk services correspond to parallel process batches. Because they focused on the queueing models which can be solved exactly, they did not address $G/G/1$ based approximate models. Furthermore, models of serial process batches were not discussed.

Serial process batches can be modeled by polling models with some strong assumptions. A polling model is a system of multiple queues accessed by a single server based on a prescribe manner. An extensive study on polling model was carried out to model computer communication networks in 1970s and local-area networks in 1980s. Under the assumption of cyclic order and Poisson arrivals, Takagi (1988, 2000) summarized the applications of polling models under four dispatching rules: exhaustive, gated, limited and decrementing service. The cyclic order can also be replaced by a periodic service order table, or a polling table (Baker & Rubin, 1987; Eisenberg, 1972). However, both the assumptions of cyclic order or polling tables are not general in practice, since the practical situations are far more complicated and have to consider the job due dates and current work-in-progress distributions on the shop floor.

In the late 1980s, the $G/G/1$ based approximate models for parallel process batches were proposed by Bitran and Tirupati (1989) and Segal and Whitt (1989). Their approximations decomposed cycle times into three parts: wait-to-batch time, queue time and service time. This approximate model is commonly applied to evaluate the performance of parallel batches in manufacturing systems. By studying the dependence between wait-to-batch time and queue time in the previous approaches, Wu, McGinnis, & Zwart, 2011a introduced a more accurate parallel batch model based on the $M/M/c$ queue.

For batch arrivals with parallel batch processing, Chiamsiri and Leonard (1981) provided the most general single-server model, $G^N/G/1$ through diffusion approximation, where arrivals occur in bulks of random size X and the parallel batch is of the random size Y. However, because Chiamsiri and Leonard’s model allows a job to join a processing batch up to the parallel batch size without affecting the service time, the model may not capture a real situation in practical manufacturing systems. Zisgen (2009) fixed the flaw and gave the diffusion approximation for the $G^N/G/1$ queue, where the parallel batch size is a fixed constant k. Hanschke (2006) provided an approximate model for $G^N/G/c$ queue based on the independence assumption used by Bitran and Tirupati (1989).

Hopp and Spearman (1996) introduced models for parallel process batches, serial process batches and transfer batches. However, their serial process batch models are limited to processes with batch arrivals where the transfer batch size is the same as the serial batch size. This situation is not very general in practice. In order to have better understanding of different types of batching, a comprehensive classification of batching models is needed.

3. Classification of batching

There are two types of batching: process and transfer batches. Process batches can be further classified as parallel and serial batches. Depending on the property of current and upstream machines, process batch and transfer batch can occur at the same machine. Therefore, we have eight different combinations.

In Table 1, the leftmost column is the transfer batch size. It can be no batch at all (i.e., single job) or arrive as a batch. The topmost row is the process batch types. It can be no batch (i.e., single job), parallel batch, serial batch or both serial and parallel batch. If jobs are transferred as single jobs, the classification is self-explanatory. If jobs are transferred as batches, we assume the transferred batch size is the same as the parallel or serial process batch size in Section 4. However, this assumption is indeed not as strict as it appears, and can be generalized as follows: If the transferred batch size is different from the parallel batch size, we can use model B-1 if the transfer batch size is a multiple of the parallel batch size, or use model S-2 if the parallel batch size is a multiple of the transfer batch size. If the transferred batch size is smaller than the serial batch size, we may use model S-3 if the serial batch size is a multiple of the transfer batch size. If the transferred batch size is greater than a periodic service order table, or a polling table (Baker & Rubin, 1987; Eisenberg, 1972). However, both the assumptions of cyclic order or polling tables are not general in practice, since the practical situations are far more complicated and have to consider the job due dates and current work-in-progress distributions on the shop floor.

In the late 1980s, the $G/G/1$ based approximate models for parallel process batches were proposed by Bitran and Tirupati (1989) and Segal and Whitt (1989). Their approximations decomposed cycle times into three parts: wait-to-batch time, queue time and service time. This approximate model is commonly applied to evaluate the performance of parallel batches in manufacturing systems. By studying the dependence between wait-to-batch time and queue time in the previous approaches, Wu, McGinnis, & Zwart, 2011a introduced a more accurate parallel batch model based on the $M/M/c$ queue.

For batch arrivals with parallel batch processing, Chiamsiri and Leonard (1981) provided the most general single-server model, $G^N/G/1$ through diffusion approximation, where arrivals occur in bulks of random size X and the parallel batch is of the random size Y. However, because Chiamsiri and Leonard’s model allows a job to join a processing batch up to the parallel batch size without affecting the service time, the model may not capture a real situation in practical manufacturing systems. Zisgen (2009) fixed the flaw and gave the diffusion approximation for the $G^N/G/1$ queue, where the parallel batch size is a fixed constant k. Hanschke (2006) provided an approximate model for $G^N/G/c$ queue based on the independence assumption used by Bitran and Tirupati (1989).

Hopp and Spearman (1996) introduced models for parallel process batches, serial process batches and transfer batches. However, their serial process batch models are limited to processes with batch arrivals where the transfer batch size is the same as the serial batch size. This situation is not very general in practice. In order to have better understanding of different types of batching, a comprehensive classification of batching models is needed.

Table 1

<table>
<thead>
<tr>
<th>Transfer</th>
<th>Process</th>
<th>Single job</th>
<th>Parallel batch</th>
<th>Serial batch</th>
<th>Serial &amp; parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch</td>
<td>B-1</td>
<td>B-2</td>
<td>B-3</td>
<td>B-4</td>
<td></td>
</tr>
<tr>
<td>S-1</td>
<td>S-2</td>
<td>S-3</td>
<td>S-4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1. Types of batching.
than the serial batch size, in order to reduce the setup frequency, we may simply enlarge the serial batch size to meet the transfer batch size in practice. The model for more general situations will be discussed in Section 6.

In a semiconductor fab, batch arrivals occur when the process of a machine is wafer-based, but the job arrivals are lot-based, or even if the machine process is lot-based, but the upstream machine is a parallel batch machine (e.g. furnace). Hence, jobs arrive according to the batch size of the upstream machines.

The departure transfer batch size is assumed to be the same as the arrival transfer batch size except for B-1, B-3, S-2 and S-4. For S-2 and S-4, jobs arrive individually but depart as a batch. For B-1 and B-3, jobs arrive as a batch, but the departure batch size can be either one or the same as the arrival transfer batch size depending on real situations.

When jobs are processed in a parallel batch (e.g. a furnace), all jobs will be completed and depart at the same time as a batch. Therefore, we do not consider the single departure case for B-2 and S-2. For B-2 and B-4, jobs arrive as a batch and depart as a batch.

In B-1, if jobs arrive as a batch (of multiple lots) at a lot-based process machine, they can depart as a single lot after the process is complete. On the other hand, if a job arrives as a lot (of multiple wafers) at a wafer-based process machine, it can only depart after the entire lot is completed. Therefore, the departure batch size can be either one (i.e., the former case) or the same as the arrival transfer batch size (i.e., the latter case). The justification of B-3 is the same except that there is a product-induced setup associated with the process.

For S-1 and S-3, jobs arrive and depart individually. For S-3, different products may induce product-induced setups.

4. The model

When the transfer batch size is the same as the parallel batch size, queueing models for B-2 are indeed the same as the standard non-batching queueing models of S-1. B-4 is the same as S-3. Hence, we will only focus on the approximate models of the other five cases.

4.1. Single job arrivals with parallel batch processing (S-2)

In S-2 models, the inter-arrival times between consecutive jobs are independent and identically distributed (i.i.d.) and jobs are processed in parallel batches with fixed batch size. This type of problem is a G/G/1 queue. Traditionally, it is approximated by the G/G/1 based approximate models (Bitran & Tirupati, 1989). The model decomposes the cycle time into three parts: wait-to-batch time (WTBT), queue time (QT) and service time (ST). The structure is illustrated in Fig. 2.

Based on this scheme, Hopp and Spearman (1996) proposed the following model to approximate the mean cycle time (CT) of parallel process batches,

\[
CT = \frac{k - 1}{2\lambda} + \left(\frac{c_b^2 k + c_s^2}{2}\right) \left(\frac{\rho}{1 - \rho}\right) \frac{1}{\mu} + \frac{1}{\mu},
\]

where \( \rho = \lambda / (k\mu) \), and \( k \) is the fixed parallel batch size. \( \lambda \) is the arrival rate of jobs (jobs/hour), \( \mu \) is the service rate of the batching machine (batches/hour), \( c_b \) is the coefficient of variation (CV) of inter-arrival times, \( c_s \) is the CV of batch service times.

Eq. (1) attempts to decompose the cycle time into three segments. To guarantee the success of this decomposition, each segment needs to behave independently. However, this approach inevitably induces dependence between batch queue time and WTBT. Hence, the cycle time is over-estimated. Wu et al. (2011a) proposed the following new approach to approximate the performance of the parallel batch processing with better accuracy. The approximation is as follows:

Step 1. Compute the mean cycle time of M/M/1 queue. Let

\[
x \equiv \frac{1}{k + 1} (1 - \rho) - \frac{4}{3} \left(\frac{k - 1}{k + 1}\right)^2 - \rho^2
\]

\[
CT(M/\mu^1/1) = \frac{1}{2\lambda} \left(\frac{k - 1}{2}\right) \left(\frac{x^2 (1 - x^{-1})}{1 - x} - \frac{(k - 1) x^{-1} - j k}{x} + j \right)
\]

Step 2. Compute the mean queue time of M/M/1 queue.

\[
QT(M/M/1) = \frac{2}{1/k + 1} \left(CT(M/\mu^1/1) - \frac{1}{\mu} - k - \frac{1}{2\lambda}\right).
\]

Step 3. Approximate the mean queue time of G/G/1 queue.

\[
QT \cong \left(\frac{c_b^2 k + c_s^2}{2}\right)QT(M/M/1).
\]

Step 4. Compute the mean cycle time of G/G/1 queue.

\[
CT \cong \frac{k - 1}{2\lambda} + QT + \frac{1}{\mu}.
\]

The definitions of all parameters in Eqs. (2)–(6) are the same as the parameters defined in Eq. (1). Although Eq. (1) approximates batch cycle time with certain accuracy, Eq. (6) gives better results without asking for additional information, especially when \( c_b \) and \( c_s \) are small. Please see Wu et al. (2011a) for the detailed discussion on Eq. (6). Eq. (6) can be extended to multiple servers, i.e., G/G/c queues, by using the exact results from the M/M/c queue (Chaudhry & Templeton, 1983). The procedure is given by Wu et al. (2011a).

Although Eq. (6) may give more reliable results in practical manufacturing systems, its form is more complex than Eq. (1). When considering the batching behavior in an optimization model of an entire production line, in order to solve the model efficiently, keeping the form simple is preferable. The advantage of Eq. (1) is that it can be easily extended to multiple server systems by using the closed-form results from Sakasegawa (1977),

\[
QT(M/M/m) \cong \left(\frac{\rho^{2(m-1)} - 1}{m(1 - \rho)}\right) \frac{1}{\mu},
\]

where \( m \) is the number of servers of the workstation, \( \mu \) is the service rate of each server, and \( \rho = \lambda / (m\mu) \). Therefore, for a G/G/m queue, Eq. (1) becomes

\[
CT \cong \frac{k - 1}{2\lambda} + \left(\frac{c_b^2 k + c_s^2}{2}\right) \left(\frac{\rho^{2(m-1)} - 1}{m(1 - \rho)}\right) \frac{1}{\mu} + \frac{1}{\mu}.
\]

where \( \lambda \) is the arrival rate of jobs (jobs/hour), \( \mu \) is the service rate of each batching machine (batches/hour), \( \rho = \lambda / (m\mu) \), \( c_b \) is CV of inter-arrival times, and \( c_s \) is the CV of batch service times.

For inspection tools, the queue time under sampling plans can also be modeled by S-2 models with slight modification. When the sampling rate is 1/k, an inspection tool inspects the first job of every \( k \) consecutive arrivals.

In this situation, we can treat \( k \) as the parallel batch size, and a batch is formed right after the first job arrives. The rest of the jobs simply pass the tool without any delay and depart from the system.
at the arrival epoch. Since there is no wait-to-batch time, the mean queue time can be approximated as follows:

\[ QT \approx \left( \frac{c_k^2/k + c_q^2}{2} \right) \left( \frac{\rho}{1 - \rho} \right) \frac{1}{k \mu} \]  

(9)

where \( \mu \) is the rate of the inspection tool (e.g. units/hour), \( \rho = \lambda/k \mu \), \( c_k \) is the CV of inter-arrival times, and \( c_q \) is the CV of inspection tool service times. Note that the mean queue time in Eq. (9) is k times less than that in Eq. (1), since the queue time only exists for the first job of every batch. This model is validated by simulations in Section 5.

4.2. Single job arrivals with serial batch processing (S-3)

The model for the single job arrivals with serial batch processing has been proposed by Hopp and Spearman (1996), where it is classified as a non-preemptive model. Since there are various non-preemptive events with different queueing models (Wu et al., 2011b), their classification is too rough for practical use. But their non-preemptive model happens to be the model of S-3. In S-3 models, the inter-arrival times between consecutive jobs are i.i.d., and jobs are processed individually. This type of batching is also called a product-induced setup, which is one kind of run-based non-preemptive interruptions (Wu, 2014). Product-induced setups occur due to changes in production processes induced by switching products. It could occur because the other product has a more urgent due date, or the downstream stations are starving for the other products, etc.

Since a product-induced setup can be induced by various independent factors, it is reasonable to assume that the occurrence of a product-induced setup is memoryless and follows geometric distributions. Hence, the machine processes an average of \( N_p \) jobs between two consecutive setups (i.e., \( N_p \) is the serial batch size), and the probability of doing a setup after any job is equal (i.e., 1/Np). Let the mean product-induced setup time and its standard deviation be \( t_0 \) and \( \sigma_0 \) respectively, the mean service time and its standard deviation be \( t_e \) and \( \sigma_e \) respectively. Denote the service time of a job by \( t_e \) and the product-induced setup time by \( t_0 \). The squared coefficient of variation (SCV) of effective process times is

\[ c_e^2 = \left( \sigma_e^2 + \frac{\sigma_0^2}{N_p} + \frac{N_p - 1}{N_p^2} t_0^2 \right) / \sigma_e^2. \]  

(12)

The squared coefficient of variation (SCV) of effective process times is

\[ c_e^2 = \left( \sigma_e^2 + \frac{\sigma_0^2}{N_p} + \frac{N_p - 1}{N_p^2} t_0^2 \right) / \sigma_e^2. \]  

(12)

By the heavy traffic approximation (Heyman 1975), the mean cycle time of S-3 models can be estimated by Eq. (13), which is known as Kingman’s approximation,

\[ CT \approx \left( \frac{c_k^2 + c_q^2}{2} \right) \left( \frac{\rho}{1 - \rho} \right) t_e + t_0. \]  

(13)

where \( \rho = \lambda t_e \) and \( c_k \) and \( c_q \) is the coefficient of variation of arrival interval and effective process time, respectively.

4.3. Single job arrivals with parallel and serial batch processing (S-4)

In manufacturing systems, a furnace may have a setup between processing two different products. The serial and parallel batches can occur at the same machine. In S-4 models, the inter-arrival times between two consecutive jobs are i.i.d. and the arrival rate is \( \lambda \) (jobs/hour). Each parallel batch contains \( k \) jobs. The machine processes an average of \( N_p \) parallel batches between two consecutive setups, and the probability of doing a setup after any batch is \( 1/N_p \).

This model has both the properties of S-2 and S-3. In S-2, Eq. (1) is developed based on the decomposition assumption, which assumes the batch cycle time can be decomposed into three independent parts: WTBT, queue time and service time. In S-3, Eq. (13) is derived from the effective process time. Combining these two models, we have the following approximation of the mean cycle time of S-4 models:

\[ CT \approx \frac{k - 1}{2\lambda} + \left( \frac{c_k^2/k + c_q^2}{2} \right) \left( \frac{\rho}{1 - \rho} \right) t_e + t_0. \]  

(14)

where \( c_k \) is the CV of inter-arrival times (of jobs), \( c_q \) is the CV of effective batch service times, \( t_e \) is the mean effective batch process time, and \( \rho = \lambda \mu/k \).

\[ t_e = t_0 + t_p/N_p, \]  

(15)

\[ c_e^2 = \left( \sigma_e^2 + \frac{\sigma_0^2}{N_p} + \frac{N_p - 1}{N_p^2} t_0^2 \right) / t_e^2. \]  

(16)

where \( t_0 \) and \( \sigma_0 \) are the mean and standard deviation of product-induced setup times respectively, and \( t_e \) and \( \sigma_e \) are the mean and standard deviation of parallel batch service times respectively. Eq. (14) can be extended to multiple servers by Eq. (7).

Note that Eq. (14) is based on Eq. (1) and hence, succeeds its shortcomings as discussed in S-2. This model is validated by simulations in Section 5.

4.4. Batch arrivals with single job processing (B-1)

Batch arrivals with single job processing can occur at a machine when it is fed by a furnace. When either the inter-arrival time or service time is exponential, the model can be solved exactly (Chaudhry & Templeton, 1983). When both inter-arrival times and service times are generally distributed, we need to resort to approximations.

If the transfer batch size is a constant \( k \), each arrival batch can be viewed as a single job as shown in Fig. 3.

Jobs arrive as a batch with rate \( \lambda \), and the mean service time of a job is \( 1/\mu \). The mean service time of a batch is \( k/\mu \). For stability, we assume \( \rho = k \lambda > 1 \). The mean cycle time of each batch can be approximated by

\[ CT \approx \left( \frac{c_k^2 + c_q^2}{2} \right) \left( \frac{\rho}{1 - \rho} \right) k \frac{k}{\mu} / \mu. \]  

(17)

where \( c_k \) and \( c_q \) are the coefficient of variation of batch arrival intervals and single job service times, respectively.

Eq. (17) gives the mean cycle time of a job when jobs come and leave as batches. If each job departs right after its completion (without forming a batch again), the cycle time of a \( G^2/G/1 \) queue becomes

![Batch arrivals with single job processing (B-1)](image-url)
\[
CT \equiv \left( \frac{c_a^2 + c_s^2}{2} \right) \left( \frac{\rho}{1-\rho} \right) k \frac{k}{\mu} + \frac{k-1}{2\mu}
= \left( \frac{c_a^2 + c_s^2}{2} \right) \left( \frac{\rho}{1-\rho} \right) k + \frac{k+1}{2\mu}.
\]

Eq. (18) is a special case of the \( G/L/1 \) approximate model derived by Yao, Chaudhry, and Templeton (1984), where \( X \) is a random variable. Eq. (18) can be extended to multiple servers by Eq. (7).

4.5. Batch arrivals with serial batch processing (B-3)

In this model, we assume the transferred batch size is the same as the serial batch size. The justification of this assumption has been given in Section 3. This situation is commonly seen in semiconductor wafer fabs. When a lot arrives at a wafer-based process machine, the first wafer may incur an extra setup and the rest of wafers are processed consecutively without setup. The approximate model has been proposed by Hopp and Spearman (1996), where it is called a serial batch model. Jobs arrive as a batch with rate \( \lambda \) (batches/hour), and leave immediately after completion without forming a batch again. The mean service time is \( 1/\mu \) (hours/job). The mean cycle time can be approximated as

\[
CT \equiv \left( \frac{c_a^2 + c_s^2}{2} \right) \left( \frac{\rho}{1-\rho} \right) t_b + t_p + \frac{k+1}{2\mu}.
\]

(19)

\( t_b = \frac{k}{\mu} + t_p \).

(20)

\[ c_a^2 = \left( k\sigma_0^2 + \sigma_s^2 \right)/t_s^2. \quad (21) \]

\[ \rho = \lambda t_s. \quad (22) \]

where \( t_p \) is the mean setup time, \( k \) is the batch size, \( c_a \) is the CV of inter-arrival times (of batches), \( \sigma_0 \) is the standard deviation of service times (of a job) and \( \sigma_s \) is the standard deviation of setup times (of a batch). For stability, we assume \( \rho < 1 \).

If jobs leave as a batch with batch size \( k \), Eq. (19) becomes

\[
CT \equiv \left( \frac{c_a^2 + c_s^2}{2} \right) \left( \frac{\rho}{1-\rho} \right) t_b + t_p.
\]

(23)

Eq. (23) is the Kingman’s approximation, where one may think that a batch job arrives at a black box and leaves as a batch job after processed in the black box. Eqs. (19) and (23) can be extended to multiple servers by Eq. (7).

5. Simulation validation

In Section 4, two new models are proposed: inspection tools (modified S-2) and S-4. In this section, we will conduct simulation experiments to validate the models. Each mean queue time or cycle time is computed based on 30 replications of 250,000 jobs after discarding the first 500,000 jobs for warm-up.

5.1. S-2 model for inspection tools

The mean queue time approximation of inspection tools is given by Eq. (9). Its validation is demonstrated in three settings with different inter-arrival time and service time distributions. In each case, the parallel batch size \( k \) of 2, 5 and 10 are considered.

In the first case, the arrival processes are Poisson with ten different mean arrival rates resulting in utilizations from 10% to 95%. The service time is gamma distributed with a mean of 30 minutes and SCV 0.8. The results are shown in Table 2, where SQT is the queue time from simulation, 90% CI is the half-width of 90% confidence intervals of the corresponding sample mean and AQT is the approximate queue time computed based on Eq. (9). The percentage difference from AQT compared with SQT is reported in “Diff%” and Diff% = AQT/SQT - 1. The results show that the percentage difference (Diff%) is large at low utilization and

### Table 2

<table>
<thead>
<tr>
<th>Utilization</th>
<th>( k = 2 )</th>
<th>( k = 5 )</th>
<th>( k = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SQT</td>
<td>90%CI</td>
<td>AQT</td>
</tr>
<tr>
<td>10%</td>
<td>0.4</td>
<td>0.5%</td>
<td>1.1</td>
</tr>
<tr>
<td>20%</td>
<td>1.3</td>
<td>0.4%</td>
<td>2.4</td>
</tr>
<tr>
<td>30%</td>
<td>2.7</td>
<td>0.4%</td>
<td>4.2</td>
</tr>
<tr>
<td>40%</td>
<td>4.8</td>
<td>0.4%</td>
<td>6.5</td>
</tr>
<tr>
<td>50%</td>
<td>7.9</td>
<td>0.3%</td>
<td>9.8</td>
</tr>
<tr>
<td>60%</td>
<td>12.7</td>
<td>0.3%</td>
<td>14.6</td>
</tr>
<tr>
<td>70%</td>
<td>20.7</td>
<td>0.4%</td>
<td>22.8</td>
</tr>
<tr>
<td>80%</td>
<td>36.7</td>
<td>0.6%</td>
<td>39.0</td>
</tr>
<tr>
<td>90%</td>
<td>86.0</td>
<td>0.8%</td>
<td>87.8</td>
</tr>
<tr>
<td>95%</td>
<td>179.7</td>
<td>1.9%</td>
<td>185.3</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Utilization</th>
<th>( k = 2 )</th>
<th>( k = 5 )</th>
<th>( k = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SQT</td>
<td>90%CI</td>
<td>AQT</td>
</tr>
<tr>
<td>10%</td>
<td>1.2</td>
<td>0.3%</td>
<td>1.2</td>
</tr>
<tr>
<td>20%</td>
<td>2.6</td>
<td>0.2%</td>
<td>2.6</td>
</tr>
<tr>
<td>30%</td>
<td>4.5</td>
<td>0.2%</td>
<td>4.5</td>
</tr>
<tr>
<td>40%</td>
<td>7.0</td>
<td>0.2%</td>
<td>7.0</td>
</tr>
<tr>
<td>50%</td>
<td>10.5</td>
<td>0.2%</td>
<td>10.5</td>
</tr>
<tr>
<td>60%</td>
<td>15.8</td>
<td>0.3%</td>
<td>15.8</td>
</tr>
<tr>
<td>70%</td>
<td>24.5</td>
<td>0.5%</td>
<td>24.5</td>
</tr>
<tr>
<td>80%</td>
<td>42.2</td>
<td>0.5%</td>
<td>42.0</td>
</tr>
<tr>
<td>90%</td>
<td>95.3</td>
<td>1.3%</td>
<td>94.5</td>
</tr>
<tr>
<td>95%</td>
<td>199.3</td>
<td>1.9%</td>
<td>199.5</td>
</tr>
</tbody>
</table>
Table 4: Queue time comparison with inter-arrival time SCV = 5.

<table>
<thead>
<tr>
<th>Utilization</th>
<th>k = 2</th>
<th>k = 5</th>
<th>k = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SQT</td>
<td>90%CI</td>
<td>AQT</td>
</tr>
<tr>
<td>10%</td>
<td>124.1</td>
<td>0.1%</td>
<td>124.1</td>
</tr>
<tr>
<td>20%</td>
<td>82.1</td>
<td>0.2%</td>
<td>82.1</td>
</tr>
<tr>
<td>30%</td>
<td>85.9</td>
<td>0.3%</td>
<td>85.9</td>
</tr>
<tr>
<td>40%</td>
<td>97.2</td>
<td>0.4%</td>
<td>97.0</td>
</tr>
<tr>
<td>50%</td>
<td>117.0</td>
<td>0.5%</td>
<td>117.2</td>
</tr>
<tr>
<td>60%</td>
<td>153.1</td>
<td>0.7%</td>
<td>153.5</td>
</tr>
<tr>
<td>70%</td>
<td>228.7</td>
<td>1.0%</td>
<td>228.7</td>
</tr>
<tr>
<td>80%</td>
<td>460.2</td>
<td>1.6%</td>
<td>457.7</td>
</tr>
<tr>
<td>90%</td>
<td>918.1</td>
<td>2.0%</td>
<td>917.5</td>
</tr>
</tbody>
</table>

Table 5: Cycle time comparison with Np = 20.

<table>
<thead>
<tr>
<th>Utilization</th>
<th>k = 2</th>
<th>k = 5</th>
<th>k = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCT</td>
<td>90%CI</td>
<td>ACT</td>
</tr>
<tr>
<td>10%</td>
<td>112.2</td>
<td>0.1%</td>
<td>112.2</td>
</tr>
<tr>
<td>20%</td>
<td>79.2</td>
<td>0.1%</td>
<td>79.1</td>
</tr>
<tr>
<td>30%</td>
<td>73.3</td>
<td>0.1%</td>
<td>73.3</td>
</tr>
<tr>
<td>40%</td>
<td>76.4</td>
<td>0.2%</td>
<td>76.3</td>
</tr>
<tr>
<td>50%</td>
<td>85.5</td>
<td>0.3%</td>
<td>85.6</td>
</tr>
<tr>
<td>60%</td>
<td>102.6</td>
<td>0.3%</td>
<td>102.7</td>
</tr>
<tr>
<td>70%</td>
<td>133.3</td>
<td>0.5%</td>
<td>133.7</td>
</tr>
<tr>
<td>80%</td>
<td>197.4</td>
<td>0.8%</td>
<td>198.0</td>
</tr>
<tr>
<td>90%</td>
<td>386.3</td>
<td>2.1%</td>
<td>391.8</td>
</tr>
<tr>
<td>95%</td>
<td>762.7</td>
<td>3.3%</td>
<td>787.2</td>
</tr>
</tbody>
</table>
time (ACT) computed based on Eq. (14), and the percentage difference from ACT compared with SCT (Diff%).

The second case is the same as the first one, except that \( N_p \) is 100. The results are shown in Table 6. The mean cycle times in Case 2 are shorter than that in Case 1 for each fixed \( k \) because the mean effective batch process time \( t_e \) decreases with \( N_p \). Hence, the mean cycle time becomes shorter.

In both two cases, the approximations perform well at all utilizations if the parallel batch size is 2 (where \( c^2/k = 1 \)). The absolute Diff% is less than 0.5% when \( N_p = 20 \) and less than 3.2% when \( N_p = 100 \). This is consistent with the finding in S-2: the model gives reliable results when \( c^2/k \) is one.

When \( c^2/k \) is far away from one, the Diff% becomes larger. It can be as large as 9.2% at 50% utilization when \( N_p = 100 \). However, the Diff% becomes smaller in both heavy and light traffic compared to that at the middle utilizations (40–70%). Similar results have been identified by Wu et al. (2011a) where they examined Eq. (1) in detail and observed that it is a good approximation for S-2 models only when the utilizations are high or low. Since Eq. (14) is derived based on the same decomposition assumption, it carries the same limitations.

6. Conclusion

Due to the complicated situations in practical manufacturing systems, a comprehensive classification of batching is essential to guide the right choice of queuing models. Through the proposed classification in Section 3, the corresponding approximate models are given in Section 4. We hope this classification can help both practitioners and researchers to use the correct models when facing batching in practical manufacturing systems.

When evaluating system performance, one should (1) identify the arrival process (i.e., single job or batch arrivals) from the upstream workstations, (2) examine the processing batch (i.e., single or multiple jobs) at the underlying workstation, and (3) investigate if there are product-induced setups when switching recipes. From (1) and (2), one can determine if there are transfer or parallel batches. From (3), one can decide if the serial batch exists. Then the right model can be chosen through Table 1.

To serve the purpose of classifications, we only provide basic models for approximation, while some more detailed models are given in the reference. In practice, transfer batch size may not be the same as parallel batch size. For example, a furnace, whose parallel batch size is four, may be fed by another batch process machine, whose parallel batch size is two. Therefore, the arrival batch size to the furnace is two but the parallel batch size is four. If the parallel batch size is a multiple of the transfer batch size, we may simply treat the transfer batch size as one, and the parallel batch size is the multiple of the transfer batch size. Then the case just reduces to S-2. When the parallel batch size is not a multiple of the transfer batch size, the ratio between transfer and parallel batches has to be considered. A general batching model derived by Hanschke (2006) can be used in this situation.

The models presented in the paper are restricted to single product. This assumption can be restrictive for serial batches. When the serial batch is induced by multiple products, polling model could be used and more general polling models need to be derived to consider the practical situations. Furthermore, the assumption of memoryless property for a product-induced setup may not hold in practical manufacturing systems. Wu and Zhao (submitted for publication) have proposed improved models based on more detailed classifications of product-induced setups. Those models can be used in conjunction with our batching taxonomy when memoryless property does not hold.

In addition to our basic models, the practical situation can be more complicated: the parallel batch size may be a random variable subject to different types of interruptions. Furthermore, based on the model proposed by Wu and McGinnis (2012), the impact of batching on system level performance can be crucial when the bottleneck is a batching machine. The impact of batching on intrinsic ratios (Wu & McGinnis, 2013) or on the throughput of parallel unreliable machines (Liu, Yang, Wu, & Hu, 2012) is left for future research.

Acknowledgments

The author would like to thank Dr. Ning Zhao for her support in simulation. This work was supported in part by Nanyang Technological University under Grant M4084193.

References


