The throughput rate of serial production lines with deterministic process times and random setups: Markovian models and applications to semiconductor manufacturing

Woo-sung Kim, James R. Morrison*

KAIST, Department of Industrial and Systems Engineering, Republic of Korea

A R T I C L E   I N F O
Available online 4 April 2014

Keywords:
Serial production line
Flow line
Tandem queues
Clustered photolithography tools
Semiconductor manufacturing

A B S T R A C T
We conduct exact analysis of serial production lines with deterministic service durations and various classes of random state-dependent setups between customers. Our focus is on assessing the production rate, or just-in-time (JIT) throughput. We demonstrate that such systems can be modeled as a Markov chain and consider various types of setups inspired by clustered photolithography tools in semiconductor manufacturing. We deduce when exact closed form expressions for the production rate are possible and when a numeric solution to the Markov chain balance equations are required. As these systems have shown promise for modeling process bound clustered photolithography tools, we study their accuracy versus detailed simulation for predicting the tool throughput. Various practical features such as the capacity of a pre-scan buffer and batch customers (to model wafer lots) are investigated.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction
Serial production lines, also known as flow lines, with random behavior can serve as prototypical models for a host of manufacturing systems including production and assembly lines; cf., [1–4] or, more recently [5–11]. As such, they have a rich history of research focus and have been applied in many contexts. However, excepting limited classes of systems, exact results do not exist and approximations and simulation methods are typically employed. We endeavor to obtain exact results for a class of such systems with practical application potential in semiconductor manufacturing.

Our performance metric of choice is the maximum production rate or just-in-time (JIT) throughput. The JIT throughput is the long term average rate at which customers exit the system when the input is never starved. That is, it is the throughput when customers arrive just-in-time to the flow line. This is equivalent to the maximum rate at which the system can produce customers. In [12], it was shown that the JIT throughput for two stage serial production lines can be expressed as \(1/E[\max(S_1, S_2)]\), where \(E\) is the expected value and \(S_i\) denotes the random service duration for stage \(i\). Based on this result, in [12–14], explicit expressions for the JIT throughput under exponential, Erlang and uniform service distributions were obtained. For the three stage case, it was demonstrated in [15] that the JIT throughput is \(1/E[\max(S_1, S_2, S_3)]\), where \(S_i\) is the random service time for stage \(i\).

A variety of papers pursued exact performance results based on this fact under different service time distributions; cf., [15–20]. Beyond these cases, exact analysis does not seem possible for stages with overlapping random service durations; cf., [1] or [4]. Refer also to [21].

Much recent work has focused on approximations based on aggregation [22–26] or decomposition methods [27–31]. These approximations can possess astonishing accuracy and have been used in many practical contexts; cf., [7]. However, like simulation (which can be computationally intensive relative to other methods), they do not as readily provide the same qualitative insights which may be possible via exact results. In addition, these approximations have not explicitly incorporated issues such as setups.

In clustered photolithography tools in semiconductor wafer manufacturing, various types of setups may be required. These setups have a large influence on the JIT throughput of system. While serial production lines have been used to study practical semiconductor manufacturing [32–35], no closed form expressions considering setups have been obtained. Our efforts are motivated by this lack of results. We focus on obtaining exact results and closed form expressions where possible. To this end, we will restrict attention to serial production lines with deterministic service times. While the JIT throughput of such a deterministic system is trivially the inverse of the longest process time, allowing random setups dramatically complicates analysis. Yet, setups are of practical importance and essential for models of certain manufacturing environments.

* Corresponding author.
E-mail address: james.morrison@kaist.edu (J.R. Morrison).
URL: http://x3D.kaist.edu (J.R. Morrison).

http://dx.doi.org/10.1016/j.cor.2014.03.022
0305-0548/© 2014 Elsevier Ltd. All rights reserved.
For certain classes of randomly occurring setups inspired by clustered photolithography tools (CPTs), we will demonstrate that the maximum production rate can be exactly calculated. This is possible by exploiting the fact that, for JIT customer arrivals, the delay customers experience in the system can be modeled as a Markov chain. In certain cases, explicit expressions for the system throughput can be obtained. In others, the throughput can be calculated by solving for the equilibrium probabilities of the Markov chain. With these results in hand, we endeavor to apply them to models of CPTs. Such tools have a structure that is particularly amenable to our approach. CPTs are essential components of semiconductor wafer fabrication and can be modeled as a serial production line when in the process bound region; cf., [32–35]. We incorporate practical features such as a wafer buffer before the scanner of the CPT and batch arrivals. Generally, the scanner has the longest process time in a CPT and there is a buffer to ensure its throughput. We compare the throughput obtained via our models with detailed simulations; they are acceptable. Despite limitations associated with the model assumptions, we hope that the results may prove useful in practical contexts where analytic throughput expressions are of value.

Though we describe the systems under consideration in detail in the next section, we require a few ideas now. A serial production line consists of $M$ stages, labeled $m_1, ..., m_M$. Each stage consists of a single server or machine, which provides service. The process time for a customer at stage $m_i$ is deterministic with duration $r_i$. There are no buffers between stages, excepting the arrival buffer before the first stage (it has infinite capacity). This restriction is easy to relax and we will do so in the next section. There is a bottleneck stage; it is the first stage $m_{B}$ whose $r_B$ is greater than or equal to all other process times.

Aside from any additional time that may be required for a setup, the service time for every customer in stage $m_i$ is $r_i$. Recent theoretical results [36] suggest that it may be possible to relax this assumption and allow service times to depend on the customer. For Type II and Type IV setups (with $\alpha = 0$), we obtain $\alpha$ explicitly (Propositions 1–4).

We study Type I, II and III setups separately. A Type IV setup that is not subsumed by another class of setup cannot be modeled by our methods. Detailed simulation will be required. We address Type V and VI setups separately. As mentioned, our focus is on the JIT throughput, the maximum production rate, of each system; call it $\alpha$. The contributions of the work are as follows.

- For Type I and III setups and Type II setups (with $S = B$ or $B - 1$), we determine $\alpha$ explicitly (Propositions 1–4).
- For Type II setups, with certain conditions on $S$, to be detailed later, we show that the system can be modeled as a Markov chain (Proposition 6), identify its recurrent states (Lemma 4) and determine when the equilibrium probabilities, and thus $\alpha$, can be obtained explicitly (Propositions 7 and 8). Otherwise, we give the balance equations that will allow the calculation of $\alpha$ (Proposition 9).

The results are summarized for ease of reference in Table 1.

The remainder of the paper is organized as follows. In Section 2, we describe our systems of interest. We also introduce relevant known results. In Section 3, we derive the JIT throughput under Type I and Type II setups. In Section 4, we focus on Type II setups with a certain condition on $S$. We identify cases where the throughput can be explicitly obtained. In Section 5, we consider combination setup Types V and VI. We apply the results to serial production line models of CPTs in Section 6. In Section 7, a detailed simulation of a process bound CPT is compared with our models. The simulation includes wafer transport robots and is based on industrial CPT data from the literature. Concluding remarks are provided in Section 8.

Hereafter, we use the term flow line. It is synonymous with serial production line. An early abbreviated version of a subset of the work reported here appeared in conference form in [37].

2. Preliminaries

We first describe deterministic flow lines and relevant known results. Type I, II and III setups are also detailed.

2.1. Deterministic flow lines

A flow line is composed of a series of stages from which customers receive service in sequence. In each stage, there is a single server or machine which provides service. There is an infinite capacity buffer before the first stage. A finite capacity buffer may be provided for each stage after the first. As detailed in [38], the finite capacity intermediate buffers can be modeled as a
stage with zero process time. For example, an intermediate buffer of capacity 30 can be treated as a series of 30 stages, each with 0 service time. Including buffers, we label the series of M stages as m1, m2, ..., mM. The service time for a customer in stage m1 is deterministic; we use τi to denote its duration. We do not allow stage failure; all stages are reliable. Let B denote the index of the bottleneck stage. That is, B is the index of that stage for which τB = max{τi : i = 1, ..., M}. If there is more than one stage with this maximum service duration, the bottleneck is the one with the smallest index. Customers are served in first come first served (FCFS) manner. This can be relaxed easily in the external queue prior to stage m1. The stages are non-idling, so that service starts immediately when a customer and stage are both available. See Fig. 1.

Let ak denote the arrival time of the kth customer, ak ≥ ak−1 for positive integers k. (Later, we assume ak = 0 for all customers k to calculate the JIT throughput.) Arrivals enter the infinite capacity buffer prior to stage m1 until all preceding customers have received service from and vacated m1. The customer then immediately enters stage m1 and begins service. After receiving service from stage m1, a customer enters stage m1+1 as soon as it is available (i < M); service begins upon entry. After service, delay occurs in stage m1 if the downstream stage is not vacant. The customer exits the system only after service completion from stage mj, such as a system is called open, since all customers eventually exit the system. The elementary evolution equations provide a recursive description of the behavior. Let Xkj denote the entry time of customer k to stage mj (it is the start of service.) The start times obey the recursions

\[ X_{k+1,j} = \max(a_{k+1}, X_{k+2,j}), \]
\[ X_{k+1,j+1} = \max(X_{k+1,j+1} + \tau_{j+1}, X_{kj+1}), \]
\[ X_{k+1,M} = \max(X_{k+1,M−1} + \tau_{M−1}, X_{km} + \tau_{M}), \]

for all k = 0, 1, ..., j = 2, ..., M − 1, with the initial conditions X0,m = −∞, for all m = 1, ..., M. (The initial conditions enforce an initially empty restriction.) Let d(k) denote the delay in stage mj for customer k. That is, d(k) = Xkj + τj − Xkj+1 for k = 1, 2, ..., and j = 1, ..., M − 1. Customer 1 experiences no delay.

Our focus is on maximum production rate. As such, we assume that customers arrive just-in-time (JIT). That is, as soon as the previous customer advances to stage m2, a new customer is immediately available to enter stage m1. Equivalently, ak = 0 for all k > 0.

In [38], it was proved that there is no contention for stages after the bottleneck. The delays after the bottleneck are zero; we hereafter ignore them.

Lemma 1 (No delay after the bottleneck). For any customer k ≥ 1, d(k) = 0, B ≤ j ≤ M.

Let C(k) denote the completion time of customer k from stage m1. That C(k) = Xkj + τj. It will be convenient to consider that the duration of service required from the bottleneck may exceed τB. This will help to model Type I setups. Thus, for the bottleneck stage only, we consider the deterministic process time to be τ(k) for customer k, so long as τ(k) ≥ τB. The following theorem is proved in [39].

Theorem 1 (Recursion for exit times). For k > 0, the exit time for customer k + 1 obeys the recursion

\[ C_{m}(k+1) = \max(a_{k+1} + T(k + 1), C_{m}(k) + \tau_{B}(k + 1)), \]

where T(k) = Σj>1τj + τB(k) is the sum of the service times of customer k.

Intuitively, either a customer experiences no delay or exits the system one bottleneck process time after its predecessor.

Our performance measure is the throughput. Consider an initially empty deterministic flow line. Let DT denote the number of customers to depart from stage mM by time T. Then, we define the throughput as

\[ \alpha = \liminf_{T \to \infty} \frac{D(T)}{T}. \]

If the departure process from the system is renewal, one obtains α as 1/C, where C is the mean inter-departure time. This follows from the elementary renewal theorem in [40].

2.2. Channel concepts and in-channel delay

Next, channel concepts and the structure of delay in a channel are described. A deterministic flow line can be decomposed into segments called channels [39]. Each channel behaves similarly.

Definition 1. In a deterministic flow line, a stage mj is called a dominating stage if τj > τj for all j < i.

Let A denote the set of indices of the dominating stages, and σ denote the number of dominating stages. Define / as the σth smallest index in A. Note that /1 = 1 and / = B. All stages between and including subsequent dominating stages are defined as a channel.

Definition 2. The σth channel in a deterministic flow line is the set of stages \( \{m_1, m_2, ..., m_\phi \} \) for 1 ≤ σ ≤ A.
Infinite queue

![Diagram of a deterministic flow line.](image)

For example, consider a deterministic flow line with $M = 10$ and column vector $\mathbf{r} = (r_1, r_2, \ldots, r_{10})^T = (40, 10, 20, 50, 30, 70, 20, 100, 90, 15)^T$. The dominating stages are $m_1, m_5, m_6$ and $m_7$. We have $A = \{1, 4, 6, 8\}$, $\beta(1) = 1$, $\beta(2) = 4$, $\beta(3) = 6$ and $\beta(4) = 8$. The flow line has three channels: $(m_1, m_2, m_3, m_4, m_5, m_6)$ and $(m_7, m_8, m_9)$. Let

Let $Y^\phi(k)$ denote the delay experienced by customer $k$ in channel $\phi$, excluding any in the last stage of the channel. That is,

$Y^\phi(k) = \sum_{i=1}^{m(\beta(\phi)}) - 1 d_i(k).$  

Let $r_i(k)$ denote residence time of customer $k$ in stage $i$, which is defined as $r_i(k) = d_i(k)$ on non-bottleneck stages. For the bottleneck stage, $r_i(k) = r_\phi(k) + d_\phi(k).$ Let $S^\phi(k)$ denote the maximum total delay that customer $k$ can experience in channel $\phi$ is $S^\phi(k) = \sum_{j=1}^{m(\beta(\phi)}) - 1 Y^\phi(k)$. Then, the maximum total delay that customer $k$ can experience in channel $\phi$ is $S^\phi(k) = \sum_{j=1}^{m(\beta(\phi)}) - 1 Y^\phi(k)$.

The following theorem is proved in [39].

**Theorem 2** (Recursion for delays in the channel $\phi, \phi > 1$). In an initially empty deterministic flow line, $Y^\phi(k)$, the delay in channel $\phi > 1$, can be obtained as follows:

$Y^\phi(k) = \min\{Y^\phi(k-1), r_\phi(k-1) + r_\phi(k-1)(k-1) - \max[r_\phi(k-1), \Delta k, \phi]^+\},$

where $[\cdot]^+ = \max[0, \cdot]$, $\Delta k, \phi = X_{k, \phi} - X_{k-1, \phi} - 1$. The delay at a stage $j$, $\beta(\phi) \leq j < \beta(\phi+1)$, is $d_j(k) = \min\{d_{j-1}(k) + (k-\beta(\phi+1)) - r_j\}$. $S^\phi(k) = \sum_{j=1}^{m(\beta(\phi)}) - 1 Y^\phi(k)$. For $k > 1$ and $j = \beta(\phi), \ldots, \beta(\phi+1), - 1$. For the last channel, the recursion can be simplified as follows.

**Theorem 3** (Recursion for delays in the last channel). In an initially empty deterministic flow line, the delay in the last channel, $Y^{\sigma-1}(k)$, can be obtained as follows:

$Y^{\sigma-1}(k) = \min\{Y^{\sigma-1}(k-1), r_\sigma(k-1) + r_\sigma(k) - \max[r_\sigma(k-1), \Delta k, \sigma]^+\},$

where $S^\sigma(k) = \sum_{j=1}^{m(\beta(\sigma-1))} \tau_\sigma(k-B+j) - r_j$.

The stage delays obey the recursion

$d_j(k) = \min(r_\sigma(k-1) - r_j, Y^{\sigma-1}(k-1) - S^\sigma(k-1) + 1),$

d_\sigma(k) = 0,

for $k > 1$ and $j = \beta(\sigma-1), \ldots, B-1$.

These results allow one to calculate the delays recursively and will be essential to our results.

**2.3. Set-ups**

In the absence of setups, the JIT throughput of a deterministic flow line is simply $\alpha = 1/\tau_\phi$. However, introducing randomness via setups complicates analysis. We first consider and treat each type of setup separately.

Type I setups occur only at the bottleneck and effectively cause an increase in the duration of the bottleneck process time for that customer. We assume that every customer requires a Type I setup with probability $1 - p$, independent of all other customers. We also define $\tau_1$ as the deterministic duration of a Type I setup. For a customer $k$, we thus consider the service time at the bottleneck including the setup time to be $\tau_\phi(k) = \tau_\phi + \tau_1$ with probability $1 - p$. Otherwise, with probability $p$, $\tau_\phi(k) = \tau_\phi$.

Type II setups require all processes prior to and including $m_{\phi-1}$ to be vacant before the setup can begin. As such, we consider it a state-dependent setup (it can only begin when the tool is in certain states). Setups occur randomly and also have deterministic duration. Independently of all other customers, with probability $1 - p_1$, a setup is required prior to the start of service for a given customer. It begins when all processes $m_1$ through $m_{\phi-1}$ are vacant and has duration $\tau_\phi$. Once the setup is complete, new wafers can enter the flow line. In practical CPT systems, this distinguished stage resides in the last channel. We can obtain results when $m_{\phi-1}$ is in the last channel.

Type III setups start when all processes are vacated. We assume that setups occur randomly and have deterministic duration. For each customer, with probability $1 - p_\phi$, independent of all other customers, a setup is required. Use $\tau_{\phi}$ to denote the deterministic duration of a Type III setup. If a Type III setup is required for customer $k$, after all previous customers have exited the system, the setup begins. After the setup is complete, customer $k$ enters the system.

**3. Maximum production rate**

In this section, we derive expressions for the maximum production rate, or JIT throughput for Type I, II and III setups. The throughput can be derived with a little effort for Type I and III setups. For Type II setups, the throughput can be derived explicitly when $B - 1 \leq S_\phi \leq B$. Otherwise, the limiting probabilities for internal delays are required to calculate throughput. When the distinguished setup stage $m_{\phi-1}$ is in the last channel and $m_{\phi-1} < m_{\phi-2} \leq B - 2$, the internal delays can be modeled as a discrete-time time-homogeneous finite-state Markov chain. We can thus obtain the desired probabilities. However, if stage $S_\phi \leq \beta(\sigma-1)$, there is no work enabling one to determine these required probabilities analytically. Fortunately, for practical CPTs, the distinguished stage resides in the last channel. (See the modeling discussions in Sections 6 and 7.)

**3.1. Throughput with Type I and Type III setups**

Recall that a Type I setup occurs with probability $1 - p$, for each customer.

**Proposition 1.** The JIT throughput of a deterministic flow line with Type I setups is given as

$$\alpha = \frac{1}{\tau_\phi + \tau_1(1 - p)}$$

Like all results in this paper, we provide the proof in the Appendix. The JIT throughput with Type III setups can be derived similarly.

**Proposition 2.** The JIT throughput of a deterministic flow line with Type III setups is given as

$$\alpha = \frac{1}{p_\phi \tau_\phi + (1 - p_\phi)(\tau_\phi + \sum_{i=1}^{M} \tau_i)}$$

Type I and Type III setups give the intuitively clear result. For Type II setups, the analysis is more complicated.
3.2. Type II setups

For Type II setups, we assume that a setup is required for each customer with probability \(1 - p_B\). The setup duration, once all stages prior to and including the distinguished stage \(m_S\) are vacant, is deterministic with value \(\tau_S\). After the setup, the waiting customer immediately enters the flow line. The JIT throughput is given next for the cases \(S_H = B-1\) and \(S_H = B\).

**Proposition 3.** When \(S_H = B-1\), the JIT throughput of a deterministic flow line with Type II setups is given as

\[
\alpha = \frac{1}{p_B \tau_B + (1 - p_B) \max(\tau_B, \tau_B + \sum_{i=1}^{m_S} \tau_i)}
\]

**Proposition 4.** When \(S_H = B\), the JIT throughput of a deterministic flow line with Type II setups is given as

\[
\alpha = \frac{1}{\tau_B + (1 - p_B)(\tau_B + \sum_{i=1}^{m_S} \tau_i)}
\]

For \(S_H < B-1\), the analysis is more complicated.

**Proposition 5.** The interdeparture times satisfy the following recursion when \(S_H < B-1\):

\[
C_M(k+1) - C_M(k) = \begin{cases} 
\tau_B, & \text{w.p. } p_B \\
\tau_B + \left(\tau_B - \tau_B + \sum_{i=1}^{S_H} \tau_i - \sum_{i=1}^{S_H} \frac{\beta-1}{\tau_{S_H}} d_i(k)\right)^+, \text{ w.p. } 1 - p_B
\end{cases}
\]

where w.p. is shorthand for “with probability”.

Note that Proposition 5 is valid regardless of the location of the distinguished stage \(m_S, S_H < B-1\). To evaluate the interdeparture times, we require the delay terms \(d_i(k)\). If the setup stage is in the last channel, we can derive these delays by employing the results of [39]. Recall that, for practical CPTs, \(S_H\) is in the last channel. By applying the results of Theorem 3 to Proposition 5, we obtain the following lemma.

**Lemma 2.** For \(B(\sigma - 1) \leq S_H < B-1\), the interdeparture times

\[
C_M(k+1) - C_M(k) = \begin{cases} 
\tau_B, & \text{w.p. } p_B \\
\tau_B + \left(\tau_B - \tau_B + \sum_{i=1}^{S_H} \tau_i - \sum_{i=1}^{S_H} \frac{\beta-1}{\tau_{S_H}} d_i(k)\right)^+, \text{ w.p. } 1 - p_B
\end{cases}
\]

All terms except \(Y^{\sigma-1}(k)\), delay in the last channel, are known. Thus, we require \(Y^{\sigma-1}(k)\).

4. Delay in the last channel

For a Type II setup with distinguished stage \(m_S < B-1\) in the last channel, we require \(Y^{\sigma-1}(k)\). We next show that \(Y^{\sigma-1}(k)\) obeys a Markovian property and, as such, a Markov chain model can be constructed. The equilibrium probabilities, and thus the JIT throughput, can be extracted from this model.

4.1. Markovian property for \(Y^{\sigma-1}(k)\)

We first study the interentry times to the last channel; these are used in Theorem 2. For convenience, we drop the \(\phi\) notation for the last channel and write \(\Delta_k = X_{\phi}^{\sigma-1}(k, 1) - X_{\phi}^{\sigma-1}(k-1, 1)\). Recall that Type II setups occur with probability \(1 - p_B\).

**Lemma 3.** The interentry times can be expressed as

\[
\Delta_k = \begin{cases} 
\tau_{\phi(k-1)} + [Y^{\sigma-1}(k-1) - S_{\phi(k-1)}^{\phi}], \text{ w.p. } p_B \\
\tau_{\phi} + \sum_{n=1}^{m_S} \tau_n + \left[Y^{\sigma-1}(k-1) - \sum_{n=1}^{m_S} (\tau_B - \tau_n)\right]^+, \text{ w.p. } 1 - p_B
\end{cases}
\]

Here, w.p. denotes “with probability”.

This enables a Markovian a property for \(Y^{\sigma-1}(k)\).

**Proposition 6.** With Type II setups, the delay in the last channel \(Y^{\sigma-1}(k)\) can be considered as a discrete-time time-homogeneous finite-state Markov chain with \(k\) as the time index. The transitions and their probabilities are given by the following:

\[
Y^{\sigma-1}(k) = \begin{cases} 
\min_{\psi \leq \gamma} \{Y^{\sigma-1}(k-1) + (\tau_B - \tau_{\phi(k-1)})\}, \text{ w.p. } p_B \\
\gamma + \min_{\psi \leq \gamma} \{Y^{\sigma-1}(k-1), \sum_{n=1}^{m_S} (\tau_B - \tau_n)\}^+, \text{ w.p. } 1 - p_B
\end{cases}
\]

Here, w.p. denotes “with probability”.

The Markov chain is irreducible since all states are in one communicating class. Since it has a finite number of states, they are thus all positive recurrent. The equilibrium probabilities can be derived [40] to obtain the limiting interdeparture time distribution and hence \(\alpha\).

4.2. Limiting distribution of delay in the last channel

All results in this section refer to Type II setups. By Proposition 6, the delay customers face in the last channel can be modeled as a discrete-time, discrete-state time-homogeneous Markov chain. To simplify the expression for \(Y^{\sigma-1}(k)\), let \(\gamma = \tau_B - \tau_{\phi} - \sum_{i=1}^{S_H} \tau_i\), \(\eta = \tau_B - \sum_{n=1}^{S_H} \tau_n + \sum_{n=1}^{S_{H-1}} (\tau_B - \tau_n)\) and \(\delta = \tau_B - \tau_{\phi}\). Then,

\[
Y^{\sigma-1}(k) = \begin{cases} 
\min_{\psi \leq \gamma} \{Y^{\sigma-1}(k-1) + \delta\}, \text{ w.p. } p_B \\
\gamma + \min_{\psi \leq \gamma} \{Y^{\sigma-1}(k-1), \sum_{n=1}^{m_S} (\tau_B - \tau_n)\}^+, \text{ w.p. } 1 - p_B
\end{cases}
\]

There are three cases: \(\gamma > 0\) (\(\eta\) is anything), \(\gamma < 0\) and \(\eta < 0\), and \(\gamma < 0\) and \(\eta \geq 0\). We address each case separately.

In the first case, \(\gamma \geq 0\), the setups are inconsequential.

**Proposition 7.** When \(\gamma \geq 0\), the JIT throughput \(\alpha = 1/\tau_B\).

For the case of \(\gamma < 0\) and \(\eta < 0\), the Markov chain for delay in the last channel jumps to the zero state every time a setup occurs (which occurs with probability \(1 - p_B\)); refer to Fig. 2. Thus, the customer entering the tool following the Type II setup experiences no delay inside the flow line. With probability \(p_B\), there is no setup and delay increases by \(\delta\). The equilibrium probabilities for this Markov chain exist and must satisfy the balance equations:

\[
\Pi_0 = p_B \Pi_0 \\
\Pi_{\delta} = p_B \Pi_{\delta} + p_B^2 \Pi_0 \\
\Pi_{\delta} = p_B \Pi_{\delta} + p_B^2 \Pi_0 \\
\Rightarrow \Pi_{k\delta} = p_B \Pi_{(k-1)\delta} + p_B^k \Pi_0 \\
(1 - p_B) \Pi_{S_{H-1}} = p_B^{k-1} \Pi_0
\]

where \(k = \max(k \ | \ k\delta < S_{\phi(k-1)}^{\phi})\). Here, \(\Pi_0\) is the equilibrium probability that the system is in state \(a\); these are the limiting
probabilities for the delay in the last channel. It is straightforward
to solve these equations.

**Proposition 8.** For the case $\gamma < 0$ and $\eta > 0$, the limiting probabil-
ities for the delay in the last channel are

$$
\Pi_{l0} = p_{l0}(1-p_{l0}), \quad l = 0, 1, \ldots, k'.
$$

$$
\Pi_{lS_{\delta-n}} = p_{l0}^{k'+1}.
$$

With the above result, the mean interdeparture time can be calculated by taking the expectation of $C_M(k+1)-C_M(k)$ from
Lemma 2. Note that the throughput is the inverse of the mean
interdeparture time.

We now turn our attention the most interesting and complicated
case, $\gamma < 0$ with $\eta > 0$. Here, a Type II setup prior to customer
$k+1$ causes $Y^*^{-1}(k+1) < Y^*^{-1}(k)$, but does not always drive it
to zero. With probability $p_{l0}$, $Y^*^{-1}(k+1)$ is greater than $Y^*^{-1}(k)$.
If $Y^*^{-1}(k) \geq \sum_{i=1}^{\infty} \left[ \tau_i - \tau_{i-1} \right]$, the minimization term in the setup case of
Lemma 2 simplifies and no longer depends on the particular value of $Y^*^{-1}(k)$. As such, we define $Y_{l}^{*-1}(k)$ as

$$
Y_{l}^{*-1}(k) = \min \left\{ \sum_{i=1}^{\infty} \left[ \tau_i - \tau_{i-1} \right], (Y_{l}^{*-1}(k-1) + \delta), \right\}, \quad \text{w.p. } p_{l0}
$$

$$
(Y_{l}^{*-1}(k-1) + \delta), \quad \text{w.p. } 1-p_{l0}.
$$

$Y_{l}^{*-1}(k)$ can also be considered as a discrete-time time-homogeneous finite-state Markov chain and has similar structure with
$Y^*^{-1}(k)$. Note that it does not exactly represent the internal delay
in the last channel since all states with delay greater than or equal to
$Y^*^{-1}(k)$ are lumped together into a single state
$\sum_{i=1}^{\infty} \left[ \tau_i - \tau_{i-1} \right]$ in $Y^*^{-1}(k)$. However, it can be used to
calculate the JIT throughput by employing it in Lemma 2.

In Fig. 3, an example of $Y_{l}^{*-1}(k)$ is provided: $\delta = 20, \gamma = -80,$
and $S' = 160$.

Since the Markov chain for delay can jump up by $\delta$, down by
$-\gamma$ and is contained in the interval $[0, \sum_{i=1}^{\infty} \left[ \tau_i - \tau_{i-1} \right]]$, it is not
immediately obvious which delay values are possible. We define $Y$
as the set of states in the single communicating class of the
Markov chain for $Y_{l}^{*-1}(k)$. Fortunately, elementary results of
abstract algebra [41] enable us to clearly describe $Y$.

**Lemma 4.** When $\gamma < 0$ and $\eta \geq 0$, the set of communicating states
of the Markov chain for $Y_{l}^{*-1}(k)$ is

$$
Y = \{ y \mid y = kc, l \in Z, 0 \leq y \leq S' \} \cup \{ y \mid y = S' - lc, l \in Z, 0 \leq y \leq S' \},
$$

where $c$ is the greatest common divisor of $-\gamma$ and $\delta$, and $S' = \sum_{i=1}^{\infty} \left[ \tau_i - \tau_{i-1} \right]$. Here, $l$ is the integers.

**Corollary 1.** The maximum number of states of the Markov chain for
$Y_{l}^{*-1}(k)$ is $2 \lfloor S'/c \rfloor$, where $[n]$ is the minimum integer which is
greater than or equal to $n$.

Since, the Markov chain is positive recurrent, the equilibrium
probabilities can be obtained by solving the balance equations with
the normalization condition.

**Proposition 9.** When $\gamma < 0$ and $\eta \geq 0$, the equilibrium probabilities
satisfy

$$
\Pi_0 = \sum_{k \in [0, \delta]} (1-p_{l0}) \Pi_k,
$$

$$
\Pi_i = (1-p_{l0}) \Pi_{i-\delta} \quad \text{for states } i \in [0, \delta) \cap Y
$$

$$
\Pi_i = p_{l0} \Pi_{i-\delta} + (1-p_{l0}) \Pi_{i-\gamma} \quad \text{for states } i \in \left[ \delta, S' + \gamma \right] \cap Y
$$

$$
\Pi_i = p_{l0} \Pi_{i-\delta} \quad \text{for states } i \in \left[ S' + \gamma, S' \right] \cap Y
$$

$$(1-p_{l0}) \Pi_k = \sum_{k' \in [S' - \delta, S') \cap Y} p_{l0} \Pi_{k'}, \sum \Pi_i = 1.
$$

For example, consider the case $\delta = 20, \gamma = -80$, and $S' = 160$.
Employing the proposition, we can obtain

$$
p_{l0} \Pi_0 = (1-p_{l0})(\Pi_{20} + \Pi_{40} + \Pi_{60} + \Pi_{80}),
$$

$$
\Pi_i = p_{l0} \Pi_{i-20} + (1-p_{l0}) \Pi_{i-80} \quad \text{for } i = 20, 40, 60, 80
$$

$$
\Pi_i = p_{l0} \Pi_{i-20} \quad \text{for } j = 100, 120, 140
$$

$$(1-p_{l0}) \Pi_{160} = p_{l0} \Pi_{140}.
$$

By solving these equations, we obtain the equilibrium prob-
abilities for the delay in the last channel. With these probabilities,
the interdeparture times can be calculated numerically by
Lemma 2. Taking the expectation, we obtain the throughput.

5. Multiple setup types: Types V and VI

We now consider Type V and Type VI setups. Type VI setups are
simpler; we consider these first.

Type VI setups allow Type I and Type III setups to occur. For
each customer, let $p_{I}, p_{II}, p_{III}$ and $p_{S}$ denote the probability the
customer prompts a Type I, Type III, Type I and III, or no setup,
respectively, independently for each customer. As such, $p_{I} + p_{II} + p_{III} + p_{S} = 1$.

**Proposition 10.** For a deterministic flow line under Type VI setups, the JIT
throughput

$$
\alpha = \frac{1}{r_B + (p_I + p_{III}) r_I + (p_{II} + p_{III}) (r_{III} + \sum_{i \neq B} r_i)}
$$

Type V setups allow Type I and Type II setups to occur. For each wafer,
let $p_I, p_{II}$, $p_{III}$ and $p_S$ denote the probability the wafer
prompts a Type I, Type II, Type I and II, or no setup, respectively,
independently for each customer. Here, $p_{I} + p_{II} + p_{III} + p_{S} = 1$.

As before, there is a distinguished setup stage $m_{S_i}$. The Type II setups require us to consider three cases.

**Proposition 11.** For a deterministic flow line under Type V setups, if $S_I = B - 1$, the JIT throughput

$$
\alpha = \frac{1}{(p_I + p_{II}) r_B + (p_{III} + p_{S}) r_I + (p_{II} + p_{S}) (\sum_{i \neq B} r_i)}
$$

**Proposition 12.** For a deterministic flow line under Type V setups, if $S_I = B$, the JIT throughput

$$
\alpha = \frac{1}{(p_I + p_{II}) r_B + (p_{III} + p_{S}) r_I + (p_{II} + p_{S}) (\sum_{i \neq B} r_i)}
$$

When $\beta/(\sigma - 1) < S_B < B - 1$, we provide a Markov chain model
for $Y^*^{-1}(k)$ that allows us to calculate the inter-departure times
and thus the throughput. The procedure is similar to the Type II
development in the prequel.

**Proposition 13.** Considering Type V setups, when $\beta/(\sigma - 1) < S_B < B - 1$, the interdeparture times satisfy

$$
C_M(k+1) - C_M(k)
$$
Here, w.p. denotes “with probability”.

**Lemma 5.** For a deterministic flow line under Type V setups, the inter-entry times \( \Delta_k = X_{k, \beta \sigma^- 1} - X_{k-1, \beta \sigma^- 1} \) obey

\[
\Delta_k = \begin{cases} 
\tau_{\beta \sigma^- 1} + [Y^{\sigma^- 1}(k) - \sum_{i=1}^{S_\delta} \tau_g(k-1)] - \tau_g(k-1), & \text{w.p. } p_N \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1}, & \text{w.p. } p_I \\
\max \{ \tau_B + \sum_{i=1}^{S_\delta} \tau_g(k-1) - \min \{ Y^{\sigma^- 1}(k), \sum_{i=1}^{S_\delta} \tau_g(k-1) \}, \tau_{\beta \sigma^- 1} \}, & \text{w.p. } p_{II} \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} - \tau_g(k-1), & \text{w.p. } p_{I,II} \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B, & \text{w.p. } p_{I,III} \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B - \sum_{i=1}^{S_\delta} \tau_g(k-1), & \text{w.p. } p_{II,III} \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B - \sum_{i=1}^{S_\delta} \tau_g(k-1) + \tau_B + \tau_{\beta \sigma^- 1} + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B, & \text{w.p. } p_{I,II,III} \\
\tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B - \sum_{i=1}^{S_\delta} \tau_g(k-1) + \tau_B + \tau_{\beta \sigma^- 1} + \tau_{\beta \sigma^- 1} - \tau_g(k-1) - \tau_B - \sum_{i=1}^{S_\delta} \tau_g(k-1), & \text{w.p. } p_{I,II,III} \\
\end{cases}
\]

where \( g(k) = k + i - B \).

**Lemma 6.** For a deterministic flow line under Type V setups, the delay in the last channel,

\[
Y^{\sigma^- 1}(k) = \begin{cases} 
\min \{ Y^{\sigma^- 1}(k) - \tau_{\beta \sigma^- 1}, \tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1), \tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1) \}, & \text{w.p. } p_N \\
\tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1), & \text{w.p. } p_I \\
\tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1) - \tau_{\beta \sigma^- 1}, & \text{w.p. } p_{II} \\
\tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1) - \tau_{\beta \sigma^- 1} + \tau_B, & \text{w.p. } p_{I,II} \\
\tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1) - \tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1}, & \text{w.p. } p_{I,II,III} \\
\tau_{\beta \sigma^- 1} + \tau_B(k-1) - \tau_g(k-1) - \tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} + \tau_{\beta \sigma^- 1}, & \text{w.p. } p_{I,II,III} \\
\end{cases}
\]

To obtain a Markov chain, we define \( X[k] = (t_{\beta \sigma^- 1}(g_{\beta \sigma^- 1}(k)), \ldots, t_{\beta \sigma^- 1}(g_{\beta \sigma^- 1}(k)), Y^{\sigma^- 1}(k)) \).

**Proposition 14.** With Type V setups, \( X[k] \) can be considered as a discrete-time time-homogeneous finite-state Markov chain with \( k \) as the time index. The transitions and their probabilities are given by the following:

\[
\begin{align*}
t_{\beta \sigma^- 1}(g(k)) &= t_{\beta \sigma^- 1}(g_{\beta \sigma^- 1}(k-1)) \quad \text{for } \beta \sigma^- 1 \leq i \leq B - 1 \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B, \quad \text{w.p. } p_N \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_i, \quad \text{w.p. } p_I \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_{\beta \sigma^- 1}, \quad \text{w.p. } p_{II} \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_{\beta \sigma^- 1} - \tau_{\beta \sigma^- 1}, \quad \text{w.p. } p_{I,II} \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_{\beta \sigma^- 1} - \tau_{\beta \sigma^- 1} + \tau_B, \quad \text{w.p. } p_{I,II,III} \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_{\beta \sigma^- 1} - \tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1}, \quad \text{w.p. } p_{I,II,III} \\
t_{\beta \sigma^- 1}(g(k)) &= \tau_B + \tau_{\beta \sigma^- 1} - \tau_{\beta \sigma^- 1} + \tau_B + \tau_{\beta \sigma^- 1} + \tau_{\beta \sigma^- 1}, \quad \text{w.p. } p_{I,II,III} \\
\end{align*}
\]

The Markov chain is irreducible and all states are positive recurrent. The equilibrium probabilities will give the limiting interdeparture time distribution and hence \( \alpha \). The state space and balance equations are harder to express analytically than before. However, for given parameter values (e.g. the number of stages and their \( \tau_i \) values) it can be done.

6. Features relevant for practical CPTs

We now turn our attention to the study of features that are relevant for CPTs in semiconductor wafer manufacturing. In this
context, the “customer” is a wafer on which the semiconductor devices are created. We will hereafter use the term wafer in place of customer. Identical wafers are typically processed in a group called a wafer lot, or simply lot. Similarly, we use the term process module (or simply module) in place of stage. A reticle is a patterned piece of glass through which light is scanned to generate a pattern on the semiconductor wafer.

Note that our approach is particularly appropriate for these largely deterministic tools with a relatively short sequence of processes. Random behaviors outside of those considered will violate our assumptions.

6.1. Clustered photolithography tools (CPTs)

Clustered photolithography tools are by far the most expensive tools in semiconductor wafer manufacturing and cost as much as US$120 million [42]; setups play a large role in determining their maximum production rate (JIT throughput). We first provide a detailed description of these tools. Issues and restrictions related to developing serial production line models for them are considered in a later section.

CPTs comprise three sections: pre-scan processes, the photolithography scanner and post-scan processes. The portions of the tool devoted to these processes are often referred to as the pre-scan track, scanner and post-scan track, respectively. Wafers require pre-scan services prior to entering the scanner. In the pre-scan track, the surface of each wafer is coated with photoresist chemicals and treated with preexpose baking operations. These wafers are then exposed to the desired wavelengths of light by the scanner. A desired image is projected from a reticle. After the scanner, wafers are baked and developed in the post-scan track.

The scanner is by far the most expensive component of the tool (it can cost as much as US$100–US$125 million [43]) and, as such, is generally (and should well be) designed as the bottleneck of the system. There is typically a significant capacity wafer buffer before the scanner, so that full line relaxation is not possible. While we allow a setup when changing from one class of wafer to another, we otherwise require identical service times for all wafers. We will see how accurate the resulting models can be in the next section.

We will introduce the technique in the next section. Wafers are transported from one process to the next via wafer transport robots. A CPT typically hosts three or four such robots to cater to its 20 or so process modules. The scheduling of these robots is an essential task, however, we will assume that it has been well done. When well done, the system steady state throughput is the same as the inverse of its bottleneck activity duration (including any robot time that cannot be conducted in parallel). As mentioned, the scanner is so expensive that it is unwise for any other part of the tool to restrict throughput. If the wafer transport robots are the system bottleneck, other methods such as Petri net models should be employed. As mentioned in [39], practical studies of the application of flow line models to the study of CPTs have been quite successful. Assuming that they are not the system bottleneck, we will incorporate wafer transport robots by adding essential wafer handling time to the process times.

Note that there are several limitations. We are incorporating the wafer transport robots only as a constant addition to the process durations. This is possible by assuming that the robots are well scheduled and that the scanner is the bottleneck. Otherwise, the flow line relaxation is not possible. While we allow a setup when changing from one class of wafer to another, we otherwise require identical service times for all wafers. We will see how accurate the resulting models can be in the next section.

The classes of setups we have studied have practical significance in CPTs. Type I setups correspond to a reticle alignment. Wafers of a new lot may require that the scanner conduct a setup to ensure the pattern or reticle is properly aligned. Type II setups model a pre-scan track setup. In some CPTs, a new lot may require all pre-scan processes to conduct a setup before the wafers from that lot can enter. This Type II setup can only commence once all previous wafers have exited the pre-scan track modules. The pre-scan track modules typically end before the wafer buffer positioned ahead of the bottleneck scanner, so that $m_{00} < B - 1$. Occasionally, a full flush of the entire tool may be required before new wafers can enter; this is the Type III setup. Type IV setups correspond to a setup of the post-scan track. When a different class of lot is ready to enter the post-scan track, it may require the post-scan modules to conduct a setup. This setup may have to wait until the post-scan modules are empty of wafers before starting. Type V and VI setups are combinations of the three basic types.

Some manufacturers conduct the reticle alignment at the beginning of every lot, no matter if it is a different class or not. This is done to improve yield, but costs throughput capability. The Type IV setup can be modeled as a Type I setup in the absence of a post-scanner buffer or if it is short in duration relative to an existing Type I setup. However, if any post-scan track setup does not satisfy these conditions,
our methods will be insufficient to address it. More detailed models (e.
g. simulation) should be considered.

For brevity, we only consider Type II setups in the remainder of this
section. The others are straightforward to address in similar
manner. We now consider two features here: a buffer immediately
before the bottleneck (the so-called pre-scan buffer) and batch
arrivals (to model wafer lots).

6.2. Pre-scan buffer

We now study the JIT throughput under Type II setups as a
function of the size of a buffer placed immediately before
the bottleneck. In the context of a CPT, such a buffer is referred to as
a pre-scan buffer. Suppose the buffer has a capacity of \( N \) and can
thus be modeled as a series of \( N \) processes with zero process times.
That is, \( \tau_{B-N} = \tau_{B-N+1} = \cdots = \tau_{B-1} = 0 \). Processes \( m_1, \ldots, m_{B-N-1} \)
must be vacant prior to the start of the Type II setup. Note that
there may also be other buffers (stages with 0 process time) in the
CPT, but we are assuming at least \( N \) buffer slots immediately prior to
the bottleneck.

Let \( D^p = \frac{B}{C_0} + \frac{1}{\bar{r}_1} \) be the maximum delay possible in
the last channel excluding the pre-scan buffer. The delay in the last
channel in Proposition 6 simplifies to:

**Corollary 2.** The delay in the last channel with a buffer of capacity \( N \)
immediately before the bottleneck satisfies:

\[
\gamma^{-1}(k) = \begin{cases}
\min\{D^p + N\tau_B, \gamma^{-1}(k-1) + (\tau_B - \tau_{\beta(s-1)})\} & \text{w.p. } p_u \\
\tau_B - \tau_{\beta} - \sum_{n=1}^{N} \tau_n + \min\{\gamma^{-1}(k-1), N\tau_B\} & \text{w.p. } 1 - p_u
\end{cases}
\]

By solving the balance equations for this Markov chain, we can
obtain the limiting distribution for the delay in the last channel as
a function of \( N \). If \( \gamma > 0 \), the JIT throughput \( \alpha = 1/\tau_B \). Otherwise, if
\( \gamma < 0 \) and

\[
\tau_B - \tau_{\beta} - \sum_{n=1}^{N} \tau_n + N\tau_B > 0,
\]

one must resort to a numerical solution of the balance equations.

However, if \( \gamma < 0 \) and

\[
\tau_B - \tau_{\beta} - \sum_{n=1}^{N} \tau_n + N\tau_B \leq 0,
\]

we can obtain an analytic expression for the JIT throughput. In this
case, employing the techniques of the previous section, the mean
interdeparture time, \( E[C] \), can be obtained as

\[
E[C] = \frac{1}{\alpha} = p_u \tau_B + q_u \sum_{n=0}^{N_u} \sum_{i=1}^{n_u} \left( n_u \delta + \sum_{i=1}^{n_u} \tau_i \right) + q_u \sum_{n=n_u+1}^{N_u} \left( n_u \eta_u + \sum_{i=1}^{n_u} \tau_i \right),
\]

where \( q_u = 1 - p_u \), \( N_u = \max\{k|k\delta < N\tau_B\} \) and \( N_s = \max\{k|k\delta < \frac{B}{C_0} + N\tau_B\} \). The JIT throughput \( \alpha = 1/E[C] \).

6.3. Batch arrivals

In semiconductor manufacturing, wafers are grouped into
batches called wafer lots, or simply lots, of up to 25 wafers.
Typically, all wafers in a lot are of the same class. As such, the
first wafer in a lot may prompt a setup but the others do not. Thus,
we now assume that a Type II setup may be required for the first
wafer of every lot with probability \( 1 - p_u \) (and is not required for
wafer lots within a lot). Let \( n_w \) denote the constant number of wafers
in a lot. Let \( \Omega(k,l) \) denote the overall index of the \( k \)th wafer of the
\( l \)th lot (\( 1 \leq l \leq n_w \)). That is, the count of the wafers to have entered
the system before and including this wafer. The channel delay
recursions simplify to

\[
y^{-1}(\Omega(k,1)) = \min\{\beta_{\Omega(s-1)}, y^{-1}(\Omega(k-1, n_w)) + (\tau_B - \tau_{\beta(s-1)})\} \quad \text{w.p. } p_u
\]

\[
+ \min\{y^{-1}(\Omega(k-1, n_w)), \sum_{n=n_s+1}^{B-1} (\tau_B - \tau_n)\} \quad \text{w.p. } 1 - p_u
\]

for the first wafer of a lot. For \( 2 \leq l \leq n_w \),

\[
y^{-1}(\Omega(k,l)) = \min\{\beta_{\Omega(s-1)}, y^{-1}(\Omega(k,l-1)) + (\tau_B - \tau_{\beta(s-1)})\} \quad \text{w.p. } p_u
\]

By the above recursion, we can calculate the delay in the last
channel numerically. However, for the case \( n_w \tau_B - \tau_{\beta(s-1)} \geq \frac{B}{C_0} \) \( \Omega(k, l) \)th wafer to enter the system is

\[
(\tau_B + \sum_{n=n_s+1}^{B-1} \tau_n + \sum_{i=1}^{n_s} \tau_i - \tau_{\beta(s-1)}) \tau_B \leq 0,
\]

we can derive the interdeparture time between two successive lots analytically.

By this recursion, when a lot requires a setup, the
first wafer of the next
lot is never delayed inside the system. If a setup is not required,
the last wafer of the lot always achieves maximum internal delay.

The mean interdeparture time can be obtained explicitly for this
case by conditioning on whether a setup occurs or not. With probability \( p_u \), the inter-departure time between two lots is \( n_w \tau_B \). If a setup is required, the inter-departure time between the
\( (\Omega(k+1,1) \) and \( \Omega(k, n_w) \)th wafer to enter the system is

\[
(\tau_B + \sum_{n=n_s+1}^{B-1} \tau_n + \sum_{i=1}^{n_s} \tau_i - \tau_{\beta(s-1)}) \tau_B \leq 0,
\]

taking \( (n_w - 1) \tau_B \) to finish the remaining wafers in the lot.

We thus obtain

\[
E[C|\Omega(k+1, n_w)] - E[C|\Omega(k, n_w)] = p_u n_w \tau_B + (1 - p_u)(n_w - 1) \tau_B + \left( \tau_B + \sum_{n=1}^{n_s} \tau_n + \sum_{i=1}^{n_s} \tau_i \right).
\]

The throughput in lots per unit time is the inverse of this.

6.4. Equipment failure

Equipment failures are common in manufacturing. Certain
classes of equipment failures can be included in the framework
we have developed. In particular, any failure that behaves as one of
the classes of setups can be equivalently studied using those
results. If one wishes to include failures with a different prob-
bility or downtime duration than an existing setup, these can be
added as a separate class of setup and treated as in the multiple
setup cases, or via similar extensions.

The issue is more complicated if we allow other types of
failures. For example, if all servers in a stage fail, one must
must determine what will be done with the wafers in the tool after
that stage, in that stage and before the stage. If a single server at
some stage fails, the tool may continue to operate at reduced
capacity until it is vacant and then the repair may be conducted.
Detailed models for each class of failure and the associated
operating practices should be developed. We relegate such develop-
ment to future work.

7. Numerical example and simulation results

In this section, we compare the throughput of our models
under Type I and II setups with simulations of full CPTs (including
wafer transport robots, wafer lots and a pre-scan buffer) based on
actual data for a deep UV CPT [44]. Type III setups are similar to the Type I case. The robot movements are dictated by the longest waiting pair (LWP) method [45]. CPTs consist of process modules which provide different services: cool plate (CP), hot plate (HP), post-scan bake hot plate (PEB), edge exposures (EE), spin coater (SC), spin developer (SD), low-pressure adhesions (LPAH) and stepper (scanner). Wafers are transferred by four single arm robots. Refer to [44] for further details. We consider two deep-UV process flows: both are top anti-reflective coating processes (TARC). These process flows, including the deterministic process times are given in Table 2 as TARC1 and TARC2. The number in parenthesis is the number of identical process modules devoted exclusively to that process (stage). For example, HP(2) in the TARC1 process means that there are two identical modules which provide the hot plate process. Just prior to the scanner, a buffer is provided to increase throughput. The capacity of the buffer is five wafers. We use detailed simulation of this tool (including robots under the LWP method) as a baseline. Our models will be compared against it.

Since some processes have more than one module, we use serialization to obtain a deterministic flow line model. The serialized flow line model can provide throughput estimates within 0.5% compared to a flow line with multiple modules devoted to each process (see [39]). In serialization, we model n parallel modules each with process duration \( \tau_m \) as n serial stages with process duration \( \tau_m/n \). For example, two HP process modules each with 80 s process durations in parallel are modeled as two stages in sequence whose service times are 40 s each. To grossly account for the robots, we add the pick and place time to every process time (for every process in the serialized model). While this ignores the robot activity sequence, it can be quite an effective relaxation for predicting throughput assuming that the robot sequence achieves the bottleneck throughput (and the system is process bound). The errors are acceptable (less than 0.1%). The results for TARC are very similar and omitted for brevity.

We conduct studies on Type I setups. First, for the TARC1 flow line model in Table 3, we allow Type I setups. Set \( n_w = 24 \) wafers/lot. The setup duration is a constant 330 s. In the detailed simulation, the number of lots between setups is randomly generated (via a simple Markov chain) with \( n \) being the average number of lots between setups (\( n \) will vary from 1,...,10); this is called the train size. We calculate the throughput using the models from this paper. In our model, we use \( 1 - p_j = 1/n \) as the probability that the first wafer of a lot requires a Type I setup. All wafers except the first never require a setup. We compare this with the results obtained from a detailed simulation of the CPT using TARC1 (from Table 2), with the same conditions, including four wafer transport robots under the longest waiting pair policy (see above). We simulate 30 times for each case for 10,000 lots and collect data from the last 7000 lots. The Markov chain model throughput is compared to the detailed simulation results in Fig. 4. The errors are acceptable (less than 0.1%). The results for TARC 2 are very similar and omitted for brevity.

We conduct similar studies with Type II setups. The setup duration is 1000 s and processes \( m_1, \ldots, m_{19} \) in the flow line model must be vacant before the setup commences. (In the detailed CPT model this is \( P_1, \ldots, P_7 \).) The results are provided in Fig. 5. When the train level is low, the errors are about 5%. As the train level increases, the frequency of setups decreases. As the setup time is required less frequently, the JIT throughput approaches the rate of the bottleneck stage. The model error also decreases as the train level decreases.

## 8. Concluding remarks

In this paper, we studied the production rate, or JIT throughput, of deterministic serial production lines with a variety of random setup types. In all cases considered, Markov chain models were constructed. Closed form analytic expressions for the production rate could be obtained for some setup types and particular parameter combinations. For others, the Markov chain was studied to develop equations from which the JIT throughput could be obtained numerically. Practical features such as batches of customers of the same class and a buffer just prior to the bottleneck were considered. Numerical studies comparing these models to detailed simulations of CPTs (including wafer handling robots) were conducted. The results were acceptable.

There are numerous questions that could be addressed in future work. Can the results be used to model unreliable machines that do not behave akin to one of our setup classes? Is it possible to allow the Type II setup distinguished process \( m_{19} \) to reside prior to the last channel? Can recent work on proportional service flow lines be extended to allow for customer classes with different process times? Can random duration setups be allowed? Finally, it would be very interesting to consider throughput data from an operating CPT.

### Table 2

<table>
<thead>
<tr>
<th>Process order</th>
<th>TARC1 process</th>
<th>PT</th>
<th>TARC2 process</th>
<th>PT</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>HP(2)</td>
<td>80</td>
<td>HP(2)</td>
<td>80</td>
</tr>
<tr>
<td>P2</td>
<td>LPAH(2)</td>
<td>90</td>
<td>LPAH(2)</td>
<td>90</td>
</tr>
<tr>
<td>P3</td>
<td>CP(2)</td>
<td>60</td>
<td>CP(2)</td>
<td>60</td>
</tr>
<tr>
<td>P4</td>
<td>SC(2)</td>
<td>65</td>
<td>SC(2)</td>
<td>65</td>
</tr>
<tr>
<td>P5</td>
<td>SC(1)</td>
<td>50</td>
<td>HP(2)</td>
<td>90</td>
</tr>
<tr>
<td>P6</td>
<td>HP(2)</td>
<td>90</td>
<td>CP(2)</td>
<td>60</td>
</tr>
<tr>
<td>P7</td>
<td>CP(2)</td>
<td>60</td>
<td>SC(1)</td>
<td>50</td>
</tr>
<tr>
<td>B</td>
<td>BUFFER(5)</td>
<td>0</td>
<td>BUFFER(5)</td>
<td>0</td>
</tr>
<tr>
<td>P8</td>
<td>SCANNER</td>
<td>100</td>
<td>SCANNER</td>
<td>100</td>
</tr>
<tr>
<td>P9</td>
<td>PEB(2)</td>
<td>90</td>
<td>PEB(2)</td>
<td>90</td>
</tr>
<tr>
<td>P10</td>
<td>CP(2)</td>
<td>60</td>
<td>CP(2)</td>
<td>60</td>
</tr>
<tr>
<td>P11</td>
<td>EE(2)</td>
<td>90</td>
<td>EE(2)</td>
<td>90</td>
</tr>
<tr>
<td>P12</td>
<td>SD(3)</td>
<td>130</td>
<td>SD(3)</td>
<td>130</td>
</tr>
<tr>
<td>P13</td>
<td>HP(2)</td>
<td>90</td>
<td>HP(2)</td>
<td>90</td>
</tr>
<tr>
<td>P14</td>
<td>CP(2)</td>
<td>60</td>
<td>CP(2)</td>
<td>60</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Stages</th>
<th>TARC1 PT</th>
<th>TARC2 PT</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_{11}</td>
<td>41.5</td>
<td>41.5</td>
</tr>
<tr>
<td>m_{12}</td>
<td>41.5</td>
<td>41.5</td>
</tr>
<tr>
<td>m_{13}</td>
<td>48.5</td>
<td>48.5</td>
</tr>
<tr>
<td>m_{14}</td>
<td>48.5</td>
<td>48.5</td>
</tr>
<tr>
<td>m_{15}</td>
<td>31.5</td>
<td>31.5</td>
</tr>
<tr>
<td>m_{16}</td>
<td>31.5</td>
<td>31.5</td>
</tr>
<tr>
<td>m_{17}</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>m_{18}</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>m_{19}</td>
<td>57</td>
<td>46.5</td>
</tr>
<tr>
<td>m_{20}</td>
<td>46.5</td>
<td>46.5</td>
</tr>
<tr>
<td>m_{21}</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>m_{22}</td>
<td>33</td>
<td>57</td>
</tr>
<tr>
<td>m_{23}</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>m_{24}</td>
<td>107</td>
<td>107</td>
</tr>
</tbody>
</table>
Acknowledgments

This work was partially supported by National Research Foundation of Korea (NRF) Grant 20110005696. The authors are grateful for the guidance of the reviewers and editors. The paper is much improved for their guidance.

Appendix A

Here, we prove the results stated in the paper.

Proof of Proposition 1. Consider customers \( k \) and \( k+1 \). With probability \( p_I \), a setup is not required for customer \( k+1 \). In this case, \( C_M(k+1) - C_M(k) = \tau_B \) by Theorem 1. Otherwise, a Type I setup is required with probability \( 1-p_I \). In this case, \( C_M(k+1) - C_M(k) = \tau_B + \tau_I \). Taking the expectation, we can obtain the results. □

Proof of Proposition 2. Consider customers \( k \) and \( k+1 \). With probability \( p_{II} \), a setup is not required for customer \( k+1 \). The inter-departure time for this case is \( \tau_B \) by Theorem 1. With probability \( 1-p_{II} \), a Type II setup is required. After customer \( k \) finishes its service in \( m_{S_{II}} \), the setup begins. Note that \( S_{II} = B-1 \). Thus, \( C_M(k+1) = \max\{C_M(k) + \sum_{i=1}^{M} \tau_i, C_M(k) + \tau_B\} \). Since, the start time of customer \( k+1 \) in stage 1, \( X_{k+1} \), is \( C_M(k+1) + \tau_B \), the exit time of customer \( k+1 \) can be obtained as follows:

\[
C_M(k+1) = \max\left\{C_M(k) + \tau_B + \sum_{i=1}^{M} \tau_i, C_M(k) + \tau_B\right\}
\]

\[
= \max\left\{C_M(k) + \sum_{i=1}^{M} \tau_i + \tau_B, C_M(k) + \tau_B\right\}
\]

\[
= \max\left\{C_M(k) + \tau_B + \sum_{i=1}^{B} \tau_i, C_M(k) + \tau_B\right\}
\]

If \( \tau_B + \sum_{i=1}^{B-1} \tau_i \) the service of customer \( k \) in \( m_B \) is not yet finished when customer \( k+1 \) finishes its service in \( m_{B-1} \). In this case, delay occurs and the inter-departure time is \( \tau_B \). Otherwise,
customer k + 1 experiences no delay, and $C_M(k + 1) = C_M(k) + \tau_B + \sum_{i=1}^{b} \tau_i$. □

**Proof of Proposition 4.** Consider customers k and k + 1. With probability $p_{II}$, a setup is not required, the inter-departure time is $\tau_B$. With probability $1 - p_{II}$, the setup begins after customer k finishes its service in $m_k$. Note that there is no delay after the bottleneck, and customer k + 1 does not experience delay. In this case, $C_M(k + 1) = C_M(k) + \tau_B + \sum_{i=1}^{b} \tau_i$. Since $C_M(k) = C_M(k) + \sum_{i=1}^{M} \tau_i$, we thus obtain the result. □

**Proof of Proposition 5.** From [39],

$$C_M(k + 1) = \max \left\{ a_k^* + 1, \min \left\{ a_k^* + 1 + \sum_{i=1}^{M} \tau_i, C_M(k) + \tau_B \right\} \right\}, \quad (1)$$

where $a_k^* = X_{k+1}$, if customer k + 1 requires a Type II setup. Otherwise, $a_k^* = 0$. Thus, if a setup is not required, the inter-departure time between two successive customers is $\tau_B$. Let $V_{k_j}$ denote the exit time of customer k from stage $m_i$. If customer k + 1 requires a setup, it enters the first stage immediately following the setup. Thus,

$$a^*_{k+1} = X_{k+1} = V_{k+S} + \tau_B, \quad C_M(k) = V_{k+S} + \sum_{i=1}^{M} \tau_i + \sum_{i=1}^{b} d_{k,i}.$$ Substituting these into (1) concludes the proof. □

**Proof of Lemma 2.** By Theorem 3,

$$\sum_{i=1}^{n} d(k) = \min \left\{ Y^{\sigma} - 1(k), \sum_{i=1}^{n} (\tau_B - \tau_n) \right\}.$$ The result is then immediate by Proposition 5. □

**Proof of Lemma 3.** Customer k immediately enters the last channel when customer k − 1 enters $m_{k,\beta_1,...,1}$ with probability $p_B$. When there is a setup, customer k enters the last channel $\tau_1 + \cdots + \tau_{(\rho_{k-1} - 1)}$ time units after the setup completes (on account of the vacant modules up to and including $m_{k1}$). Thus, the start time of customer k in $m_{k,\beta_1,...,1}$, $X_{k,\beta_1,...,1}$, is

$$X_{k,\beta_1,...,1} = \begin{cases} X_{k-1,\beta_1,...,1} + \tau_{(\rho_{k-1} - 1)} + d_{(\rho_{k-1} - 1)}(k-1), & \text{w.p. } p_B \\ V_{k-1,S} + \tau_n + \sum_{i=1}^{\beta_1-1} \tau_i, & \text{w.p. } 1 - p_B. \end{cases}$$

We study these terms separately. By Theorem 3, $d_{(\rho_{k-1} - 1)}(k-1) = \min(\tau_{(\rho_{k-1} - 1)} - Y^{\sigma} - 1(k-1) - \sum_{i=1}^{\beta_1-1} \tau_i, \tau_{(\rho_{k-1} - 1)} - 1)^+$. The right term always achieves the minimum. Thus, with probability $p_B$,

$$\Delta_k = \tau_{(\rho_{k-1} - 1)} - Y^{\sigma} - 1(k-1) - \sum_{i=1}^{\beta_1-1} \tau_i - \tau_{(\rho_{k-1} - 1)}.$$ With probability $1 - p_B$, a setup is required, we can write $V_{k-1,S}$ as

$$V_{k-1,S} = X_{k-1,\beta_1,...,1} + \sum_{n=\beta_1-1}^{\beta_k} \tau_n + \sum_{n=\beta_1-1}^{\beta_k} d_n(k-1),$$

where

$$\sum_{n=\beta_1-1}^{\beta_k} d_n(k-1) = \min \left\{ \tau_{(\rho_{k-1} - 1)} - Y^{\sigma} - 1(k-1) - \sum_{n=\beta_1-1}^{\beta_k} (\tau_n - \tau_{n}), \sum_{n=\beta_1-1}^{\beta_k} (\tau_n - \tau_{n}) \right\}.$$ Using this fact completes the proof. □

**Proof of Theorem 6.** The recursion for $Y^{\sigma} - 1(k)$ follows by substituting Lemma 3 into Theorem 3. Some algebra is required. $Y^{\sigma} - 1(k)$ is a stochastic process whose state space is countable and finite. Since $k$ is a positive integer, the process has discrete-state and evolves in discrete-time. By the recursion, the probabilities for the next value $Y^{\sigma} - 1(k + 1)$ are thus dictated with given $Y^{\sigma} - 1(k)$. As such this is a discrete-time homogeneous finite-state Markov chain. □

**Proof of Proposition 7.** By Proposition 6, if $\gamma \geq 0$, the delay always increases whether a setup is required or not. Every customer faces delay in the system. Via Theorem 1, the inter-departure times are $\tau_B$. □

**Proof of Proposition 8.** By solving the balance equations, we prove the result. □

**Proof of Lemma 4.** We first explicitly enumerate the states in the single communicating class; denote the set of these states by $Y$. $Y = \{y | y = b^0 + nY, 1 \leq n \in Z, q \in \{0, 1\}, 0 \leq y \leq S^{\sigma - 1}\}$.

There are two cases: $q = 0$ or $q = 1$. So, the communicating class is the union of two sets: $\{y | y = b^0 + nY, 1 \leq n \in Z, 0 \leq y \leq S^{\sigma - 1}\}$ and $\{y | y = b^0 + nY, 1 \leq n \in Z, 0 \leq y \leq S^{\sigma - 1}\}$.

This representation for the communicating class can be simplified using fundamental elementary results of abstract algebra. In [41], the set $A = \{m \omega + nY | m, n \in Z\}$ is equivalent to the set $\{l | l = tc, t \in Z$ and $c = \gcd(\delta, \gamma)\}$, where $\gcd(a,b)$ represents the greatest common divisor of $a$ and $b$.

For each case, applying the above results, $b^0 + nY$ can be expressed as $tc$, where $c = \gcd(\delta, \gamma)$. Substituting $tc$ into $b^0 + nY$, we can prove the lemma. □

**Proof of Propositions 10-13.** The results follow similar to Propositions 1-5, Depending on if the setup is required or not and which type of setup is required, we use the approach from the previous proofs. For example, with probability $p_{II}$, the customer does not require a setup. For this case, the inter-departure time is always $\tau_B$ by Theorem 1. With probability $p_B$, a Type I setup is required. For this case, $C_M(k + 1) = C_M(k) + \tau_B + \tau_1$ (see the proof of Proposition 1). For the Type II setup, see the proof of Propositions 3-5. When two setups are required, note that the Type I setup begins when the customer enters the bottleneck stage. This is equivalent to a bottleneck time of $\tau_B + \tau_1$. Thus, when two setups are required, it takes $\tau_1$ longer than the case with only a Type II or III setup. □

**Proof of Lemmas 5 and 6.** We prove each case separately. We begin with $X_{k,\beta}$. With probability $p_B$, the kth customer does not require a setup. For this case, the kth customer enters the last channel right after the $k$−1th customer departs from $m_{k,\beta_1,...,1}$. Thus,

$$X_{k,\beta_1,...,1} = X_{k-1,\beta_1,...,1} + \tau_B + d_{(\rho_{k-1} - 1)}(k-1).$$

By Theorem 2, $d_{(\rho_{k-1} - 1)}(k-1)$ can be expressed as follows:

$$d_{(\rho_{k-1} - 1)}(k-1) = \min \left\{ \tau_B + d_{(\rho_{k-1} - 1)}(k-1) - \tau_{(\rho_{k-1} - 1)}, \sum_{i=1}^{(\rho_{k-1} - 1)} (\tau_0 - \tau_{i}) + \tau_{(\rho_{k-1} - 1)} - 1 \right\}.$$ Since $\sum_{i=1}^{(\rho_{k-1} - 1)} (\tau_0 - \tau_{i})$ is the maximum internal delay for customer k − 1 in the last channel, the right side is always smaller than the left side.

With probability $p_B$, the kth customer requires a Type I setup. Note that it does not relate to $X_{k,\beta_1,...,1}$, so the entry time for customer k − 1 to the last channel is the same as above.

With probability $p_B + p_{II}$, the kth customer requires a Type II setup. Thus, the entry time to the last channel is

$$X_{k,\beta_1,...,1} = V_{k-1,\beta_1,...,1} + \tau_B + \sum_{n=\beta_1-1}^{\beta_k} \tau_n.$$
\( V_{k-1,S_k} \) is the departure time of the \( k \)-th customer from the setup stage, and is expressed as

\[ V_{k-1,S_k} = X_{k-1} - (\beta \sigma - 1) + \frac{S_k}{\beta \sigma - 1} \tau_i + \sum_{i=1}^{S_k} d(k-1). \]

From Theorem 2,

\[ \sum_{i=1}^{S_k} d(k-1) = \min \left\{ \frac{S_k}{\beta \sigma - 1} \tau_i \right\}. \]

\[ \tau^0(k-1) - \sum_{i=1}^{S_k+1} \left[ \rho_i (g_i(k-1)) - \tau_{i-1} \right] \]

Applying the above equations, we prove Lemma 5. Lemma 6 can be proved by applying Lemma 5 to Theorem 2.

Proof of Proposition 14. We use Lemma 6. Recall that \( g_i(k) = g_{i+1} \) \( (k-1) = k - 1 - B \). Thus, \( \tau_{i} g_i(k) = \tau_{i+1} g_{i+1}(k-1) \) for \( \beta \sigma - 1 \leq B \leq 1 \). There are 4 cases.

With probability \( \rho_{i+1} \) customer \( k \) does not require any setup. In this case, \( \tau_{i} g_i(k) = \tau_i \). \( \tau_{i} g_i(k) \) is obtained from Lemma 6. With probability \( 1 - \beta \sigma - 1 \), customer \( k \) requires a Type I setup. For this case, \( \tau_{i} g_i(k) = \tau_i + \tau_i \). \( \tau_{i} g_i(k) \) is also obtained from Lemma 6. Similarly, we can derive the other cases: \( \tau_{i} g_i(k) = \tau_i + \tau_i \) when both setups are required. \( \tau_{i} g_i(k) \) can be derived from Lemma 6.

Note that \( \rho_1 g_1(k) \) can have two values: \( \tau_1 \) and \( \tau_1 + \tau_1 \). The possible \( \tau_{i} g_i(k) \) values are countable. Thus, \( X(k) \) is a multidimensional Markov chain with countable state space. All terms in \( X(k) \) including \( S^0(k) \) \( (k) \) can be derived from \( X(k-1) \). Thus, \( X(k) \) can be considered as a discrete-time time-homogeneous finite-state Markov chain.

References