Transient and Steady-State Analysis of Multistage Production Lines With Residence Time Limits

Feifan Wang, *Student Member, IEEE*, and Feng Ju, *Member, IEEE*

***Abstract*— Residence time limits for the intermediate products are commonly seen in production systems due to quality requirements. A part in a buffer typically needs to be scrapped or reworked if its residence time exceeds the maximum allowable residence time, while a part needs to keep waiting in the buffer if its residence time is less than the minimum required residence time. Such limits could be required at multiple stages of the production line, making it difficult to analyze both the steady-state and transient production system performance. The problem under study is formulated as a multistage geometric serial production line with residence time limits. To deal with its large state space, a two-machine-one-buffer subsystem isolated from a multistage geometric serial production line is first analyzed for both steady-state and transient performance. Furthermore, a novel aggregation method, including the steady-state analysis and transient analysis, is proposed to evaluate the overall system performance. The proposed aggregation method substantially reduces the complexity of the problem and makes the analysis of the problem tractable. Compared with the simulation, the aggregation method maintains high accuracy in estimating both steady-state and transient performance measures. Such a method provides quantitative tools for effective performance evaluation and prediction on the factory floor.**

***Note to Practitioners*—It is well known that potential quality problems may occur and production cost may increase if the residence time of intermediate products is left uncontrolled. The residence time limits become a concern, especially in the automotive industry, food industry, and semiconductor industry, where intermediate products stay in a stage susceptible to defects. The limits also apply to large scale additive manufacturing of thermoplastic polymers. The deposition of each layer is subject to lower and upper bounds of surface temperature, which, in practice, are equivalent to minimum required residence time limit and maximum allowable residence time limit, respectively. With residence time considered, the size of state space for the overall production system model will grow exponentially, as the system's size increases, which brings tremendous challenges to evaluate such a system both in the short term and in the long run. In this article, we introduce a novel modeling approach for a multistage geometric serial production line with residence time limits and propose a method that drastically reduces the complexity of the system. Such a method provides a quantitative tool to effectively evaluate the performance of multistage geometric lines with residence time limits.**

***Index Terms*— Aggregation method, multistage production lines, residence time, transient analysis.**

Manuscript received July 24, 2019; revised November 4, 2019; accepted February 3, 2020. Date of publication March 25, 2020; date of current version January 6, 2021. This article was recommended for publication by Associate Editor B. Vogel-Heuser and Editor F.-T. Cheng upon evaluation of the reviewers' comments. This work was supported by NSF under Grant CMMI-1829238. *(Corresponding author: Feng Ju.)*

The authors are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: (feifan.wang@asu.edu; feng.ju@asu.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2020.2979179

NOMENCLATURE

D	Number of machines in multistage line.
m_i	ith machine of multistage line.
m_1^{sub}	First machine of subsystem.
m_2^{sub}	Second machine of subsystem.
s_i^{sub}	State of machine m_i^{sub} in subsystem.
p_i	Failure probability of machine m_i in multistage line.
p_i^{sub}	Failure probability of machine m_i^{sub} in subsystem.
r_i	Repair probability of machine m_i in multistage line.
r_i^{sub}	Repair probability of machine m_i^{sub} in subsystem.
B_i	ith buffer in multistage line.
B	Buffer in subsystem.
N_i	Capacity of buffer B_i.
N	Capacity of buffer B.
n	Buffer occupancy for buffer B.
$T_{i,\min}$	Minimum required residence time for buffer B_i.
$T_{\min}$	Minimum required residence time for buffer B.
$T_{i,\max}$	Maximum allowable residence time for buffer B_i.
$T_{\max}$	Maximum allowable residence time for buffer B.
τ_i	Residence time of the ith part in buffer B.
$p_i^s(t)$	Starvation probability of machine m_i in cycle t.
p^s	Starvation probability of machine m_1^{sub}.
$p_i^b(t)$	Blockage probability of machine m_i in cycle t.
p^b	Blockage probability of machine m_2^{sub}.
$P_{1,1}^{(k)}$	Probability that machine m_k^{sub} that is up in one cycle is still up in the next cycle.
$P_{1,0}^{(k)}$	Probability that machine m_k^{sub} that is up in one cycle is down in the next cycle.
$P_{0,1}^{(k)}$	Probability that machine m_k^{sub} that is down in one cycle is up in the next cycle.
$P_{0,0}^{(k)}$	Probability that machine m_k^{sub} that is down in one cycle is still down in the next cycle.
$x(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}}, t)$	Probability for state $(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}})$ in cycle t.
$\Phi(n, \tau_1, \tau_2)$	Conditional probability that the second part in buffer B has residence time τ_2 given that there are n parts in buffer B and the first part in buffer B has residence time τ_1.

I. INTRODUCTION

IN PRODUCTION systems, residence time typically refers to the time that a part spends in a buffer before entering the next machine. Such time is usually subject to certain limits, including both upper and lower limits, to make sure the requirement of product quality is met. For instance, in the food industry, the perishability of food is often considered [1], [2]. There are time limits for intermediate products in every processing stage from raw materials to final products. These requirements need to be met for products like yogurt, beer, and bread [3], [4]. In semiconductor manufacturing, residence time limits also receive substantial attention [5]–[7]. The time that a wafer stays in a process module of a cluster tool is subject to limits to prevent the wafer from being premature or defective. Residence time limits can also be observed in battery production as well [8].

The rapid development in information and communication technologies makes it possible to monitor and control production systems with residence time limits in real-time [9]. Each part carries its individual process information, such as residence time [10]–[12]. Therefore, the residence time of each part in the buffer could be captured in real-time so as to determine the usability of the part—whether to be sent downstream or scrapped. In addition, machines and material flow could be controlled to increase system throughput and reduce wastes due to scrap. How to develop an analytical method for evaluating the performance of such a complex system becomes a central problem.

Early works study residence time limits mainly on a two-machine serial production line and do not yet extend the analysis to a large-scale problem with more than two machines [8], [13]–[15]. To analyze a multistage geometric serial production line with residence time limits, we take a two-machine-one-buffer subsystem, isolated from the multistage geometric serial production line, as a building block, and develop a Markov chain model to analyze the subsystem first. An approximate method to model residence time is introduced to reduce the size of the state space of the subsystem. In addition, based on the analysis of two-machine-one-buffer subsystems, the aggregation method is applied to obtain both the steady-state and transient performance of a multistage geometric serial production line with residence time limits. Validated by simulation experiments, the proposed aggregation method maintains high accuracy in estimating both steady-state and transient performance measures. The main contribution of this article is twofold. One is the approximate method to model a two-machine-one-buffer subsystem, which makes the analysis of a two-machine-one-buffer subsystem efficient and also provides a building block for the aggregation method. The other is the aggregation method, which evades the direct modeling for multistage geometric serial production lines with residence time limits. Instead of defining virtual machines like the early work for the aggregation method [16], we use starvation probabilities and blockage probabilities to support the iteration of the aggregation method without modifying parameters of machines and buffers in each iteration, which increases the flexibility of the aggregation method to deal with systems with large state space.

The remainder of this article is structured as follows. Section II reviews the related literature. Section III introduces assumptions and formulates the problem. In Section IV, we present the approximate modeling for a two-machine-one-buffer subsystem. In Section V, the aggregation method is proposed to estimate both the steady-state and transient performance of a multistage geometric serial production line with residence time limits. The accuracy of the proposed aggregation method is investigated in Section VI. Finally, conclusion and future directions are provided in Section VII.

II. LITERATURE REVIEW

The terms, deterioration and perishability, are widely used to represent the feature of the maximum allowable residence time limits [17], and they were initially applied to blood banks [17]–[20]. There is plenty of research working on classification and modeling on the maximum allowable residence time limits. A product may become obsolete after a certain time, or it may be decaying continuously [21]. For the decaying case, the deterioration can be age-dependent ongoing deterioration or age-independent ongoing deterioration [21]. Amorim *et al.* [3] and Pahl and Voß [17] provide several classifications to deal with perishability. Those studies on residence time limits primarily focus on the maximum allowable residence time. In this article, we consider both minimum required residence time and maximum allowable residence time as residence time limits and define the limits by constant thresholds without considering continuous decay.

Failures of machines may occur randomly and influence the performance of production systems [22]–[24]. Research on residence time limits is conducted on serial production lines under the uncertainty of machine reliability. One direction is to estimate and utilize the probability distribution of residence time, and those studies help design buffer capacity to reduce the defective rate [25]–[27]. However, defective parts in those production systems can only be detected at the end of the production line, and thus it wastes resources to process defective parts. Naebulharam and Zhang [28] evaluate Bernoulli serial production lines with deteriorating product quality by defining the quality buy rate for a buffer, and a part is removed from the system immediately after it is detected to be defective. Another direction to study residence time limits is to take residence time as a constraint into modeling. Ju *et al.* [15] evaluated the two-machine Bernoulli line with perishable intermediate products, and Ju *et al.* [14] further studied the production control of the two-machine Bernoulli line. Kang *et al.* [8] and Wang *et al.* [13] extended the analysis from a Bernoulli machine to a geometric machine, which, practically, is a more general reliability model. When residence time is considered in modeling, the state space can become too large to perform analysis. Kang *et al.* [8], Wang *et al.* [13], and Ju *et al.* [14], [15] use approximated methods to model residence time.

The aggregation method provides a framework to approximately evaluate multistage serial production lines. Li and Meerkov [22] proposed the aggregation method to estimate the steady-state performance measures of multistage Bernoulli serial production lines. Zhang *et al.* [29] extended the aggregation method to perform transient analysis for multistage Bernoulli

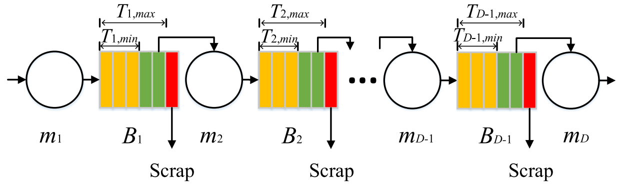


Fig. 1. Multistage geometric serial line with residence time limits.

serial production lines. Lee and Li [30] and Lee *et al.* [31] studied Bernoulli serial production lines with waiting time limits through the aggregation method. In those studies, a virtual Bernoulli machine is created in the analysis of multistage serial production lines with Bernoulli machines. Chen *et al.* [16] applied the aggregation method to geometric serial production lines by defining virtual geometric machines. In this article, each machine is assumed to be a geometric machine. Instead of defining virtual geometric machines, we use starvation probabilities and blockage probabilities to model the connection of neighboring two-machine-one-buffer subsystems without modifying parameters of machines and buffers. Such an aggregation method is flexible and easily applied to geometric serial production lines with residence time limits.

III. Problem Formulation

For simplicity purposes, the term "multistage line" is used to represent a multistage geometric serial production line with residence time limits for the rest of this article. The multistage line under study is shown in Fig. 1. Raw materials enter machine m_1 to be processed and continue to flow downstream until they finish the process in machine m_D or get scrapped from a buffer. The following assumptions define the machines, the buffers, and their interactions.

1) The multistage line consists of D machines (denoted by $m_1, m_2, \ldots, m_D$) and $(D-1)$ buffers (denoted by $B_1, B_2, \ldots, B_{D-1}$), where $D > 2$.
2) All machines are synchronized with a constant processing time (cycle time), which is the time to process a part.
3) Machines are subject to failures, and their reliability models are independent. The reliability model for each machine follows a geometric distribution. Specifically, if machine m_i is up in cycle $(k-1)$, it will still be up with probability $(1-p_i)$ and down with probability p_i during the kth cycle, for $i = 1, 2, \ldots, D$, and $k = 2, 3, \ldots$ If machine m_i is down in cycle $(k-1)$, it will be up with probability r_i and down with probability $(1-r_i)$ during the kth cycle, for $i = 1, 2, \ldots, D$, and $k = 2, 3, \ldots$ Here, p_i and r_i are defined as the failure probability and repair probability, respectively, for $i = 1, 2, \ldots, D$. The machine efficiency of machine m_i, denoted by e_i, is represented by $e_i = (r_i/(r_i + p_i))$.
4) Buffer B_i has a finite capacity N_i $(1 \leq N_i < \infty)$, for $i = 1, 2, \ldots, D-1$. First-in-first-out (FIFO) policy is assumed regarding the buffer outflow process.
5) The maximum allowable residence time for parts in buffer B_i is characterized by $T_{i,\max}$, for $i = 1, 2, \ldots, D-1$, counted as the number of cycles. A part in buffer B_i will be scrapped immediately at the beginning of the cycle when its residence time reaches $T_{i,\max}$. Let $T_{i,\max} \geq N_i$, otherwise N_i has no effect on the system.
6) The minimum required residence time for parts in buffer B_i is denoted by $T_{i,\min}$, for $i = 1, 2, \ldots, D-1$, counted as the number of cycles. A part is allowed to leave buffer B_i and enter machine m_{i+1} only when its residence time reaches or exceeds $T_{i,\min}$.
7) Machine m_i, for $i = 1, 2, \ldots, D-1$, is blocked during a time slot, if at the beginning of the cycle: 1) machine m_i is up; 2) buffer B_i is full; 3) machine m_{i+1} does not produce a part in this cycle due to machine failure or blockage; and 4) there will be no part scrapped from buffer B_i at the beginning of the next cycle. Machine m_D is never blocked. In addition, block-before-service policy is assumed.
8) Machine m_i, for $i = 2, \ldots, D$, is starved during a time slot, if machine m_i is up and no part in buffer B_{i-1} has residence time greater than or equal to $T_{i-1,\min}$. Machine m_1 is never starved.

The problem to be studied is to develop a method under assumptions 1–8 to evaluate both the steady-state and transient behaviors of the multistage line. Specifically, the system behavior of a multistage line is described by performance measures defined as follows.

1) *Production Rate, $PR_i(t)$, for $i = 1, \ldots, D$:* The expected number of parts produced by machine m_i in cycle t.
2) *Overall Production Rate, $PR(t)$:* The expected number of parts produced by the multistage line in cycle t.
3) *Overall Consumption Rate, $CR(t)$:* The expected number of parts that enter the multistage line in cycle t.
4) *Scrap Rate, $SR_i(t)$, for $i = 1, \ldots, D-1$:* The expected number of scrapped parts from buffer B_i in cycle t.
5) *Overall Scrap Rate, $SR(t)$:* The expected number of scrapped parts from the multistage line in cycle t.
6) *Work-in-Process, $WIP_i(t)$, for $i = 1, \ldots, D-1$:* The expected number of parts in buffer B_i in cycle t.
7) *Overall Work-in-Process, $WIP(t)$:* The expected number of parts in the multistage line in cycle t.
8) *Starvation Probability, $p_i^s(t)$, for $i = 1, \ldots, D$:* The probability that machine m_i is starved in cycle t, when machine m_i is up.
9) *Blockage Probability, $p_i^b(t)$, for $i = 1, \ldots, D$:* The probability that machine m_i is blocked in cycle t, when machine m_i is up.

For a multistage line, the overall production rate is equal to the production rate of the last machine, and thus we have $PR(t) = PR_D(t)$ for all t. The overall consumption rate is equal to the production rate of the first machine, so we have $CR(t) = PR_1(t)$ for all t. Scrap occurs in each buffer in the multistage line. Thus, the overall scrap rate is the summation of scrap rates of all buffers. Similarly, the overall work-in-process is the summation of work-in-processes of all buffers. Thus, we have $\mathrm{SR}(t) = \sum_{i=1}^{D-1} \mathrm{SR}_i(t)$ and $WIP(t) = \sum_{i=1}^{D-1} WIP_i(t)$ for all t. By assumptions 7 and 8, we have $p_1^s(t) = 0$ and $p_D^b(t) = 0$ for all t.

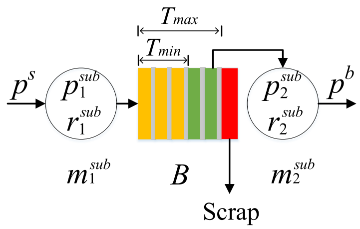


Fig. 2. Two-machine-one-buffer subsystem.

IV. Two-Machine-One-Buffer Subsystems

A. Model Formulation

The multistage line cannot be modeled directly using a single Markov chain due to its large state space. For instance, for a multistage line with ten machines, nine buffers, buffer capacity being $N_i = 6$ and maximum allowable residence time $T_{i,\max} = 8$ for $i = 1, 2, \ldots, 9$, the total number of system states is as large as 3.5×10^{24}. Alternatively, we start with a simple two-machine-one-buffer subsystem with a much smaller state space as shown in Fig. 2, which plays a role as a building block for the performance evaluation of the multistage line later. In the rest of this article, we will simply use the term "subsystem" to represent the two-machine-one-buffer subsystem.

A subsystem consists of two machines (denoted by m_1^{sub} and m_2^{sub}) and a buffer (denoted by B). Similar to a machine in a multistage line, machine m_i^{sub} in a subsystem is characterized by failure probability p_i^{sub} and repair probability r_i^{sub}, for $i = 1, 2$. Buffer B is described by its maximum allowable residence time $T_{\max}$, minimum required residence time $T_{\min}$, and buffer capacity N. A subsystem, isolated from a multistage line, is influenced by its upstream buffer through starvation and its downstream buffer through blockage. In order to cope with such effects, we use two probabilities, starvation probability p^s and blockage probability p^b, to model the starvation from the upstream buffer and the blockage from the downstream buffer. Specifically, p^s presents the probability that starvation occurs to machine m_1^{sub} when machine m_1^{sub} is up. p^b denotes the probability that blockage occurs to machine m_2^{sub} when machine m_2^{sub} is up.

To analyze a subsystem, we include only residence time of the first part in buffer B in the state, instead of recording residence times of all the parts. The rest of the residence time is then estimated using approximation. The detail of the approximate method is discussed in Section IV-C. Let $n \in \{0, 1, \ldots, N\}$ denote the buffer occupancy in buffer B. $\tau_1 \in \{0, 1, \ldots, T_{\max} - 1\}$ denotes the residence time of the first part in buffer B. We denote the states of machines m_1^{sub} and m_2^{sub} by s_1^{sub} and s_2^{sub}, respectively. Specifically, $s_i^{\text{sub}} = 1$ means that machine m_i^{sub} is up, and $s_i^{\text{sub}} = 0$ means that machine m_i^{sub} is down. Then, the system state of a subsystem is represented by $(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}})$. For the same example mentioned above with 3.5×10^{24} states, the number of states of the approximate model for each subsystem is 136. The size of the state space for a single model to be analyzed is significantly reduced.

B. Transition Equations

Based on the states of subsystems, the transition equations can be constructed. Let $x(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}}, t)$ denote the probability of state $(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}})$ in cycle t. Let $P_{i,j}^{(k)}$ denote the conditional probability that the state of machine m_k^{sub} is j given that its state is i in the previous cycle, for $i, j = 0, 1$ and $k = 1, 2$. Specifically, $P_{1,1}^{(k)} = 1 - p_k^{\text{sub}}$, $P_{1,0}^{(k)} = p_k^{\text{sub}}$, $P_{0,1}^{(k)} = r_k^{\text{sub}}$, and $P_{0,0}^{(k)} = 1 - r_k^{\text{sub}}$, for $k = 1, 2$. We introduce the operator $\Phi(n, \tau_1, \tau_2)$, which is defined as the conditional probability that the second part in buffer B has residence time τ_2 given that there are n parts in buffer B and the first part in buffer B has residence time τ_1. The method to estimate $\Phi(n, \tau_1, \tau_2)$ will be introduced in Section IV-C in detail. Let us consider the state $(1, i, s_1^{\text{sub}}, s_2^{\text{sub}})$ in cycle $(t+1)$, for $1 \le i \le T_{\max} - 1$ and $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$. There is one part in the buffer and its residence time could be any feasible value except 0. The transition equation for state $(1, i, s_1^{\text{sub}}, s_2^{\text{sub}})$ can be expressed as

$$
\begin{aligned}
&x\left(1, i, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\right) \\
&= x(1, i-1, 0, 0, t) P_{0,s_1^{\text{sub}}}^{(1)} P_{0,s_2^{\text{sub}}}^{(2)} \\
&\quad + x(1, i-1, 1, 0, t) p^s P_{1,s_1^{\text{sub}}}^{(1)} P_{0,s_2^{\text{sub}}}^{(2)} \\
&\quad + x(1, i-1, 0, 1, t) p^b P_{0,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \mathbf{1}_{\mathbb{N}^+}(i - T_{\min}) \\
&\quad + x(1, i-1, 1, 1, t) p^s p^b P_{1,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \mathbf{1}_{\mathbb{N}^+}(i - T_{\min}) \\
&\quad + x(1, i-1, 0, 1, t) P_{0,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \mathbf{1}_{\mathbb{N}^+}(T_{\min} + 1 - i) \\
&\quad + x(1, i-1, 1, 1, t) p^s P_{1,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \mathbf{1}_{\mathbb{N}^+}(T_{\min} + 1 - i) \\
&\quad + \sum_{j=\max(i, T_{\min})}^{T_{\max}-2} x(2, j, 0, 1, t)\left(1 - p^b\right) P_{0,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \Phi(2, j, i-1) \\
&\quad + \sum_{j=\max(i, T_{\min})}^{T_{\max}-2} x(2, j, 1, 1, t) p^s \left(1 - p^b\right) P_{1,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \Phi(2, j, i-1) \\
&\quad + x(2, T_{\max} - 1, 0, 0, t) P_{0,s_1^{\text{sub}}}^{(1)} P_{0,s_2^{\text{sub}}}^{(2)} \Phi(2, T_{\max} - 1, i-1) \\
&\quad + x(2, T_{\max} - 1, 1, 0, t) p^s P_{1,s_1^{\text{sub}}}^{(1)} P_{0,s_2^{\text{sub}}}^{(2)} \Phi(2, T_{\max} - 1, i-1) \\
&\quad + x(2, T_{\max} - 1, 0, 1, t) P_{0,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \Phi(2, T_{\max} - 1, i-1) \\
&\quad + x(2, T_{\max} - 1, 1, 1, t) p^s P_{1,s_1^{\text{sub}}}^{(1)} P_{1,s_2^{\text{sub}}}^{(2)} \Phi(2, T_{\max} - 1, i-1)
\end{aligned}
\tag{1}
$$

where $\mathbf{1}_{\mathbb{N}^+}(x)$ is an indicator function. $\mathbf{1}_{\mathbb{N}^+}(x) = 1$ if x is an positive integer ($x \in \mathbb{N}^+$), and 0 otherwise. Similar to (1), transition equations for all the other states can be formulated as shown in the Appendix.

C. Approximation of Residence Time

The operator $\Phi(n, \tau_1, \tau_2)$ is used in (1) for approximation of residence time. The operator $\Phi(n, \tau_1, \tau_2)$ is first proposed in [15] and first applied to geometric serial line in [8]. However, the first machine of a subsystem studied in this article can be starved, and the operator $\Phi(n, \tau_1, \tau_2)$ is influenced by starvation from the upstream buffer. Thus, the method that derives $\Phi(n, \tau_1, \tau_2)$ in the literature cannot be directly used.

A method that derives $\Phi(n, \tau_1, \tau_2)$ by taking starvation into consideration is provided in what follows.

Given current time t and residence time of the first part τ_1, we denote the state sequence of machine m_1^{sub} from cycle $(t - \tau_1 - 1)$ to $(t-1)$ by a (τ_1+1)-dimension vector V. Specifically

$$V = \left(s_1^{\text{sub}}(t-\tau_1-1), s_1^{\text{sub}}(t-\tau_1), \ldots, s_1^{\text{sub}}(t-1)\right) \quad (2)$$

where $s_1^{\text{sub}}(i)$ is the state of machine m_1^{sub} in cycle i. Denote by $\gamma(V)$ the probability that a sequence V occurs

$$\gamma(V) = \prod_{i=1}^{\tau_1} P^{(1)}_{s_1^{\text{sub}}(t-\tau_1-2+i), s_1^{\text{sub}}(t-\tau_1-1+i)}. \quad (3)$$

Since starvation may occur to machine m_1^{sub}, it is possible that no part is produced by machine m_1^{sub} during one cycle even though machine m_1^{sub} is up. Therefore, we define the (τ_1+1)-dimension vector W as a sequence for production of machine m_1^{sub}. Specifically

$$W = (w_1(t-\tau_1-1), w_1(t-\tau_1), \ldots, w_1(t-1)) \quad (4)$$

where $w_1(i) = 1$ represents that m_1^{sub} produces a part in cycle i, and 0 otherwise. Then, we have $w_1(i) \le s_1^{\text{sub}}(i)$ for any i. The probability that a sequence W occurs can be derived from the sequence V. Given W defined by (4), define $\mathcal{C}$ to be a collection of all V that can result in W. Specifically

$$\mathcal{C} = \left\{V \mid w_1(i) \le s_1^{\text{sub}}(i), i = t-\tau_1-1, \ldots, t-1\right\}. \quad (5)$$

We denote by $\Gamma(W)$ the probability that a sequence W occurs. Then, $\Gamma(W)$ can be expressed as

$$\Gamma(W) = \sum_{V \in \mathcal{C}} \gamma(V)\left(p^s\right)^k\left(1-p^s\right)^{\tau_1-k} \quad (6)$$

where $k = \sum_{i=t-\tau_1}^{t-1}(s_1^{\text{sub}}(i) - w_1(i))$.

We define a set, denoted by $\mathcal{A}$, that consists of all possible W that satisfies $w_1(t-\tau_1-1) = 1$ and $\sum_{i=1}^{\tau_1+1} w_1(t-i) = n$. Similarly, let $\mathcal{B}$ be a set that contains all possible W that satisfies $w_1(t-\tau_1-1) = 1$, $w_1(t-\tau_2-1) = 1$, and $\sum_{i=1}^{\tau_2} w_1(t-i) = n-2$. Specifically

$$\mathcal{A} = \left\{W \middle| w_1(t-\tau_1-1) = 1, \sum_{i=1}^{\tau_1+1} w_1(t-i) = n\right\}$$

$$\mathcal{B} = \left\{W \middle| w_1(t-\tau_1-1) = 1, w_1(t-\tau_2-1) = 1, \sum_{i=1}^{\tau_2} w_1(t-i) = n-2\right\}.$$

Then, the operator $\Phi(n, \tau_1, \tau_2)$ can be estimated as follows:

$$\Phi(n, \tau_1, \tau_2) = \frac{\sum_{W\in\mathcal{B}} \Gamma(W)}{\sum_{W\in\mathcal{A}} \Gamma(W)} \quad (7)$$

where the denominator is the probability that buffer occupancy is n and residence time of the first part is τ_1, while the numerator represents the probability that buffer occupancy is n, residence time of the first part is τ_1, and the residence time of the second part is τ_2.

D. Performance Measures of Subsystems

The estimated performance measures of a subsystem, for $t \in \mathbb{N}^+ \cup \{\infty\}$, include production rate $\widehat{PR}^{\text{sub}}(t)$, consumption rate $\widehat{CR}^{\text{sub}}(t)$, scrap rate $\widehat{\text{SR}}^{\text{sub}}(t)$, work-in-process, $\widehat{WIP}^{\text{sub}}(t)$, starvation probability $\widehat{\text{ST}}^{\text{sub}}(t)$, and blockage probability $\widehat{\text{BL}}^{\text{sub}}(t)$. Given p^s, p^b, and $x(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}}, t)$, the performance measures are estimated in (8)–(13).

The estimated production rate $\widehat{PR}^{\text{sub}}(t)$ is the expected number of parts the subsystem produces in cycle t. It is equal to probability that machine m_2^{sub} is up, there is at least one part in the buffer with residence time equal to or greater than $T_{\min}$, and machine m_2^{sub} is not blocked. The estimated consumption rate $\widehat{CR}^{\text{sub}}(t)$ represents the expected number of parts that enter the subsystem in cycle t. It is equivalent to the probability that buffer is not full and machine m_1^{sub} produces a part. $\widehat{\text{SR}}^{\text{sub}}(t)$ denotes the estimated number of scrapped parts from the subsystem in cycle t. It can be calculated as the probability that residence time of the first part in the buffer reaches $(T_{\max} - 1)$ but machine m_2^{sub} is not able to produce due to machine failure or blockage. $\widehat{WIP}^{\text{sub}}(t)$ denotes the estimated number of parts in buffer B in the subsystem in cycle t. $\widehat{\text{ST}}^{\text{sub}}(t)$ and $\widehat{\text{BL}}^{\text{sub}}(t)$ are starvation probability of machine m_2^{sub} and blockage probability of machine m_1^{sub}, respectively. The denominator of (12) presents the probability that machine m_2^{sub} is up, and the numerator is the probability that machine m_2^{sub} is up and the buffer is empty. Similarly, the denominator of (13) is the probability that machine m_1^{sub} is up, and the numerator is the probability that machine m_1^{sub} is up and the buffer is full.

V. Modeling Multistage Line Using Aggregation Method

For a production line with multiple stages, the state space is typically too large to directly perform analysis. Alternatively, the aggregation method is typically pursued to estimate the performance measures of a multistage line based on the analysis of all its subsystems. The aggregation method for the multistage line consists of the steady-state analysis and transient analysis, which are to be introduced in this section.

A. Steady-State Analysis

For a multistage line in the steady state, the state probability of each state keeps unchanged, the starvation probability and blockage probability of each machine become constant, and the expected performance measures do not vary with time. Given system parameters, the steady-state analysis is aimed at obtaining performance measures introduced in Section III for $t = \infty$.

When a multistage line is in the steady state, each subsystem, isolated from the multistage line, is also in the steady state. It means that the starvation probability p^s and the blockage probability p^b for any subsystem do not change with time. The steady-state probability $x(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}}, \infty)$ for all $(n, \tau_1, s_1^{\text{sub}}, s_2^{\text{sub}})$ can be obtained via transition equations such as (1). Thus, the performance measures $\widehat{PR}^{\text{sub}}(\infty)$, $\widehat{CR}^{\text{sub}}(\infty)$,

$\widehat{\mathrm{SR}}^{\mathrm{sub}}(\infty)$, $\widehat{WIP}^{\mathrm{sub}}(\infty)$, $\widehat{\mathrm{ST}}^{\mathrm{sub}}(\infty)$, and $\widehat{\mathrm{BL}}^{\mathrm{sub}}(\infty)$ can be calculated by (8)–(13), as shown at the bottom of this page. By the analysis of subsystems with $p_i^s(\infty)$ and $p_i^b(\infty)$ known, for $i = 1, \ldots, D$, $\widehat{PR}^{\mathrm{sub}}(\infty)$ of the $(i-1)$th subsystem can be the estimate of $PR_i(\infty)$ for $i = 2, \ldots, D$. $\widehat{CR}^{\mathrm{sub}}(\infty)$ of the first subsystem can be the estimate of $PR_1(\infty)$. Let $\mathrm{SR}_i(\infty)$ and $WIP_i(\infty)$ be $\widehat{\mathrm{SR}}^{\mathrm{sub}}(\infty)$ and $\widehat{WIP}^{\mathrm{sub}}(\infty)$ of the ith subsystem, respectively, for $i = 1, \ldots, D-1$. Then, $PR(\infty)$, $CR(\infty)$, $\mathrm{SR}(\infty)$, and $WIP(\infty)$ can be derived.

The aggregation method provides iterative procedures to estimate $p_i^s(\infty)$ and $p_i^b(\infty)$ for $i = 1, \ldots, D$, shown in Fig. 3. In each iteration, a backward aggregation and a forward aggregation are performed. We start with the first iteration. By Assumptions 7 and 8, we have $p_1^s(\infty) = 0$ and $p_D^b(\infty) = 0$. We set the initial $p_i^s(\infty)$ to be 0 for $i = 2, \ldots, D$ and initial $p_i^b(\infty)$ to be 0 for $i = 1, \ldots, D-1$.

1) *Backward Aggregation:* The first iteration starts from the backward aggregation, shown in Fig. 3(a). We first take machine m_{D-1}, machine m_D, and buffer B_{D-1} to form a subsystem. In the subsystem, the parameters of machine m_1^{sub}, machine m_1^{sub}, and buffer B are the same as the parameters of machine m_{D-1}, machine m_D, and buffer B_{D-1}, respectively. The values of $p_{D-1}^s(\infty)$ and $p_D^b(\infty)$ of the multistage line are assigned to p^s and p^b of the subsystem, respectively. With all the parameters for a subsystem ready, the steady-state performance measures of the subsystem can be obtained. The steady-state blockage probability $\widehat{\mathrm{BL}}^{\mathrm{sub}}(\infty)$ of the subsystem is used to update the value of $p_{D-1}^b(\infty)$ in the multistage line. After this step, a new multistage line is created with the number of machines reduced by one, the number of buffers reduced by 1, and the blockage probability $p_{D-1}^b(\infty)$ updated. Then, the process continues by taking machine m_{D-2}, machine m_{D-1}, and buffer B_{D-2} from the new multistage line to form a subsystem. Continue the process until the number of machines is reduced to be one and all the blockage probabilities $p_i^b(\infty)$, for $i = 1, \ldots, D-1$, are updated.
2) *Forward Aggregation:* Similar to the backward aggregation, the forward aggregation takes two machines and one buffer to form a subsystem but starts from the left side of the multistage line, shown in Fig. 3(b). We first take machine m_1, machine m_2, and buffer B_1 to form a subsystem. The parameters of machine m_1, machine m_2, and buffer B_1 of the multistage line are assigned to machine m_1^{sub}, machine m_2^{sub}, and buffer B of the subsystem, respectively. p^s and p^b of the subsystem are assigned the values of $p_1^s(\infty)$ and $p_2^b(\infty)$ of the multistage line, respectively. By performing analysis on the subsystem, we obtain the steady-state starvation probability $\widehat{\mathrm{ST}}^{\mathrm{sub}}(\infty)$, which is then used to replace $p_2^s(\infty)$ of the multistage line. After the step, a new multistage line is created with the number of machines reduced by one, the number of buffers reduced by one, and the starvation probability $p_2^s(\infty)$ updated. Continue the process until the number of machines is reduced to be one and all the starvation probabilities $p_i^s(\infty)$, for $i = 2, \ldots, D$, are updated.

An iteration is finished when both one backward aggregation and one forward aggregation are completed. The estimated

$$\widehat{PR}^{\mathrm{sub}}(t) = \left(1 - p^b\right) \sum_{n=1}^{N} \sum_{\tau_1=\max(n-1,T_{\min})}^{T_{\max}-1} \sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(n, \tau_1, s_1^{\mathrm{sub}}, 1, t\right) \tag{8}$$

$$\begin{aligned}\widehat{CR}^{\mathrm{sub}}(t) = \left(1 - p^s\right) \Bigg(& \sum_{s_2^{\mathrm{sub}}=0}^{1} x\left(0, 0, 1, s_2^{\mathrm{sub}}, t\right) + \sum_{n=1}^{N-1} \sum_{\tau_1=n-1}^{T_{\max}-1} \sum_{s_2^{\mathrm{sub}}=0}^{1} x\left(n, \tau_1, 1, s_2^{\mathrm{sub}}, t\right) \\ & + \left(1 - p^b\right) \sum_{\tau_1=\max(N-1,T_{\min})}^{T_{\max}-2} x(N, \tau_1, 1, 1, t) + \sum_{s_2^{\mathrm{sub}}=0}^{1} x\left(N, T_{\max} - 1, 1, s_2^{\mathrm{sub}}, t\right) \Bigg)\end{aligned} \tag{9}$$

$$\widehat{\mathrm{SR}}^{\mathrm{sub}}(t) = \sum_{n=1}^{N} \sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(n, T_{\max} - 1, s_1^{\mathrm{sub}}, 0, t\right) + p^b \sum_{n=1}^{N} \sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(n, T_{\max} - 1, s_1^{\mathrm{sub}}, 1, t\right) \tag{10}$$

$$\widehat{WIP}^{\mathrm{sub}}(t) = \sum_{n=1}^{N} \sum_{\tau_1=n-1}^{T_{\max}-1} \sum_{s_1^{\mathrm{sub}}=0}^{1} \sum_{s_2^{\mathrm{sub}}=0}^{1} n x\left(n, \tau_1, s_1^{\mathrm{sub}}, s_2^{\mathrm{sub}}, t\right) \tag{11}$$

$$\widehat{\mathrm{ST}}^{\mathrm{sub}}(t) = \begin{cases} \dfrac{\sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(0, 0, s_1^{\mathrm{sub}}, 1, t\right) + \sum_{n=1}^{\max(N,T_{\min})} \sum_{\tau_1=n-1}^{T_{\min}-1} \sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(n, \tau_1, s_1^{\mathrm{sub}}, 1, t\right)}{\sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(0, 0, s_1^{\mathrm{sub}}, 1, t\right) + \sum_{n=1}^{N} \sum_{\tau_1=n-1}^{T_{\max}-1} \sum_{s_1=0}^{1} x\left(n, \tau_1, s_1^{\mathrm{sub}}, 1, t\right)}, & \text{if } T_{\min} > 0 \\ \dfrac{\sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(0, 0, s_1^{\mathrm{sub}}, 1, t\right)}{\sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(0, 0, s_1^{\mathrm{sub}}, 1, t\right) + \sum_{n=1}^{N} \sum_{\tau_1=n-1}^{T_{\max}-1} \sum_{s_1^{\mathrm{sub}}=0}^{1} x\left(n, \tau_1, s_1^{\mathrm{sub}}, 1, t\right)}, & \text{if } T_{\min} = 0 \end{cases} \tag{12}$$

$$\widehat{\mathrm{BL}}^{\mathrm{sub}}(t) = \frac{\sum_{\tau_1=N-1}^{T_{\max}-2} x(N, \tau_1, 1, 0, t) + p^b \sum_{\tau_1=N-1}^{T_{\max}-2} x(N, \tau_1, 1, 1, t)}{\sum_{s_2=0}^{1} x(0, 0, 1, s_2, t) + \sum_{n=1}^{N} \sum_{\tau_1=n-1}^{T_{\max}-1} \sum_{s_2=0}^{1} x(n, \tau_1, 1, s_2, t)} \tag{13}$$

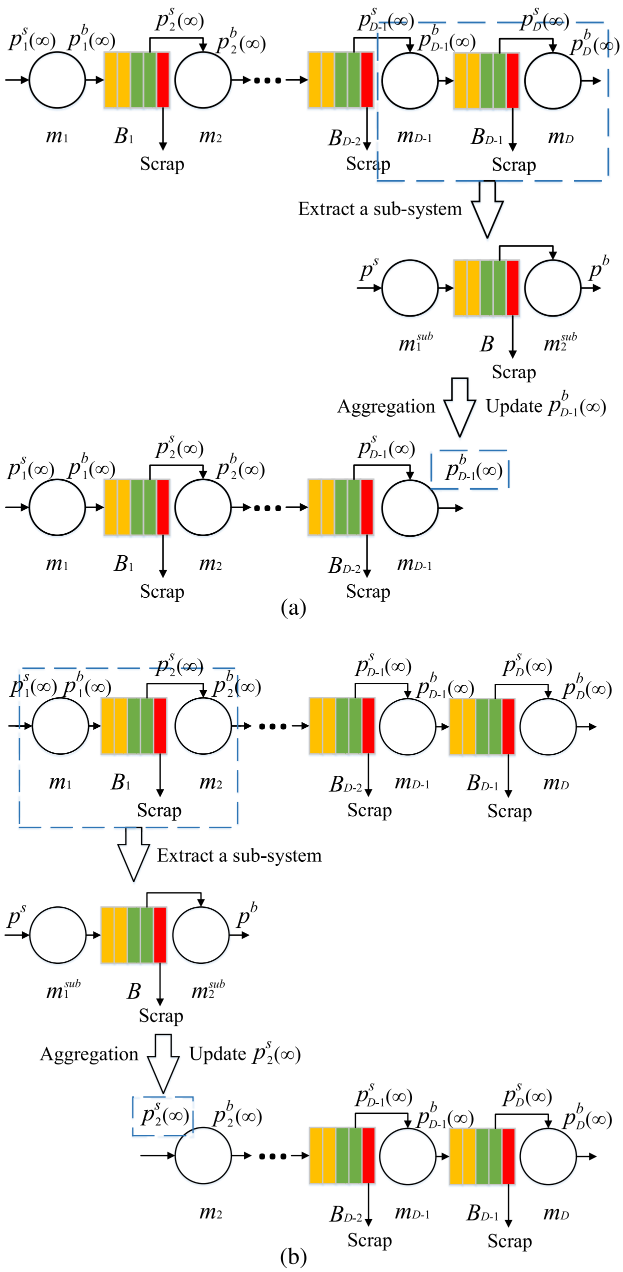


Fig. 3. Steady-state analysis of the aggregation method. (a) Backward aggregation. (b) Forward aggregation.

1: Initialize $p_i^s(\infty) = 0,\ i = 1, \cdots, D$
2: Initialize $p_i^b(\infty) = 0,\ i = 1, \cdots, D$
3: Determine the total number of iterations: $iter$
4: **for** $j = 1, iter$ **do** $\triangleright$ Iterations
5: **for** $k = 1, D-1$ **do** $\triangleright$ The backward aggregation
6: $l = D - k$ $\triangleright$ Set index
7: $m = D - k + 1$
8: $I = \left[p_l, r_l, p_l^s(\infty), p_m, r_m, p_m^b(\infty), T_{l,max}, T_{l,min}, N_l\right]$
9: $\left[\widehat{ST}^{sub}(\infty), \widehat{BL}^{sub}(\infty)\right] = getSubPerformance(I)$
10: $p_l^b(\infty) = \widehat{BL}^{sub}(\infty)$ $\triangleright$ Update blockage probability
11: **end for**
12: **for** $k = 1, D-1$ **do** $\triangleright$ The forward aggregation
13: $l = k$ $\triangleright$ Set index
14: $m = k + 1$
15: $I = \left(p_l, r_l, p_l^s(\infty), p_m, r_m, p_m^b(\infty), T_{l,max}, T_{l,min}, N_l\right)$
16: $\left[\widehat{ST}^{sub}(\infty), \widehat{BL}^{sub}(\infty)\right] = getSubPerformance(I)$
17: $p_m^s(\infty) = \widehat{ST}^{sub}(\infty)$ $\triangleright$ Update starvation probability
18: **end for**
19: **end for**

Fig. 4. Iterative procedures of the steady-state analysis.

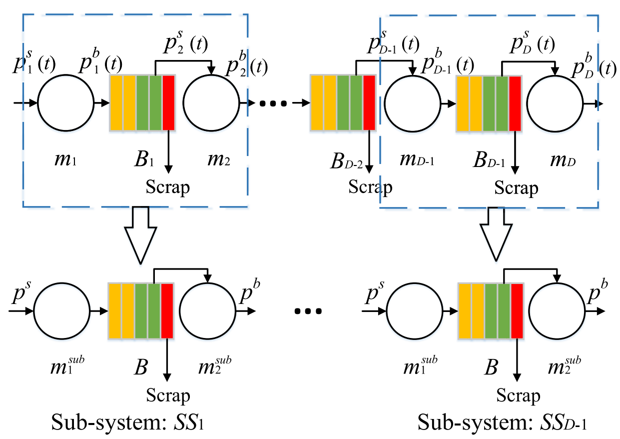


Fig. 5. Multistage line is decomposed into subsystems for transient analysis.

steady-state performance measures can be obtained after several iterations. The pseudocode for the aggregation method is shown in Fig. 4. Lines 1 and 2 are to initialize starvation probability and blockage probability. The iterative procedures of the aggregation method are represented by the loop from lines 4 to 19, among which the backward aggregation is given by the loop from lines 5 to 11 and the forward aggregation is given by the loop from lines 12 to 18. The function that appears in lines 9 and 16 transfers the parameters into the transition matrix by (1) and outputs starvation probability and blockage probability by (12) and (13), respectively.

B. Transient Analysis

With the system parameters and initial system state known, the transient analysis is aimed at obtaining transient performance measures introduced in Section III for $t \in \mathbb{N}^+$. To perform a transient analysis of a multistage line, we first decompose a multistage line into several subsystems. Fig. 5 shows the decomposition, where any two neighboring machines and the buffer between the two machines are isolated to form a subsystem. A multistage line with D machines and $(D-1)$ buffers is decomposed into $(D-1)$ subsystems. The subsystem that consists of machine m_i, machine m_{i+1}, and buffer B_i is denoted by SS_i, for $i = 1, \ldots, D-1$. In the transient analysis, the starvation probability p^s and the blockage probability p^b for a subsystem change over time, and each subsystem is modeled to be a time-varying Markov chain. The objective of the transient analysis is to capture the time-varying transition matrix of each subsystem over time so that the transient behavior of both subsystems and the whole multistage line can be predicted.

1: Initialize $X_i(1)$, $i = 1, \cdots, D-1$
2: Obtain $p_i^s(1)$, $i = 1, \cdots, D$
3: Obtain $p_i^b(1)$, $i = 1, \cdots, D$
4: Set the length of time: T
5: **for** $j = 1, T-1$ **do**
6: **for** $k = 1, D-1$ **do** ▷ Update state probability
7: $l = D - k$ ▷ Set index
8: $m = D - k + 1$
9: $I = \left[p_l, r_l, p_l^s(j), p_m, r_m, p_m^b(j), T_{l,max}, T_{l,min}, N_l\right]$
10: $Q_l(j) =$ *getTransition*(I)
11: $X_l(j+1) = X_l(j)Q_l(j)$
12: **end for**
13: **for** $k = 1, D-1$ **do** ▷ Update starvation and blockage probabilities
14: $l = D - k$ ▷ Set index
15: $m = D - k + 1$
16: Update $p_m^s(j+1)$ and $p_l^b(j+1)$ from $X_l(j+1)$
17: **end for**
18: **end for**

Fig. 6. Procedures of the transient analysis.

Let $X_i(t)$ denote the row vector of state probabilities of subsystem SS_i in cycle t, for $i = 1, \ldots, D-1$. We denote the transition matrix of subsystem SS_i in cycle t by $Q_i(t)$. The pseudocode for the transient analysis is shown in Fig. 6, which provides procedures to use the time-varying transition matrices of subsystems to perform a transient analysis of the multistage line. Given the initial state of a multistage line, the initial states of its subsystems are determined. It further determines the state probability $X_i(1)$ for $i = 1, \ldots, D-1$, which is initialized in line 1. With $X_i(1)$ for $i = 1, \ldots, D-1$ known, $p_i^s(1)$ and $p_i^b(1)$ for $i = 1, \ldots, D$ can be obtained. A loop from lines 5 to 18 is then to calculate the transient system state probability of the multistage line from cycle $t = 1$ to cycle $t = T$. There are two loops inside the loop. The first loop from lines 6 to 12 is to update the state probabilities of each subsystem, and the second loop from lines 13 to 17 is to update the starvation probabilities and the blockage probabilities. Line 10 is a function to transfer system parameters to the transition matrix of a subsystem. The update in line 16 can be achieved by (12) and (13). When the calculation for all the loops is completed, the state probabilities and performance measures of each subsystem (from SS_1 to SS_{D-1}) for each cycle (from $t = 1$ to $t = T$) are obtained. Then the transient performance measures of the entire multistage line are derived.

C. Comparison Between Steady-State and Transient Analysis

The steady-state analysis is aimed at long-term performance. When the system reaches steady state soon and stays in steady state for a long time, the production during the transient stage is negligible. The long-term performance obtained from the steady-state analysis can be used to estimate production capacity, make the production plan, and conduct continuous improvement. Transient analysis is required, when production operates partially or entirely in the transient regime for reasons such as long cycle time, disruptions, etc. The system performance is not stable in the transient stage and may be increasing, decreasing, or fluctuating in this stage. The transient analysis is aimed at capturing such dynamics. For a simple discrete-time Markov chain, system transition can be represented by a transition matrix. Steady-state analysis and transient analysis can be performed by manipulating the transition matrix. However, there is no single matrix that can model system transition for a multistage line, and thus the aggregation method is proposed to address the problem.

The procedure for steady-state analysis and the procedure for transient analysis are different from several aspects. First, it is assumed that there exist constant starvation probability, blockage probability, and steady-state probabilities in steady-state analysis, while in transient analysis those probabilities change over time. Second, starvation probability and blockage probability for each subsystem are unknown and initialized to zero in steady-state analysis, whereas the two probabilities are initially known from the initial system state in transient analysis. Third, the loop from lines 4 to 19 in Fig. 4 represents the iterative procedure for steady-state analysis, and it converges from performance measures under the initial setting to the performance measures in a steady state. The intermediate measures in the iterative procedure have no physical meaning. In contrast, the loop from lines 5 to 18 in Fig. 6 is to obtain transient behavior. For any j in the loop, transient starvation probability, blockage probability, state probabilities and performance measures for cycle $(j+1)$ are derived. Fourth, the number of iterations for both methods is different. The backward aggregation and forward aggregation are performed $iter$ times in steady-state analysis, while there is only one backward aggregation in transient analysis. Finally, the procedure of steady-state analysis cannot analyze the transient behavior of a multistage line. In contrast, transient analysis can be used to obtain steady-state performance measures by running for a sufficiently large number of cycles as the system reaches a steady state, but such a way is not computationally efficient.

TABLE I
PARAMETER SETTING FOR ILLUSTRATIVE EXAMPLE

i	1	2	3	4	5	6	7	8
e_i	0.62	0.75	0.85	0.73	0.68	0.91	0.81	0.72
r_i	0.35	0.44	0.37	0.24	0.27	0.35	0.29	0.46
N_i	5	6	7	7	7	5	6	6
$T_{i,max}$	8	8	10	10	8	7	9	9
$T_{i,min}$	1	1	2	2	1	1	2	1

VI. Model Validation

A. Illustrative Example

To evaluate the accuracy of the proposed analytical method, we compare the results obtained from the proposed analytical method with simulation. A MATLAB program is constructed to conduct a numerical experiment. We first consider a single case with the parameters shown in Table I.

Initially, all the machines are set to be up, and all the buffers are set to be empty. Both steady-state analysis and transient

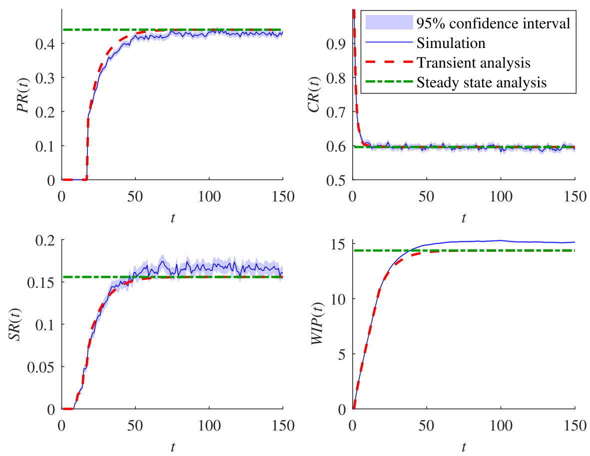


Fig. 7. Comparison of performance measures from the aggregation approach and simulation.

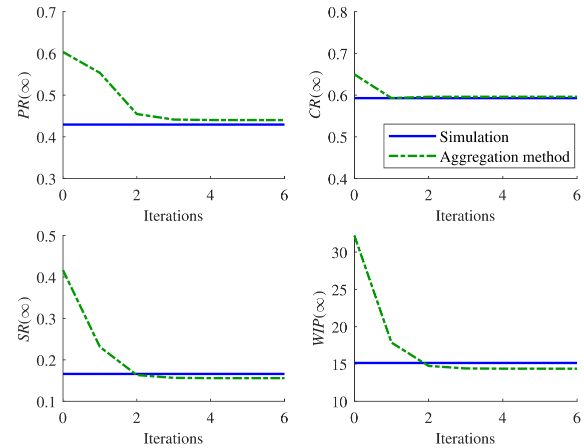


Fig. 8. Steady-state performance measures estimated in each iteration of the aggregation method.

analysis of the aggregation method are performed. For the steady-state analysis of the aggregation method, six iterations are conducted to get the steady-state performance measures. Simulation repeats 10 000 times to obtain the average value and 95% confidence interval of each performance measure in each cycle. The result of the numerical study is shown in Fig. 7. Simulated performance measures are plotted in blue solid line, with the shaded area indicating the 95% confidence interval. The red dashed lines represent the transient performance measures obtained from the transient analysis of the aggregation method. The green dashed-dotted lines represent the steady-state performance measures obtained from a steady-state analysis of the aggregation method. The result of the experiment suggests that the proposed aggregation method can capture both the steady-state and transient behaviors of the multistage line accurately.

As is shown in Fig. 8, the blue solid lines represent the steady-state performance measures obtained from the simulation, while the green dashed-dotted lines represent the estimated performance measures after each iteration of the aggregation method. The initial starvation probability and blockage probability for each machine of the multistage line are set to be 0, and the estimated performance measures in iteration 0 in Fig. 8 represent the estimated performance measures with the initial parameter setting. It is suggested that the convergence can be achieved usually within three iterations, and converging performance measures are close to the true values.

B. Experiment With Random Parameters

To evaluate the accuracy of the proposed method in a more general sense, the experiment with random parameters is conducted. We compare the estimated performance measures obtained through the aggregation method with the ones estimated by the simulation. Let T_{st} be the threshold of time where one can guarantee that the system in the simulation study can reach the steady state, and let T denote the run length of the simulation. We denote by $PM(t)$ the true performance measures, like $PR(t)$, $CR(t)$, $\text{SR}(t)$, and $WIP(t)$, and they represented by average performance measures of all repeats of the simulation in cycle t. The true steady-state performance is obtained as follows:

$$PM(\infty) = \frac{1}{T - T_{\text{st}} + 1} \sum_{i=T_{\text{st}}}^{T} PM(i). \tag{14}$$

Denote the steady-state performance measures obtained through the aggregation method by $\widehat{PR}(\infty)$, $\widehat{CR}(\infty)$, $\widehat{\text{SR}}(\infty)$, and $\widehat{WIP}(\infty)$. The absolute errors of the estimated steady-state performance measures, denoted by $\epsilon_{PR(\infty)}$, $\epsilon_{CR(\infty)}$, $\epsilon_{\text{SR}(\infty)}$ and $\epsilon_{WIP(\infty)}$, are provided as follows:

$$\epsilon_{PM(\infty)} = |PM(\infty) - \widehat{PM}(\infty)| \tag{15}$$

where $\widehat{PM}(\infty)$ represents $\widehat{PR}(\infty)$, $\widehat{CR}(\infty)$, $\widehat{\text{SR}}(\infty)$, and $\widehat{WIP}(\infty)$, and $\epsilon_{PM(\infty)}$ represents $\epsilon_{PR(\infty)}$, $\epsilon_{CR(\infty)}$, $\epsilon_{\text{SR}(\infty)}$, and $\epsilon_{WIP(\infty)}$. The relative errors of the estimated steady-state performance measures are denoted by $\delta_{PR(\infty)}$, $\delta_{CR(\infty)}$, $\delta_{\text{SR}(\infty)}$, and $\delta_{WIP(\infty)}$. Let $\delta_{PM(\infty)}$ stand for $\delta_{PR(\infty)}$, $\delta_{CR(\infty)}$, $\delta_{\text{SR}(\infty)}$, and $\delta_{WIP(\infty)}$. The relative errors are defined as follows:

$$\delta_{PM(\infty)} = \frac{\epsilon_{PM(\infty)}}{PM(\infty)}. \tag{16}$$

Let T_{tr} be the time before which the transient behaviors of the aggregation method and the simulation are compared. The absolute errors of the estimated transient performance measures are denoted by $\epsilon_{PR(t)}$, $\epsilon_{CR(t)}$, $\epsilon_{\text{SR}(t)}$ and $\epsilon_{WIP(t)}$. Let $\epsilon_{PM(t)}$ stand for $\epsilon_{PR(t)}$, $\epsilon_{CR(t)}$, $\epsilon_{\text{SR}(t)}$ and $\epsilon_{WIP(t)}$. The absolute errors are defined as follows:

$$\epsilon_{PM(t)} = \frac{\sum_{i=1}^{T_{tr}} |PM(i) - \widehat{PM}(i)|}{T_{tr}}. \tag{17}$$

The relative errors of the estimated transient performance measures are denoted by $\delta_{PR(t)}$, $\delta_{CR(t)}$, $\delta_{\text{SR}(t)}$ and $\delta_{WIP(t)}$. Let $\delta_{PM(t)}$ stand for $\delta_{PR(t)}$, $\delta_{CR(t)}$, $\delta_{\text{SR}(t)}$ and $\delta_{WIP(t)}$. The relative errors are defined as follows:

$$\delta_{PM(t)} = \frac{\epsilon_{PM(t)}}{PM(\infty)}. \tag{18}$$

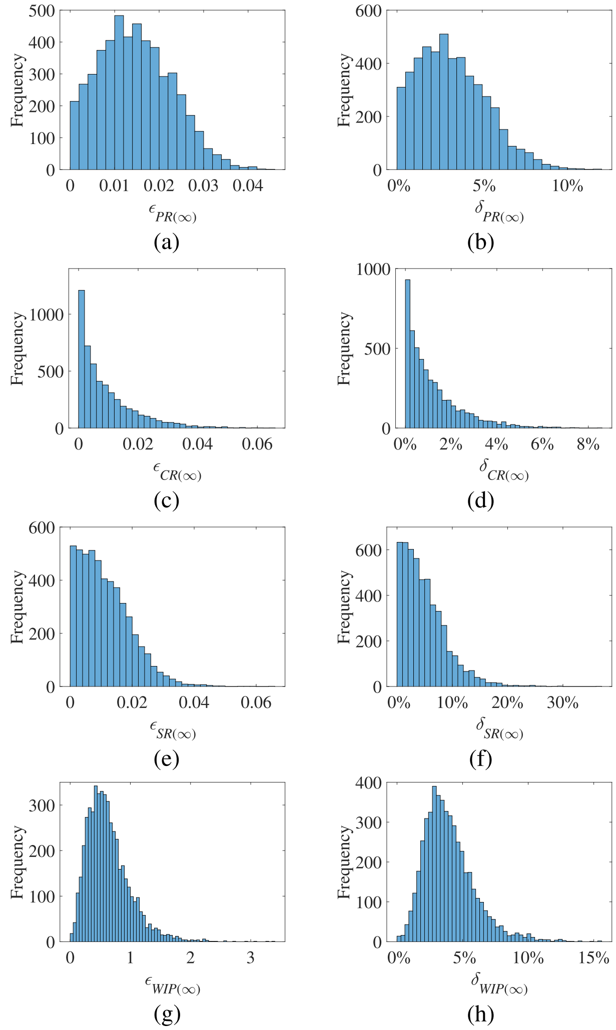


Fig. 9. Accuracy of the steady-state analysis. (a) Absolute error of overall production rate. (b) Relative error of overall production rate. (c) Absolute error of overall consumption rate. (d) Relative error of overall consumption rate. (e) Absolute error of overall scrap rate. (f) Relative error of overall scrap rate. (g) Absolute error of overall work-in-process. (h) Relative error of overall work-in-process.

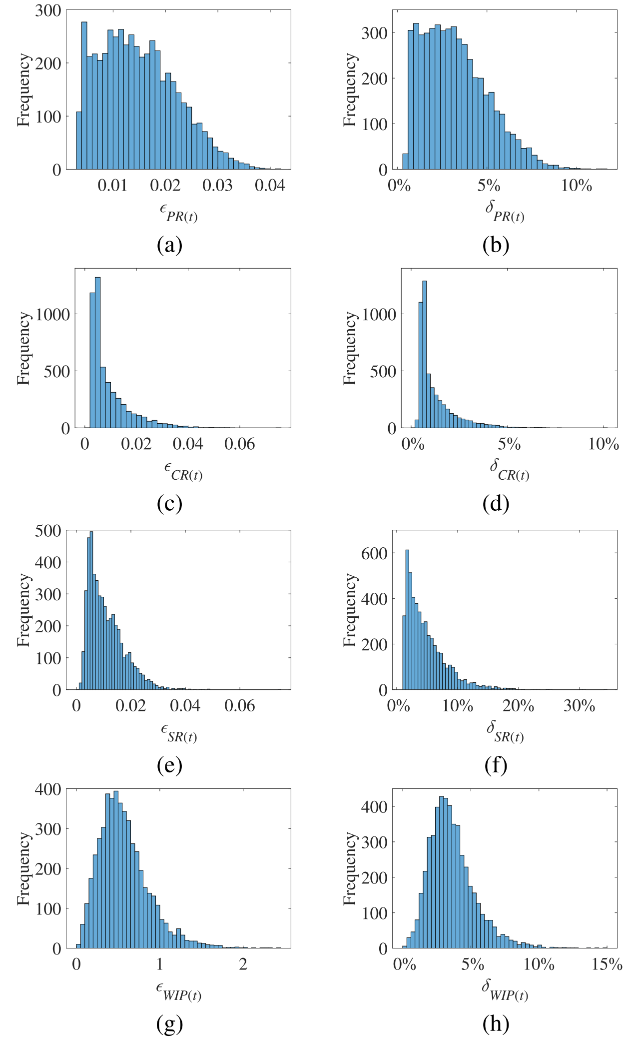


Fig. 10. Accuracy of the transient analysis. (a) Absolute error of overall production rate. (b) Relative error of overall production rate. (c) Absolute error of overall consumption rate. (d) Relative error of overall consumption rate. (e) Absolute error of overall scrap rate. (f) Relative error of overall scrap rate. (g) Absolute error of overall work-in-process. (h) Relative error of overall work-in-process.

To test the aggregation method in both steady-state analysis and transient analysis, the parameter settings are randomly selected from the range shown as follows:

$$\begin{aligned} D &\in \{4, 5, 6, 7, 8, 9, 10\} \\ e_i &\in [0.60, 0.99], \quad \text{for } i = 1, \ldots, D \\ r_i &\in [0.20, 0.50], \quad \text{for } i = 1, \ldots, D \\ N_i &\in \{5, 6, 7\} \quad \text{for } i = 1, \ldots, D-1 \\ T_{i,\max} &\in \{N_i + 1, N_i + 2, N_i + 3\} \quad \text{for } i = 1, \ldots, D-1 \\ T_{i,\min} &\in \{1, 2\}, \quad \text{for } i = 1, \ldots, D-1. \end{aligned} \tag{19}$$

The number of machines is selected from the range $\{4, 5, 6, 7, 8, 9, 10\}$, and this can cover a large number of applications. Machine efficiency is commonly seen in the range of $[0.60, 0.99]$, from which the machine efficiency is randomly chosen in the experiment. The selection of N and $T_{i,\min}$ and $T_{i,\max}$ covers a large portion of applications. The steady-state performance of a production system is worth analyzing, when the buffer is capable of supporting a smooth production, and the majority of parts are finally produced with an acceptable quality. Thus, the repair probability is selected from $[0.20, 0.50]$. For the experiments of steady-state analysis, 5000 random parameter settings are generated. In each parameter setting, the run length is $T = 100\,000$, and 40 runs are carried out. The cycles after $T_{\text{st}} = 20\,001$ are considered in the steady-state analysis. In the transient analysis, 5000 parameter settings are randomly selected. In each parameter setting, the simulation runs $T = 2000$ cycles and repeats $10\,000$ times. In addition, we set $T_{\text{st}} = 1001$ and $T_{\text{tr}} = 400$.

The accuracy of the steady-state analysis and transient analysis is shown in Figs. 9 and 10, respectively. The overall consumption rate can be estimated accurately. The median of the relative error of the overall consumption rate is 0.81% in both steady-state analysis and transient analysis, and the relative errors are less than 2.0% for most cases. The estimates of the overall production rate and work-in-process have higher errors. In most cases, it can maintain relative errors smaller than 6%, which is also acceptable. The overall scrap rate has the largest error. The median of relative error is 4.1% for steady-state analysis and 3.9% for transient analysis. The absolute error of the scrap rate is small, and the large relative error is partially due to the small denominator. The experiment with random parameters suggests that the proposed analytical method, combining the approximate modeling of residence time and the aggregation method, can estimate performance measures of a multistage line in high accuracy.

Both the simulation and the aggregation method are developed with MATLAB and run on a computer with Intel(R) Core(TM) i7-8700 CPU, 16-GB RAM, and 64-bit Windows 10 Enterprise operating system. The average time to perform simulation for a parameter setting in the steady-state analysis is 39.63 s, and it takes only 0.13 s on average to perform steady-state analysis of the aggregation method. It shows that the aggregation method is more efficient. In the transient analysis, the simulation takes 219.11 s on average for each parameter setting, and the aggregation method takes 27.36 s. The transient analysis of the aggregation method takes more time than the steady-state analysis, but it is still more efficient than the simulation.

VII. Conclusion and Future Work

In this article, we introduce a novel modeling approach for both transient and steady-state performance evaluation of multistage geometric serial production lines with residence time limits. Specifically, to deal with the complexity of such systems, a two-machine-one-buffer subsystem is defined, and a Markov chain model is developed to analyze the subsystem. The approximate modeling of residence time is applied to the analysis of the two-machine-one-buffer subsystem to further reduce the size of state space. The aggregation method is then developed to evaluate the performance of a multistage geometric serial line with residence time limits. Compared with simulation, the proposed aggregation method is verified to possess high accuracy in evaluating both steady-state and transient performance. Such a method can provide production engineers a quantitative tool to evaluate complex production systems efficiently and accurately. Future work can be directed to investigating real-time control policies that optimize such a complex system by leveraging the developed modeling approach in this article.

Appendix
Remaining Transition Equations in Section IV-B

We start with the simple state $(0, 0, s_1^{\text{sub}}, s_2^{\text{sub}})$ in the $(t+1)$-th cycle, for $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$. It represents the system state that the buffer is empty and the states for both machines are s_1^{sub} and s_2^{sub}, respectively. Transitions regarding state $(0, 0, s_1^{\text{sub}}, s_2^{\text{sub}})$ in cycle $t+1$ can be obtained using the following equation:

$$
\begin{aligned}
x\big(0, 0, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\big)
&= x(0,0,0,1,t) P^{(1)}_{0,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + x(0,0,1,1,t) p^s P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + x(0,0,0,0,t) P^{(1)}_{0,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(0,0,1,0,t) p^s P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + \sum_{\tau_1=T_{\min}}^{T_{\max}-2} x(1,\tau_1,0,1,t)\big(1-p^b\big) P^{(1)}_{0,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + \sum_{\tau_1=T_{\min}}^{T_{\max}-2} x(1,\tau_1,1,1,t) p^s \big(1-p^b\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,0,0,t) P^{(1)}_{0,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,1,0,t) p^s P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,0,1,t) P^{(1)}_{0,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,1,1,t) p^s P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}}
\end{aligned} \tag{20}
$$

for $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$.

State $(1, 0, s_1^{\text{sub}}, s_2^{\text{sub}})$ in cycle $t+1$, for $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$, represents the state that there is one part in buffer B and its residence time is 0. It implies the part is produced by machine m_1^{sub} at the end of cycle t. Thus, machine m_1^{sub} must be up in cycle t, and starvation does not happen to machine m_1^{sub}. The system evolution for $x(1, 0, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1)$ can be represented as

$$
\begin{aligned}
x\big(1, 0, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\big)
&= x(0,0,1,0,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(0,0,1,1,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + \sum_{\tau_1=T_{\min}}^{T_{\max}-2} x(1,\tau_1,1,1,t)\big(1-p^s\big)\big(1-p^b\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,1,0,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(1,T_{\max}-1,1,1,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}}
\end{aligned} \tag{21}
$$

for $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$.

The rest of the transitions are shown in the following:

$$
\begin{aligned}
x\big(j, j-1, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\big)
&= x(j-1,j-2,1,0,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{0,s_2^{\text{sub}}} \\
&\quad + x(j-1,j-2,1,1,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} \\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}} \mathbf{1}_{\mathbb{N}^+}(T_{\min}+2-j) \\
&\quad + x(j-1,j-2,1,1,t)\big(1-p^s\big) p^b \\
&\quad \times P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \mathbf{1}_{\mathbb{N}^+}(j-1-T_{\min}) \\
&\quad + \sum_{i=\max(j-1,T_{\min})}^{T_{\max}-2} x(j,i,1,1,t)\big(1-p^s\big)\big(1-p^b\big) \\
&\quad \times P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \Phi(j,i,j-2) \\
&\quad + x(j,T_{\max}-1,1,0,t)\big(1-p^s\big) P^{(1)}_{1,s_1^{\text{sub}}} \\
&\quad \times P^{(2)}_{0,s_2^{\text{sub}}} \Phi(j,T_{\max}-1,j-2) \\
&\quad + x(j,T_{\max}-1,1,1,t)\big(1-p^s\big) \\
&\quad \times P^{(1)}_{1,s_1^{\text{sub}}} P^{(2)}_{1,s_2^{\text{sub}}} \Phi(j,T_{\max}-1,j-2)
\end{aligned} \tag{22}
$$

for $2 \le j \le N$, and $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$

$$
\begin{aligned}
&x\big(j, i, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\big)\\
&= x(j-1, i-1, 1, 0, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(j-1, i-1, 1, 1, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(j-1, i-1, 1, 1, t)\big(1-p^s\big)p^b P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})\\
&\quad + x(j, i-1, 0, 0, t)P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(j, i-1, 0, 1, t)P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(j, i-1, 0, 1, t)p^b P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})\\
&\quad + x(j, i-1, 1, 0, t)p^s P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(j, i-1, 1, 1, t)p^s P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(j, i-1, 1, 1, t)p^s p^b P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})\\
&\quad + \sum_{k=\max(i,T_{\min})}^{T_{\max}-2} x(j, k, 1, 1, t)\big(1-p^s\big)\big(1-p^b\big)\\
&\quad \times P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j, k, i-1)\\
&\quad + x(j, T_{\max}-1, 1, 0, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{0,s_2^{\text{sub}}}\Phi(j, T_{\max}-1, i-1)\\
&\quad + x(j, T_{\max}-1, 1, 1, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j, T_{\max}-1, i-1)\\
&\quad + \sum_{k=\max(i,T_{\min})}^{T_{\max}-2} x(j+1, k, 0, 1, t)\big(1-p^b\big)\\
&\quad \times P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j+1, k, i-1)\\
&\quad + \sum_{k=\max(i,T_{\min})}^{T_{\max}-2} x(j+1, k, 1, 1, t)p^s\big(1-p^b\big)\\
&\quad \times P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j+1, k, i-1)\\
&\quad + x(j+1, T_{\max}-1, 0, 0, t)P^{(1)}_{0,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{0,s_2^{\text{sub}}}\Phi(j+1, T_{\max}-1, i-1)\\
&\quad + x(j+1, T_{\max}-1, 0, 1, t)P^{(1)}_{0,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j+1, T_{\max}-1, i-1)\\
&\quad + x(j+1, T_{\max}-1, 1, 0, t)p^s P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{0,s_2^{\text{sub}}}\Phi(j+1, T_{\max}-1, i-1)\\
&\quad + x(j+1, T_{\max}-1, 1, 1, t)p^s P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}}\Phi(j+1, T_{\max}-1, i-1)
\end{aligned} \tag{23}
$$

for $2 \le j \le N-1$, $j \le i \le T_{\max}-1$, and $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$

$$
\begin{aligned}
&x\big(N, i, s_1^{\text{sub}}, s_2^{\text{sub}}, t+1\big)\\
&= x(N-1, i-1, 1, 0, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(N-1, i-1, 1, 1, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(N-1, i-1, 1, 1, t)\big(1-p^s\big)p^b P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})\\
&\quad + x(N, i-1, 0, 0, t)P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(N, i-1, 0, 1, t)P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(N, i-1, 0, 1, t)p^b P^{(1)}_{0,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})\\
&\quad + \sum_{j=\max(i,T_{\min})}^{T_{\max}-2} x(N, j, 1, 1, t)\big(1-p^s\big)\big(1-p^b\big)P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}}\Phi(N, j, i-1)\\
&\quad + x(N, T_{\max}-1, 1, 0, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{0,s_2^{\text{sub}}}\Phi(N, T_{\max}-1, i-1)\\
&\quad + x(N, T_{\max}-1, 1, 1, t)\big(1-p^s\big)P^{(1)}_{1,s_1^{\text{sub}}}\\
&\quad \times P^{(2)}_{1,s_2^{\text{sub}}}\Phi(N, T_{\max}-1, i-1)\\
&\quad + x(N, i-1, 1, 0, t)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{0,s_2^{\text{sub}}}\\
&\quad + x(N, i-1, 1, 1, t)P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(T_{\min}+1-i)\\
&\quad + x(N, i-1, 1, 1, t)p^b P^{(1)}_{1,s_1^{\text{sub}}}P^{(2)}_{1,s_2^{\text{sub}}}\mathbf{1}_{\mathbb{N}^+}(i-T_{\min})
\end{aligned} \tag{24}
$$

for $N \le i \le T_{\max}-1$, and $s_1^{\text{sub}}, s_2^{\text{sub}} = 0, 1$.

REFERENCES

[1] Y. Li, F. Chu, Z. Yang, and R. W. Calvo, "A production inventory routing planning for perishable food with quality consideration," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 407–412, 2016.

[2] J. Wang, Y. Hu, and J. Li, "Transient analysis to design buffer capacity in dairy filling and packing production lines," *J. Food Eng.*, vol. 98, no. 1, pp. 1–12, May 2010.

[3] P. Amorim, H. Meyr, C. Almeder, and B. Almada-Lobo, "Managing perishability in production-distribution planning: A discussion and review," *Flexible Services Manuf. J.*, vol. 25, no. 3, pp. 389–413, Sep. 2013.

[4] G. Liberopoulos and P. Tsarouhas, "Reliability analysis of an automated pizza production line," *J. Food Eng.*, vol. 69, no. 1, pp. 79–96, Jul. 2005.

[5] S. Rostami, B. Hamidzadeh, and D. Camporese, "An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 609–618, Oct. 2001.

[6] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[7] C. Pan, M. Zhou, Y. Qiao, and N. Wu, "Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 586–601, Apr. 2018.

[8] N. Kang, F. Ju, and L. Zheng, "Transient analysis of geometric serial lines with perishable intermediate products," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 149–156, Jan. 2017.

[9] Y. Lu and F. Ju, "Smart manufacturing systems based on cyber-physical manufacturing services (CPMS)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15883–15889, Jul. 2017.

[10] F. Wang, F. Ju, and Y. Lu, "A study on performance evaluation and status-based decision for cyber-physical production systems," in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Aug. 2017, pp. 1000–1005.

[11] P. Stephan, G. Meixner, H. Koessling, F. Floerchinger, and L. Ollinger, "Product-mediated communication through digital object memories in heterogeneous value chains," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2010, pp. 199–207.

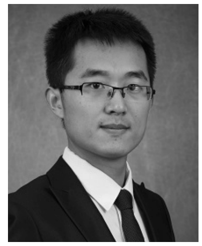
[12] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards Industry 4.0—Standardization as the crucial challenge for highly modular, multi-vendor production systems," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 579–584, 2015.

[13] F. Wang, F. Ju, and N. Kang, "Transient analysis and real-time control of geometric serial lines with residence time constraints," *IISE Trans.*, vol. 51, no. 7, pp. 709–728, Jul. 2019.

[14] F. Ju, J. Li, and J. A. Horst, "Transient analysis of serial production lines with perishable products: Bernoulli reliability model," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 694–707, Feb. 2017.

[15] F. Ju, J. Li, and J. Horst, "Transient analysis of Bernoulli serial line with perishable products," in *Proc. IFAC Symp. Inf. Control Problems Manuf.*, 2015, pp. 1670–1675.

[16] G. Chen, C. Wang, L. Zhang, J. Arinez, and G. Xiao, "Transient performance analysis of serial production lines with geometric machines," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 877–891, Apr. 2016.
[17] J. Pahl and S. Voß, "Integrating deterioration and lifetime constraints in production and supply chain planning: A survey," *Eur. J. Oper. Res.*, vol. 238, no. 3, pp. 654–674, Nov. 2014.
[18] M. B. Dumas and M. Rabinowitz, "Policies for reducing blood wastage in hospital blood banks," *Manage. Sci.*, vol. 23, no. 10, pp. 1124–1132, Jun. 1977.
[19] E. Brodheim and G. P. Prastacos, "The long island blood distribution system as a prototype for regional blood management," *Interfaces*, vol. 9, no. 5, pp. 3–20, Nov. 1979.
[20] J. Beliën and H. Forcé, "Supply chain management of blood products: A literature review," *Eur. J. Oper. Res.*, vol. 217, no. 1, pp. 1–16, Feb. 2012.
[21] H.-M. Wee, "Economic production lot size model for deteriorating items with partial back-ordering," *Comput. Ind. Eng.*, vol. 24, no. 3, pp. 449–458, Jul. 1993.
[22] J. Li and S. M. Meerkov, *Production Systems Engineering*. New York, NY, USA: Springer, 2008.
[23] S. B. Gershwin, *Manufacturing Systems Engineering*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.
[24] F. Wang, Y. Lu, and F. Ju, "Condition-based real-time production control for smart manufacturing systems," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 1052–1057.
[25] C. Shi and S. B. Gershwin, "Part waiting time distribution in a two-machine line," *IFAC Proc. Volumes*, vol. 45, no. 6, pp. 457–462, May 2012.
[26] A. Angius, M. Colledani, A. Horváth, and S. B. Gershwin, "Analysis of the lead time distribution in closed loop manufacturing systems," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 307–312, 2016.
[27] C. Shi and S. B. Gershwin, "Part sojourn time distribution in a two-machine line," *Eur. J. Oper. Res.*, vol. 248, no. 1, pp. 146–158, Jan. 2016.
[28] R. Naebulharam and L. Zhang, "Bernoulli serial lines with deteriorating product quality: Performance evaluation and system-theoretic properties," *Int. J. Prod. Res.*, vol. 52, no. 5, pp. 1479–1494, Mar. 2014.
[29] L. Zhang, C. Wang, J. Arinez, and S. Biller, "Transient analysis of Bernoulli serial lines: Performance evaluation and system-theoretic properties," *IIE Trans.*, vol. 45, no. 5, pp. 528–543, May 2013.
[30] J.-H. Lee and J. Li, "Performance evaluation of Bernoulli serial lines with waiting time constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1087–1092, Jul. 2017.
[31] J.-H. Lee, J. Li, and J. A. Horst, "Serial production lines with waiting time limits: Bernoulli reliability model," *IEEE Trans. Eng. Manag.*, vol. 65, no. 2, pp. 316–329, May 2018.



Feifan Wang (Student Member, IEEE) received the bachelor's degree from the Department of Industrial Engineering, Zhejiang University of Technology, Hangzhou, China, and the master's degree from the Department of Industrial and Systems Engineering, Zhejiang University, Hangzhou, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA.

His research interests include the modeling, analysis, and control of production systems.

Feng Ju (Member, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the M.S. degree in electrical and computer engineering, and the Ph.D. degree in industrial and systems engineering from the University of Wisconsin, Madison, WI, USA, in 2011 and 2015, respectively.

He is a member of the Institute for Operations Research and the Management Sciences, the Institute of Industrial and Systems Engineers, and the Institute of Electrical and Electronics Engineers. He is currently an Assistant Professor with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA. His research interests include modeling, analysis, continuous improvement, and optimization of manufacturing systems.

Dr. Ju was a recipient of multiple awards, including the Best Paper Award in IFAC MIM and a Best Student Paper Finalist in the IEEE CASE and IFAC INCOM.