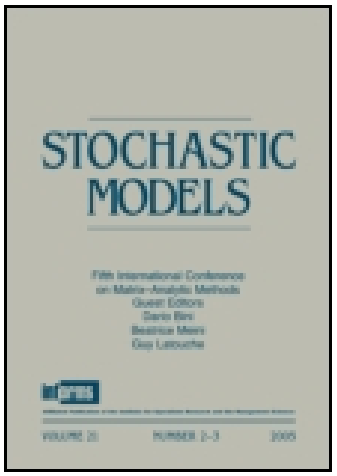
Performance analysis of exponential multi-server production lines with fluid flow and finite buffers



Stochastic Models
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/lstm20

Performance Analysis of Exponential Multi-Server Production Lines with Fluid Flow and Finite Buffers

Stijn Fleuren[a], Remco Bierbooms[b] & Ivo Adan[a]
[a] Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
[b] Lime B.V., Eindhoven, The Netherlands
Published online: 06 Nov 2014.

To cite this article: Stijn Fleuren, Remco Bierbooms & Ivo Adan (2014) Performance Analysis of Exponential Multi-Server Production Lines with Fluid Flow and Finite Buffers, Stochastic Models, 30:4, 469-493, DOI: 10.1080/15326349.2014.959183

To link to this article: http://dx.doi.org/10.1080/15326349.2014.959183

Stochastic Models, 30:469–493, 2014

ISSN: 1532-6349 print / 1532-4214 online
DOI: 10.1080/15326349.2014.959183

Taylor & Francis
Taylor & Francis Group

PERFORMANCE ANALYSIS OF EXPONENTIAL MULTI-SERVER PRODUCTION LINES WITH FLUID FLOW AND FINITE BUFFERS

Stijn Fleuren,[1] Remco Bierbooms,[2] and Ivo Adan[1]
[1]*Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands*
[2]*Lime B.V., Eindhoven, The Netherlands*

□ *This article presents an approximation method for fluid flow production lines with multi-server workstations and finite buffers. Each workstation consists of parallel identical servers, which are subject to operation-dependent failures with exponentially distributed uptimes and downtimes. The proposed method decomposes the production line into single-buffer subsystems, each described by a continuous state Markov process, the parameters of which are determined iteratively. The approximation method is appropriate for the analysis of longer production lines, able to accurately estimate performance characteristics (e.g., throughput and mean buffer content), and shown to perform well on a large test set.*

Keywords Approximate analysis; Multi-server production lines; Operation dependent failures; Decomposition technique.

Mathematics Subject Classification 90B22.

1. INTRODUCTION

This article studies production lines with multi-server workstations in series and finite buffers in between. Each workstation consists of parallel identical servers. The flow through the workstations is assumed to be continuous, that is, fluid instead of discrete items. Figure 1 illustrates a fluid-flow multi-server production line, labeled L, with four workstations $W_1, \ldots, W_4$ in series, where W_i denotes the ith workstation.

The downstream buffer of W_i is denoted by B_i, the size of which is b_i. Each workstation W_i has s_i identical parallel servers, which are subject to operation dependent failures. This means that servers can only break down when actually producing. Each server in W_i breaks down at an exponential rate λ_i (when producing), and it is repaired at an exponential rate μ_i. The

Received March 2014; Accepted August 2014
Address correspondence to Ivo Adan, Department of Mechanical Engineering, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands; E-mail: iadan@win.nl

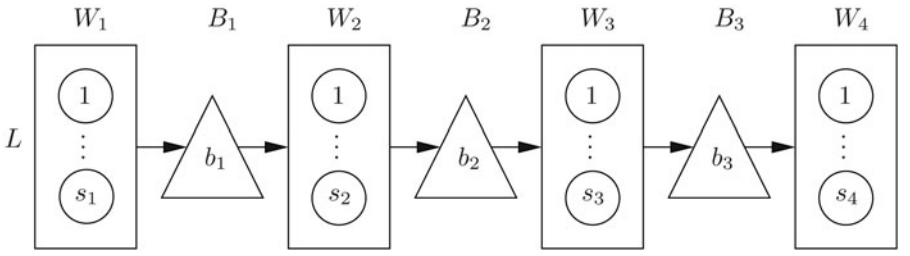


FIGURE 1 A multi-server production line with four workstations.

speed of a single server in workstation W_i is v_i/s_i, and thus the maximum speed at which workstation W_i can process fluid (when all servers are up) is v_i. When the downstream buffer B_i becomes full, workstation W_i slows down to the production rate of W_{i+1} or becomes blocked if W_{i+1} is not producing. Similarly, when upstream buffer B_{i-1} becomes empty, it slows down to the production rate of W_{i-1}, or becomes starved if W_{i-1} is not producing.

Fluid flow models are natural models for production systems producing continuous material but also for high-volume discrete production systems, as seen in, for example, back-end semiconductor manufacturing. Actually, the present study is motivated by the design of a production line for small appliances, consisting of six workstations, where the second and third one consist of multiple identical machines; see also the next section. One of the issues in the design of this production line are the required buffer sizes in between the workstations in order to meet the target throughput.

An exact analysis of the fluid flow models proposed in this article is intractable. Hence, we aim at finding an analytical method to efficiently approximate the performance of fluid flow models, such as throughput and mean buffer content. In the literature, approximations have been developed for single-server fluid flow lines, based on aggregation[14,15], decomposition[17,12,22,4,11,18,6,5,3], and homogenization[13,16,19]. Recently, a decomposition-based approximation for fluid flow lines has been proposed in Ref.[5]. The model in Ref.[5] includes multi-server lines. An alternative approach for multi-server lines can be found in Refs.[20,4], proposing to replace multi-server workstations with equivalent single-server workstations and thus making the model suitable for application of single-server methods.

The approximation method in this article is based on decomposition and aggregation: the production line is decomposed into single-buffer subsystems, each of which is composed of an *arrival* workstation, aggregating the upstream part of production line, and a *departure* workstation, aggregating the downstream part. The parameters of the subsystems are then determined iteratively. Use of aggregation is crucial to keep the (size of the) subsystems manageable. A single-buffer subsystem can be described by a continuous-time continuous-state Markov chain, the steady-state distribution of which satisfies a set of linear differential equations with constant coefficients. The

solution of this set of linear differential equations can be expressed as a matrix exponential function or alternatively, in terms of the eigenvalues and eigenvectors of the matrix exponent. However, both expressions may be numerically unstable, severely limiting their applicability, cf. Section 4.3 in Ref.[3]. In this article, we avoid these numerical issues by using the numerically stable matrix-analytic methods for fluid flow models developed in Ref.[10,9], which have been inspired by earlier work in Ref.[21] exploring the similarity between fluid flow models and discrete-state quasi-birth-and-death processes.

The decomposition-aggregation technique in Ref.[3] has been developed for the approximate analysis of single-server fluid flow lines with generally distributed up- and downtimes. This technique, however, cannot be directly applied to the exponential multi-server lines considered in the current article, since multi-server stations have *state-dependent* uptimes and production speeds (i.e., depending on the number of servers being up). As mentioned above, the approximation method in Ref.[5], also applies to exponential multi-server lines. The main difference between this method and the current one is in the level of detail at which subsystems are modeled. In Ref.[5] the state space of the detailed Markov model of a subsystem grows almost exponentially in the length of the production line, whereas this state space explosion is avoided by the current method due to aggregation.

The article is organized as follows. In Section 2 we elaborate on the production line of small appliances motivating this research. In Sections 3 and 4 we first describe the approximation method at high level, and then fill in the details in Section 5. In Section 6 we validate the approximation method on a large test set of 8,640 cases. Results show good performance of the proposed method. We also compare this method to other approximation methods from the literature and perform a sensitivity analysis of the practical case.

2. APPLICATION

The present study is motivated by the design of a new production line for small appliances. Figure 2 shows the proposed layout of this production line, consisting of six workstations, where the second and third one consist of multiple identical machines. In this figure, $W_{i,j}$ presents the jth machine of the ith workstation (W_i); B_i is a conveyor belt transporting the products from the ith to the $i+1$th workstation. The conveyor belts are also used for buffering. The circles (with R) are robots taking products from the machines to the conveyor belt (indicated by dashed arrows) and vice versa (indicated by solid arrows). Each machine is subject to operation-dependent failures. Products are transported through the system by means of carriers: products are placed on a carrier at the beginning of the production line, the carrier then moves along the workstations, and after processing at the last workstation, the final product is removed from the carrier and the empty carrier is returned to the beginning of the line.

TABLE 1 Estimated data for the production line of Figure 2

Workstation	**Mean uptime** $1/\lambda_i$	**Mean downtime** $1/\mu_i$	**Workstation speed** v_i	**Size of bufffer** B_i
W_1	169	9.0	26.8	63
W_2	143	11.3	9.4	150
W_3	221	11.1	5.0	125
W_4	257	7.2	25.9	13
W_5	757	24.0	26.8	8
W_6	227	4.4	26.8	

This production line is modeled as follows. First, the flow of products is represented as a fluid flow, which is justified by the high processing speed compared to the uptimes and downtimes, and we assume that there are always enough carriers available at the beginning of the line (so the influx never stops because of lack of carriers). We also assume that uptimes and downtimes are exponentially distributed. Finally, we assume that each buffer position can be reached from any machine of the upstream and downstream workstation. Based on these assumptions, the model of the production line fits in the present framework for exponential multi-server production lines, as shown in Figure 1 with $s_i = 1$ for $i = 1, 4, 5, 6$, and $s_2 = 3$, $s_3 = 5$.

Since this concerns the design of a new (not yet existing) production lines, only estimates of the machine and buffer data are available; see Table 1. A sensitivity analysis of this production line is referred to Section 6.

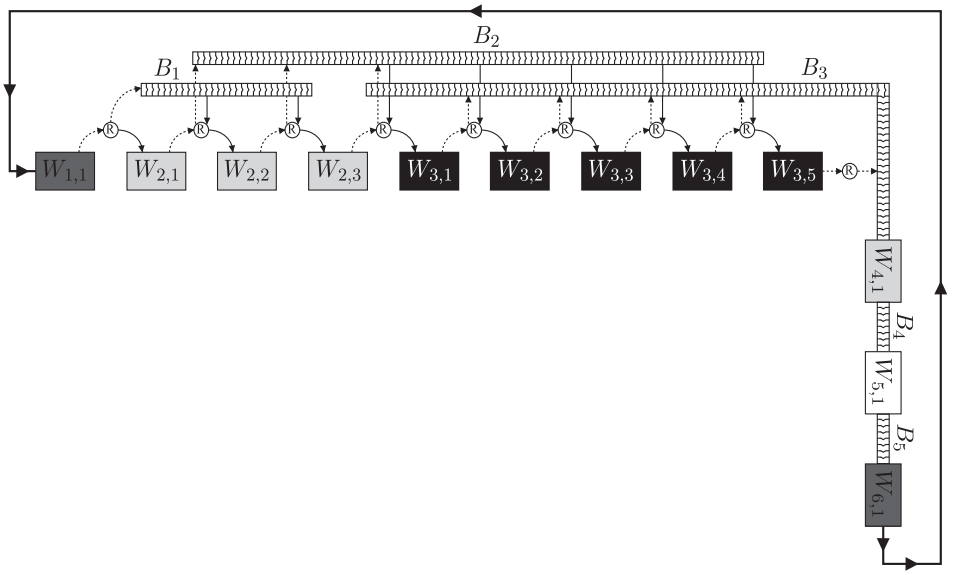


FIGURE 2 Schematic representation of a production line for small appliances.

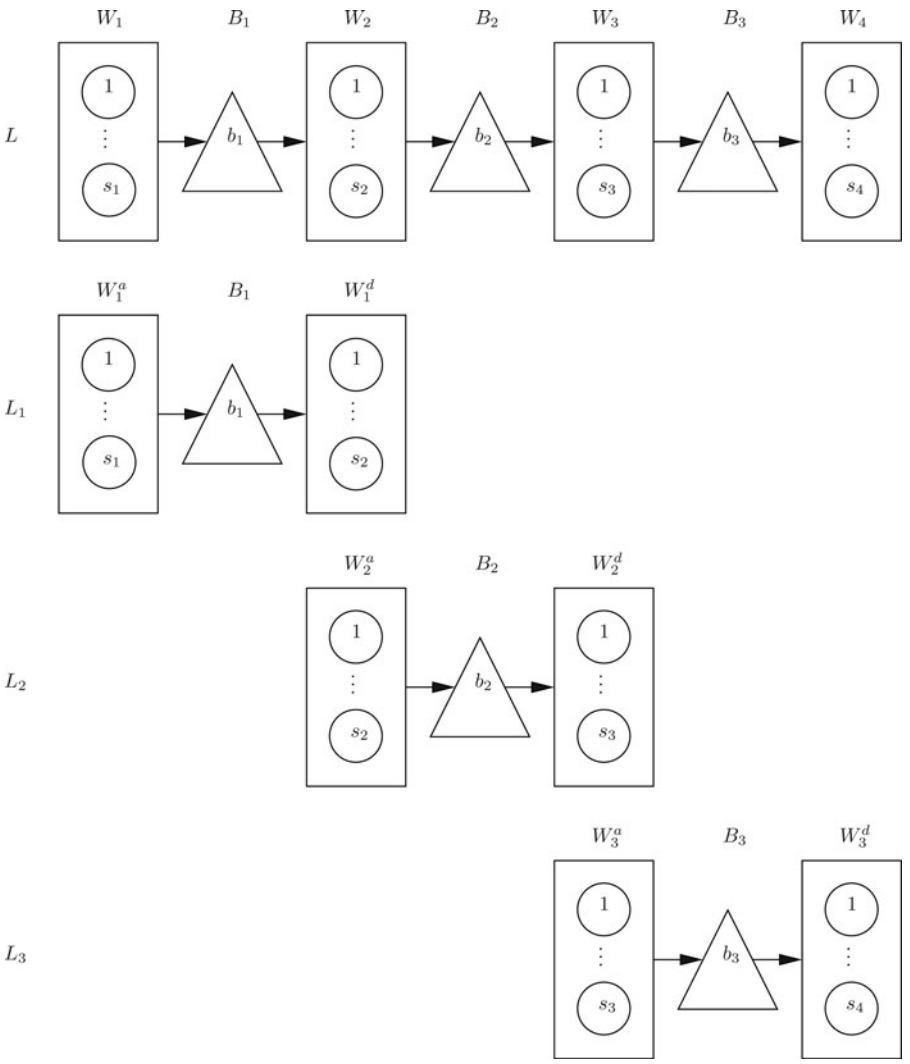


FIGURE 3 Decomposition of production line L in three single-buffer subsystems L_1, L_2, and L_3.

3. DECOMPOSITION

We decompose the production line L into subsystems $L_1, \ldots, L_{N-1}$ as illustrated in Figure 3. Each subsystem consists of the following three elements:

- *Arrival workstation* W_i^a, which behaves as original workstation W_i with s_i servers including starvation and speed adaptation caused by the *upstream* part of the production line. We model this workstation as a continuous-time Markov chain (CTMC) with $k_A^{(i)}$ states and generator $\mathbf{Q}_A^{(i)}$. The vector

of production speeds is defined as $\mathbf{r}_A^{(i)}$, the jth element of which is the production speed corresponding to state j of the CTMC, $j = 1, \ldots, k_A^{(i)}$.
- *Buffer* B_i of size b_i.
- *Departure workstation* W_i^d, which is original workstation W_{i+1} with s_{i+1} servers including blocking and speed adaptation caused by the *downstream* part of the production line. We model this workstation as a CTMC with $k_D^{(i)}$ states, generator $\mathbf{Q}_D^{(i)}$, and speed vector $\mathbf{r}_D^{(i)}$.

In the next section, we propose an iterative method to estimate the elements of $\mathbf{Q}_A^{(i)}$, $\mathbf{r}_A^{(i)}$, $\mathbf{Q}_D^{(i)}$, and $\mathbf{r}_D^{(i)}$, and ultimately the throughput and the average buffer content of the production line.

4. ITERATIVE METHOD

We present an iterative method to obtain the performance measures of a multi-server production line L, based on decomposition into subsystems $L_1, \ldots, L_{N-1}$ as explained in the previous section.

Step 0: Initialization

We initially assume that each subsystem L_i, $i = 1, \ldots, N-1$, is not affected by starvation or blocking due to upstream or downstream subsystems. The parameters in $\mathbf{Q}_A^{(i)}$, $\mathbf{r}_A^{(i)}$, $\mathbf{Q}_D^{(i)}$, and $\mathbf{r}_D^{(i)}$ are set accordingly.

Step 1: Subsystem analysis from left to right and update the arrival workstations

We analyze all subsystems, starting with L_1 up to L_{N-1}.

(a) **Construction of Markov chains for W_i^a and W_i^d**
We construct a CTMC describing the phase behavior of W_i^a using information from the upstream subsystem and a CTMC describing the phase behavior of W_i^d using information from the downstream subsystem. The elements of $\mathbf{Q}_A^{(i)}$, $\mathbf{r}_A^{(i)}$, $\mathbf{Q}_D^{(i)}$, and $\mathbf{r}_D^{(i)}$ are determined in this step; see Section 5.1.

(b) **Determination of steady-state distribution**
We determine the steady state distribution of the subsystem. This step is referred to Section 5.2.

(c) **Determination of throughput estimate**
Using the steady-state distribution, we determine the throughput $t_{n,1}^{(i)}$, where the subscript n, 1 refers to step 1 of the nth iteration. In Section 5.1 we formulate expressions for the throughput.

(d) **Update starvation and speed parameters if $i < N-1$**
This step updates the important parameters to construct new CTMCs for the phase behavior of arrival workstation W_{i+1}^a if $i < N-1$. More specifically, we determine the rate at which W_{i+1}^a goes starved and

the mean starvation time of W_{i+1}^a (both depending on the number of active servers). Furthermore, we determine the average production speed of W_{i+1}^a when j servers are up, $j = 1, \ldots, s_{i+1}$, which includes speed adaptation due to a slower upstream machine W_i^a. This step is explained in Section 5.3.

Step 2: Subsystem analysis from right to left and update the departure workstations

We analyze all subsystems, starting with L_{N-1} down to L_1.

(a) **Construction of Markov chains for W_i^a and W_i^d**
We construct a CTMC describing the phase behavior of W_i^a using information from the upstream subsystem and a CTMC describing the phase behavior of W_i^d using information from the downstream subsystem. The elements of $\mathbf{Q}_A^{(i)}$, $\mathbf{r}_A^{(i)}$, $\mathbf{Q}_D^{(i)}$, and $\mathbf{r}_D^{(i)}$ are determined in this step; see Section 5.1.

(b) **Determination of steady-state distribution**
We determine the steady-state distribution of the subsystem.

(c) **Determination of throughput estimate**
Using the steady-state distribution, we determine the throughput $t_{n,2}^{(i)}$, where subscript $n, 2$ refers to step 2 of the nth iteration.

(d) **Update blocking and speed parameters if $i > 1$**
This step updates the important parameters to construct new CTMCs for the phase behavior of departure workstation W_{i-1}^d if $i > 1$. That is, we update the rate at which W_{i-1}^d jumps to the blocked state, the mean blocking time of W_{i-1}^d (both depending on the number of active servers), and the average production rate of W_{i-1}^d when j servers are up, $j = 1, \ldots, s_i$, which includes speed adaptation due to a lower production speed of downstream machine W_i^d. This step is explained in Section 5.3.

Step 3: Repeat

We repeat steps 1–2 until the throughput estimates have converged. If for some small ϵ it holds that

$$\frac{|t_{n,2}^{(i)} - t_{n-1,2}^{(i)}|}{t_{n-1,2}^{(i)}} < \epsilon \quad \text{for all} \quad i \leq N-1,$$

then we stop; otherwise, we perform another iteration.

In the next section, we elaborate on the steps of the iterative algorithm. In contrast to the approximation for single-server lines[2,3], the conservation of flow property is not guaranteed for this algorithm, i.e., the throughput

estimates of the different subsystems do not necessarily converge to the same value. Therefore, we implement a conservation of flow equation to force the throughput estimates to be equal. This is explained in Section 5.4. Consistency equations, such as conservation of flow, are often used in decomposition methods; cf. Refs.[17,12].

Very few theoretical results are available in the literature about the convergence of iterative algorithms based on decomposition, and the available results typically concern special cases. Convergence is established by[23] for single-server *discrete* production lines with unreliable servers, all having the same processing rate. In the case of exponential single-server discrete production lines with *reliable* servers, Ref.[7] proves that, for each subsystem in their iterative algorithm, the successive estimates of the exponential service rate of the arrival and departure server are monotone and bounded sequences and thus converge. However, in the current setting of multi-server fluid flow production lines, we believe that the complexity of the subsystems prohibits a proof of convergence.

5. SUBSYSTEM ANALYSIS

In this section we analyze subsystem L_i, $i = 1, ..., N-1$. First, we construct a CTMC describing the phase process of both W_i^a and W_i^d. Using these CTMCs, we determine the steady-state distribution of L_i using matrix-analytic methods and update the parameters of W_{i+1}^a and W_{i-1}^d using the steady-state distribution of L_i. Last, we implement a conservation of flow equation to assure that the (limiting) throughput values are equal for all subsystems.

5.1. Phase Behavior of Arrival and Departure Workstation

We model the phase behavior of W_i^a and W_i^d as CTMCs with generators $\mathbf{Q}_A^{(i)}$ and $\mathbf{Q}_D^{(i)}$, respectively. Each state of the CTMC has a corresponding production speed; the vectors of production speeds are defined as $\mathbf{r}_A^{(i)}$ and $\mathbf{r}_D^{(i)}$, respectively.

Starting with the arrival workstation, we define state $u(j)$ as the state where W_i^a is not starved and j servers are up, $j = 0, \ldots, s_i$. Furthermore, the state $st(j)$ implies that W_i^a is starved by the upstream part of the line and j servers are up, $j = 1, \ldots, s_i$. With these definitions, the state space of the CTMC for W_i^a is given by $S_A^{(i)} = \{u(0), \ldots, u(s_i), st(1), \ldots, st(s_i)\}$. The number of states of this CTMC is

$$k_A^{(i)} = 2s_i + 1.$$

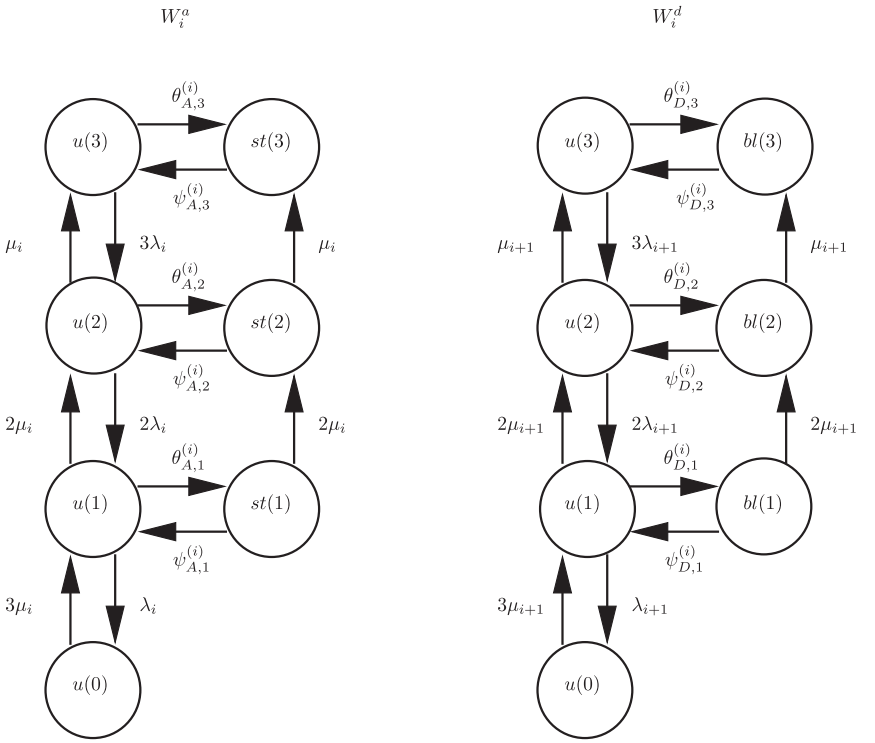


FIGURE 4 Division of states for W_i^a and W_i^d in case $s_i = 3$.

The left part of Figure 4 illustrates the structure of the CTMC for W_i^a in case $s_i = 3$. We distinguish four types of transitions:

- *Machine breakdowns*, which are transitions from $u(j)$ to $u(j-1)$, $j = 1, \ldots, s_i$. Since j servers are up in state $u(j)$, the rate of transitions of this type is given by $j\lambda_i$, so

$$\mathbf{Q}_A^{(i)}(u(j), u(j-1)) = j\lambda_i, \qquad j = 1, \ldots, s_i.$$

  Because the failures are operation dependent, machine breakdowns do not occur if W_i^a is starved.
- *Machine repairs*, which are transitions from $u(j)$ to $u(j+1)$, $j = 0, \ldots, s_i - 1$, or transitions from $st(j)$ to $st(j+1)$, $j = 1, \ldots, s_i - 1$. There are $s_i - j$ servers under repair in state $u(j)$ and $st(j)$, so the repair rate in these states is given by $(s_i - j)\mu_i$;

$$\begin{aligned} \mathbf{Q}_A^{(i)}(u(j), u(j+1)) &= (s_i - j)\mu_i, \quad j = 0, \ldots, s_i - 1, \\ \mathbf{Q}_A^{(i)}(st(j), st(j+1)) &= (s_i - j)\mu_i, \quad j = 1, \ldots, s_i - 1. \end{aligned}$$

- *Transitions from up to starved*, or equivalently, from $u(j)$ to $st(j)$. We define the rate of these transitions as $\theta_{A,j}^{(i)}$;

$$\mathbf{Q}_A^{(i)}(u(j), st(j)) = \theta_{A,j}^{(i)}, \qquad j = 1, \ldots, s_i.$$

- *Transitions from starved to up*, or equivalently, from $st(j)$ to $u(j)$, which occur at a rate of $\psi_{A,j}^{(i)}$;

$$\mathbf{Q}_A^{(i)}(st(j), u(j)) = \psi_{A,j}^{(i)}, \qquad j = 1, \ldots, s_i.$$

The diagonal elements of $\mathbf{Q}_A^{(i)}$ are chosen such that the sum of each row is equal to zero. The parameters $\theta_{A,j}^{(i)}$ and $\psi_{A,j}^{(i)}$ are obtained from the analysis of L_{i-1}; the determination of these parameters is referred to the next section.

The average production speed of workstation W_i^a in state $u(j)$, so when j servers are up, $j = 1, \ldots, s_i$, is defined as $v_{A,j}^{(i)}$. The average speeds are also obtained from the analysis of L_{i-1}; see the next section. Note that $v_{A,j}^{(i)}$ is not necessarily equal to jv_i/s_i, since W_i^a adjusts its speed when B_{i-1} is empty and W_{i-1}^a produces at a lower speed. The elements of speed vector $\mathbf{r}_A^{(i)}$ are given by

$$\begin{aligned}
&\mathbf{r}_A^{(i)}(u(0)) = 0, \\
&\mathbf{r}_A^{(i)}(u(j)) = v_{A,j}^{(i)}, \qquad j = 1, \ldots, s_i, \\
&\mathbf{r}_A^{(i)}(st(j)) = 0, \qquad j = 1, \ldots, s_i.
\end{aligned}$$

For the departure server, we define state $u(j)$ as the state where W_i^d is not blocked and j servers are up, $j = 0, \ldots, s_{i+1}$, and $bl(j)$ as the state where W_i^d is blocked and j servers are up, $j = 1, \ldots, s_{i+1}$. The CTMC for W_i^d is illustrated in the right part of Figure 4. The number of states for this CTMC is equal to

$$k_D^{(i)} = 2s_{i+1} + 1.$$

Since the analysis of W_i^d is symmetrical to the analysis of W_i^a, we just specify the elements of $\mathbf{Q}_D^{(i)}$.

- *Machine breakdowns*

$$\mathbf{Q}_D^{(i)}(u(j), u(j-1)) = j\lambda_{i+1}, \qquad j = 1, \ldots, s_{i+1}.$$

- *Machine repairs*

$$\mathbf{Q}_D^{(i)}(u(j), u(j+1)) = (s_{i+1} - j)\mu_{i+1},\ j = 0, \ldots, s_{i+1} - 1,$$
$$\mathbf{Q}_D^{(i)}(bl(j), bl(j+1)) = (s_{i+1} - j)\mu_{i+1},\ j = 0, \ldots, s_{i+1} - 1.$$

- *Transitions from up to blocked*

$$\mathbf{Q}_D^{(i)}(u(j), bl(j)) = \theta_{D,j}^{(i)}, \qquad j = 1, \ldots, s_{i+1}.$$

- *Transitions from blocked to up*

$$\mathbf{Q}_D^{(i)}(bl(j), u(j)) = \psi_{D,j}^{(i)}, \qquad j = 1, \ldots, s_{i+1}.$$

The diagonal elements of $\mathbf{Q}_D^{(i)}$ are chosen such that the sum of each row is equal to zero. The elements of $\mathbf{r}_D^{(i)}$ are given by

$$\begin{aligned}
&\mathbf{r}_D^{(i)}(u(0)) = 0,\\
&\mathbf{r}_D^{(i)}(u(j)) = v_{D,j}^{(i)}, \qquad j = 1, \ldots, s_{i+1},\\
&\mathbf{r}_D^{(i)}(bl(j)) = 0, \qquad j = 1, \ldots, s_{i+1},
\end{aligned}$$

where $v_{D,j}^{(i)}$ is defined as the average production speed of W_i^d when j servers are up, $j = 1, \ldots, s_{i+1}$. The determination of $\theta_{D,j}^{(i)}$, $\psi_{D,j}^{(i)}$, and $v_{D,j}^{(i)}$ is referred to the next section.

5.2. Steady-State Distribution

In this section, we determine the steady-state distribution of L_i. For ease of notation, we drop the subscript i and superscript (i) referring to the ith subsystem. We merge the CTMCs of both W^a and W^d into a new CTMC describing the phase behavior of the whole subsystem. The tuple of variables (i_A, i_D, x) describes the state of the subsystem, where $i_A \in S_A$ is the state (or phase) of W^a, $i_D \in S_D$ is the state of W^d, and $0 \le x \le b$ is the fluid level of the buffer. The CTMC describing L_i has generator $\mathbf{Q}$ and (net) speed vector $\mathbf{r}$ given by

$$\mathbf{Q} = \mathbf{Q}_A \otimes \mathbf{I}_{k_D} + \mathbf{I}_{k_A} \otimes \mathbf{Q}_D,$$
$$\mathbf{r} = \mathbf{r}_A \otimes \mathbf{1}_{k_D} - \mathbf{1}_{k_A} \otimes \mathbf{r}_D,$$

where $A \otimes B$ is the Kronecker product of matrices A and B, $\mathbf{I}_n$ is the identity matrix of size n and $\mathbf{1}_n$ is a column vector of ones of size n. This CTMC has

state space S, where $S = S_A \times S_D$. In generator $\mathbf{Q}$ and (net) speed vector $\mathbf{r}$ these states are ordered lexicographically.

Because of operation-dependent failures, no servers of W^a can go down whenever B is full and W^d is in a state with zero-speed. In this situation, it is also not possible for W^a to go starved. Similarly, W^d cannot become blocked, and no servers of W^d can go down when W^a is in a state with zero-speed and B is empty. Therefore, we define a "full-buffer" generator $\mathbf{Q}_A^F$ containing the transition rates of the phase process of W^a that apply when B is full and W^d is in a state with zero-speed. So in case B is full and W^d is in state $u(0)$ or $bl(j)$, the transition rates of W^a from state $u(j)$ to $u(j-1)$ and from state $u(j)$ to $st(j)$, are set to zero. The other transition rates in $\mathbf{Q}_A^F$ are equal to those in $\mathbf{Q}_A$. The full-buffer generator $\mathbf{Q}^F$, describing the CTMC of the whole subsystem when buffer B is full, is given by

$$\mathbf{Q}^F = \mathbf{Q}_A^F \otimes \tilde{\mathbf{I}}_{k_D} + \mathbf{Q}_A \otimes (\mathbf{I}_{k_D} - \tilde{\mathbf{I}}_{k_D}) + \mathbf{I}_{N_A} \otimes \mathbf{Q}_D,$$

where the jth diagonal element of the diagonal matrix $\tilde{\mathbf{I}}_{k_D}$ is 1 if $r_{D,j} = 0$ and 0 otherwise. Further, we define an "empty-buffer" generator $\mathbf{Q}_D^E$ containing transition rates of W^d that apply when W^a is in a state with zero-speed and B is empty. So in case B is empty and W^a is in state $u(0)$ or $st(j)$ the transition rates of W^d from states $u(j)$ to $u(j-1)$ and from $u(j)$ to $st(j)$ are set to zero. The other transition rates in $\mathbf{Q}_D^E$ are the same as those in $\mathbf{Q}_D$. Hence, the empty-buffer generator $\mathbf{Q}^E$, describing the CTMC of the whole subsystem when buffer B is empty, is given by

$$\mathbf{Q}^E = \mathbf{Q}_A \otimes \mathbf{I}_{k_D} + \tilde{\mathbf{I}}_{k_A} \otimes \mathbf{Q}_D^E + (\mathbf{I}_{k_A} - \tilde{\mathbf{I}}_{k_A}) \otimes \mathbf{Q}_D,$$

where the jth diagonal element of the diagonal matrix $\tilde{\mathbf{I}}_{k_A}$ is 1 if $r_{A,j} = 0$ and 0 otherwise.

We define $F_i(x)$ as the probability that the phase process is in state $i \in S$, and the buffer level is less than or equal to x, $0 \leq x \leq b$. This probability is also referred to as the cumulative density function of subsystem L.

We determine these cumulative density functions using the numerically stable matrix-analytic methods for fluid models developed in Refs.[10,9]; see also Chapter 2 in Ref.[1].

5.3. Update Parameters

In this section, we update the parameters for the downstream arrival workstation and for the upstream departure workstation. More specifically, for W_{i+1}^a we update the rate from up to starved ($\theta_{A,j}^{(i+1)}$), the rate from starved to up ($\psi_{A,j}^{(i+1)}$), and the average speed ($v_{A,j}^{(i+1)}$) for $j = 1, \ldots, s_{i+1}$. For W_{i-1}^d, we update the transition rate from up to blocked ($\theta_{D,j}^{(i-1)}$), the rate from

blocked to up ($\psi_{D,j}^{(i-1)}$), and the production speed ($v_{D,j}^{(i-1)}$) for $j = 1, \ldots, s_i$. We introduce the following variables, all obtained from the cumulative density function of L_i:

- $\pi_{k_A,k_D}^{(i)}$: the probability that W_i^a is in state $k_A \in S_A^{(i)}$ and W_i^d is in state $k_D \in S_D^{(i)}$.
- $p_{k_A,k_D}^{(i)}(0)$ and $p_{k_A,k_D}^{(i)}(b)$: boundary probabilities at the levels 0 and b. The subscript (k_A, k_D) indicates that W_i^a is in state $k_A \in S_A^{(i)}$ and W_i^d is in state $k_D \in S_D^{(i)}$.
- $f_{k_A,k_D}^{(i)}(0)$ and $f_{k_A,k_D}^{(i)}(b)$: the value of the probability density function (pdf) at the levels 0 and b for state $k_A \in S_A^{(i)}$ and $k_D \in S_D^{(i)}$.

5.3.1. Update Arrival Workstation W_{i+1}^a

Note that arrival workstation W_{i+1}^a corresponds to workstation W_i^d. Starting with $\theta_{A,j}^{(i+1)}$, we argue that W_{i+1}^a can go from up to starved in two different ways:

(i) First, W_i^a is not producing (i.e., starved or down), W_i^d is up and B_i is non-empty. This means that W_i^d drains the buffer until finally it happens that B_i becomes empty, and thus W_{i+1}^a gets starved.
(ii) First, W_i^a and W_i^d are both up and B_i is empty. Then W_i^a stops producing, and simultaneously W_{i+1}^a gets starved.

The probability that B_i becomes empty in a small time interval dt when W_i^a is in a zero-speed state k and W_i^d is draining the buffer at a speed $v_{D,j}(i)$ is given by $f_{k,u(j)}^{(i)}(0)\, v_{D,j}^{(i)} dt$. Dividing by dt and summing over all zero-speed states $k \in S_i^a$, the frequency of occurrence of type-(i) jumps per time unit is given by

$$f_{u(0),u(j)}^{(i)}(0)\, v_{D,j}^{(i)} + \sum_{k=1}^{s_i} f_{st(k),u(j)}^{(i)}(0)\, v_{D,j}^{(i)}.$$

Type-(ii) jumps occur at a rate of $Q_A^{(i)}(u(1), u(0)) + Q_A^{(i)}(u(1), st(1))$ if one server of W_i^a is up and B_i is empty (w.p. $p_{u(1),u(j)}^{(i)}(0)$) and at a rate of $Q_A^{(i)}(u(k), st(k))$ if $k = 2, \ldots, s_i$ servers are up and B_i is empty (w.p. $p_{u(k),u(j)}^{(i)}(0)$). Together, the number of type-(ii) jumps per time unit is given by

$$p_{u(1),u(j)}^{(i)}(0)\, Q_A^{(i)}(u(1), u(0)) + \sum_{k=1}^{s_i} p_{u(k),u(j)}^{(i)}(0)\, Q_A^{(i)}(u(k), st(k)).$$

Conditioning on the fact that j servers of W_i^d (or equivalently, W_{i+1}^a) are up and W_i^d is producing, we obtain the transition rate from up to starved

$$\theta_{A,j}^{(i+1)} = \frac{f_{u(0),u(j)}^{(i)}(0)\,v_{D,j}^{(i)} + \sum_{k=1}^{s_i} f_{st(k),u(j)}^{(i)}(0)\,v_{D,j}^{(i)}}{\sum_{k=0}^{s_i}\pi_{u(k),u(j)}^{(i)} - p_{u(0),u(j)}^{(i)}(0) + \sum_{k=1}^{s_i}(\pi_{st(k),u(j)}^{(i)} - p_{st(k),u(j)}^{(i)}(0))}$$

$$+ \frac{p_{u(1),u(j)}^{(i)}(0)\,Q_A^{(i)}(u(1),u(0)) + \sum_{k=1}^{s_i} p_{u(k),u(j)}^{(i)}(0)\,Q_A^{(i)}(u(k),st(k))}{\sum_{k=0}^{s_i}\pi_{u(k),u(j)}^{(i)} - p_{u(0),u(j)}^{(i)}(0) + \sum_{k=1}^{s_i}(\pi_{st(k),u(j)}^{(i)} - p_{st(k),u(j)}^{(i)}(0))}.$$

Next, we determine rate $\psi_{A,j}^{(i+1)}$ at which W_{i+1}^a goes from starved to up for $j = 1,\ldots,s_{i+1}$. If W_{i+1}^a is in state $st(j)$, W_{i+1}^a goes up with rate $Q_A^{(i)}(u(0),u(1))$ if all servers of workstation W_i^a are down (w.p. $\frac{p_{u(0),u(j)}^{(i)}(0)}{p_{u(0),u(j)}^{(i)}(0)+\sum_{k=1}^{s_i} p_{st(k),u(j)}^{(i)}(0)}$) and with rate $Q_A^{(i)}(st(k),u(k))$ if W_i^a is also starved and $k = 1,\ldots,s_i$ servers of W_i^a are up (w.p. $\frac{p_{st(k),u(j)}^{(i)}(0)}{p_{u(0),u(j)}^{(i)}(0)+\sum_{k=1}^{s_i} p_{st(k),u(j)}^{(i)}(0)}$). Together, the rate of transitions from starved to up of W_{i+1}^a, when j servers are up, is given by

$$\psi_{A,j}^{(i+1)} = \frac{p_{u(0),u(j)}^{(i)}(0)\,Q_A^{(i)}(u(0),u(1)) + \sum_{k=1}^{s_i} p_{st(k),u(j)}^{(i)}(0)\,Q_A^{(i)}(st(k),u(k))}{p_{u(0),u(j)}^{(i)}(0) + \sum_{k=1}^{s_i} p_{st(k),u(j)}^{(i)}(0)}.$$

Next, we determine the average production speed $v_{A,j}^{(i+1)}$ at which W_{i+1}^a produces when j servers are up. When B_i is nonempty, W_{i+1}^a produces at a speed of jv_{i+1}/s_{i+1}. If B_i is empty and k servers of W_i^a are active and producing, W_{i+1}^a lowers its speed to $v_{A,k}^{(i)}$. Thus, the average speed $v_{A,j}^{(i+1)}$ is given by

$$v_{A,j}^{(i+1)} = jv_{i+1}/s_{i+1} + \frac{\sum_{k=1}^{s_i} p_{u(k),u(j)}^{(i)}(0)\,(v_{A,k}^{(i)} - jv_{i+1}/s_i)}{\sum_{k=0}^{s_i}\pi_{u(k),u(j)}^{(i)} - p_{u(0),u(j)}^{(i)}(0) + \sum_{k=1}^{s_i}(\pi_{st(k),u(j)}^{(i)} - p_{st(k),u(j)}^{(i)}(0))}. \quad (1)$$

This concludes step 1(d) of the iterative algorithm.

5.3.2. Update departure workstation W_{i-1}^d

Since the parameters for W_{i-1}^d are determined along the same lines as the parameters of W_{i+1}^a, we just provide the resulting expressions:

$$\theta_{D,j}^{(i-1)} = \frac{f_{u(j),u(0)}^{(i)}(b)\,v_{A,j}^{(i)} + \sum_{k=1}^{s_{i+1}} f_{u(j),bl(k)}^{(i)}(b)\,v_{A,j}^{(i)}}{\sum_{k=0}^{s_{i+1}} \pi_{u(j),u(k)}^{(i)} - p_{u(j),u(0)}^{(i)}(b) + \sum_{k=1}^{s_{i+1}} (\pi_{u(j),bl(k)}^{(i)} - p_{u(j),bl(k)}^{(i)}(b))},$$

$$+ \frac{p_{u(j),u(1)}^{(i)}(b)\,Q_D^{(i)}(u(1),u(0)) + \sum_{k=1}^{s_{i+1}} p_{u(j),u(k)}^{(i)}(b)\,Q_D^{(i)}(u(k),bl(k))}{\sum_{k=0}^{s_{i+1}} \pi_{u(j),u(k)}^{(i)} - p_{u(j),u(0)}^{(i)}(b) + \sum_{k=1}^{s_{i+1}} (\pi_{u(j),bl(k)}^{(i)} - p_{u(j),bl(k)}^{(i)}(b))}$$

$$\psi_{D,j}^{(i-1)} = \frac{p_{u(j),u(0)}^{(i)}(b)\,Q_D^{(i)}(u(0),u(1)) + \sum_{k=1}^{s_{i+1}} p_{u(j),bl(k)}^{(i)}(b)\,Q_D^{(i)}(bl(k),u(k))}{p_{u(j),u(0)}^{(i)}(b) + \sum_{k=1}^{s_{i+1}} p_{u(j),bl(k)}^{(i)}(b)},$$

$$v_{D,j}^{(i-1)} = jv_i/s_i$$

$$+ \frac{\sum_{k=1}^{s_{i+1}} p_{u(j),u(k)}^{(i)}(b)\,(v_{D,k}^{(i)} - jv_i/s_i)}{\sum_{k=0}^{s_{i+1}} \pi_{u(j),u(k)}^{(i)} - p_{u(j),u(0)}^{(i)}(b) + \sum_{k=1}^{s_{i+1}} (\pi_{u(j),bl(k)}^{(i)} - p_{u(j),bl(k)}^{(i)}(b))}.$$

This concludes step 2(d) of the iterative algorithm. The next section is devoted to the conservation of flow equation.

5.4. Conservation of Flow

Conservation of flow is not always satisfied when using the iterative method of Section 4. This means that the throughput estimates are not equal for all subsystems, when all parameters have converged. Therefore, after convergence of the parameters, we include a conservation of flow equation assuring that all throughput estimates converge to the same value, i.e., $t_{(1)} = \cdots = t_{(N-1)}$. So we proceed the iteration, now with the conservation of flow equation, until all throughput estimates have converged (to the same value). In this section, we explain the implementation of the conservation of flow equation. We note that use of consistency equations, such as conservation of flow, is common in decomposition methods; cf. Refs.[17,12]. Surprisingly, for the iterative methods in Ref.[2,3], developed for *single-server* fluid flow production lines, it seems that conservation of flow is always satisfied, also without the inclusion of a conservation of flow equation.

Conservation of flow will be imposed by scaling both speeds $v_{A,j}^{(i+1)}$ of the arrival workstation (in step 1) and speeds $v_{D,j}^{(i-1)}$ of the departure workstation

(in step **2**) to match the average throughput over all subsystems. To this end, we add step **1(e)** and step **2(e)** to the iterative method in Section 4.

In step **1(e)** we scale the speeds $v_{A,j}^{(i+1)}$ calculated in step **1(d)**. Note that by scaling we change only the magnitude of speeds and not the shape of the speed profile. Let $\tilde{v}_{A,j}^{(i+1)}$ denote the new estimate of the speed of the arrival workstation of subsystem L_{i+1}, $i = 1, \ldots, N-2$ when j servers are up. This speed is related to $v_{A,j}^{(i+1)}$ as follows:

$$\tilde{v}_{A,j}^{(i+1)} = z_A^{(i+1)} v_{A,j}^{(i+1)}, \qquad j = 1, \ldots, s_{i+1}.$$

The throughput of subsystem L_{i+1}, $i = 1, \ldots, N-2$ is given by

$$t^{(i+1)} = \sum_{j=1}^{s_{i+1}} \sum_{k_D \in S_D^{(i+1)}} \left(\pi_{u(j),k_D}^{(i+1)} - p_{u(j),k_D}^{(i+1)}(b) \right) v_{A,j}^{(i+1)} + \sum_{j=1}^{s_{i+1}} \sum_{k=1}^{s_{i+2}} p_{u(j),u(k)}^{(i+1)}(b)\, v_{D,k}^{(i+1)}. \tag{2}$$

We now replace throughput $t^{(i+1)}$ by the average throughput obtained in step **2** of the previous iteration $n-1$, i.e., $t_{n-1,2} = \frac{1}{N-1} \sum_{i=1}^{N-1} t_{n-1,2}^{(i)}$, and we scale the speeds $v_{A,j}^{(i+1)}$, such that (2) is again satisfied. Thus,

$$z_A^{(i+1)} = \frac{t_{n-1,2} - \sum_{j=1}^{s_{i+1}} \sum_{k=1}^{s_{i+2}} p_{u(j),u(k)}^{(i+1)}(b)\, v_{D,k}^{(i+1)}}{\sum_{j=1}^{s_{i+1}} \sum_{k_D \in S_D^{(i+1)}} \left(\pi_{u(j),k_D}^{(i+1)} - p_{u(j),k_D}^{(i+1)}(b) \right) v_{A,j}^{(i+1)}}, \tag{3}$$

where we use the most recent estimates for $\pi_{u(j),k_D}^{(i+1)}$, $p_{u(j),u(k)}^{(i+1)}(b)$, $v_{A,j}^{(i+1)}$ and $v_{D,k}^{(i+1)}$.

Similarly, in step 2(e) we scale the speeds $v_{D,j}^{(i-1)}$ calculated in step 2(d). Let $\tilde{v}_{D,j}^{(i-1)}$ denote the new estimate of the speed of the departure workstation subsystem L_{i-1}, $i = 2, \ldots, N-1$ when j servers are up. This speed is then related to $v_{D,j}^{(i-1)}$ as follows:

$$\tilde{v}_{D,j}^{(i-1)} = z_D^{(i-1)} v_{D,j}^{(i-1)}, \qquad j = 1, \ldots, s_{i-1}.$$

The throughput of subsystem L_{i-1}, $i = 2, \ldots, N-1$ is given by

$$t^{(i-1)} = \sum_{j_A \in S_A^{(i-1)}} \sum_{k=1}^{s_i} \left(\pi_{j_A,u(k)}^{(i-1)} - p_{j_A,u(k)}^{(i-1)}(0) \right) v_{D,k}^{(i-1)} + \sum_{j=1}^{s_{i-1}} \sum_{k=1}^{s_i} p_{u(j),u(k)}^{(i-1)}(0)\, v_{A,j}^{(i-1)}. \tag{4}$$

Again, we now replace throughput $t^{(i-1)}$ with the average throughput obtained in step **1** of the current iteration n, i.e., we replace $t^{(i-1)}$ by

$t_{n,1} = \frac{1}{N-1} \sum_{i=1}^{N-1} t_{n,1}^{(i)}$, and we scale the speeds $v_{D,j}^{(i-1)}$, such that (4) is satisfied. Thus,

$$z_D^{(i-1)} = \frac{t_{n,1} - \sum_{j=1}^{s_{i-1}} \sum_{k=1}^{s_i} p_{u(j),u(k)}^{(i-1)}(0)\, v_{A,j}^{(i-1)}}{\sum_{j_A \in S_A^{(i-1)}} \sum_{k=1}^{s_i} \left(\pi_{j_A,u(k)}^{(i-1)} - p_{j_A,u(k)}^{(i-1)}(0)\right) v_{D,k}^{(i-1)}}, \tag{5}$$

where we use the most recent estimates for $\pi_{j_A,u(k)}^{(i-1)}$, $p_{u(j),u(k)}^{(i-1)}(0)$, $v_{A,j}^{(i-1)}$ and $v_{D,k}^{(i-1)}$.

6. RESULTS

In this section we test the quality of the proposed approximation method for multi-server production lines. First, we test the method on a large test set, then we compare it to other approximation methods from the literature. Second, we perform a sensitivity analysis for the application mentioned in Section 2.

6.1. Test Set

The performance of the approximation method is tested on a large test set, in which six parameters are varied: the number of workstations in the line, mean uptimes, mean downtimes, the number of servers per workstation, the maximum speeds per workstation, and buffer sizes. Table 2 provides the different settings for each parameter. The test set includes imbalance in mean uptimes, mean downtimes, machine speeds, and number of servers per workstation. The maximum speed of a workstation is defined as the number of servers of that workstation times the speed per server. We consider three different setups for the maximum workstation speeds: a homogeneous setup (all workstations produce at equal speeds), an alternating speed setup (all odd workstations produce at speed 10 and all even ones produce at speed 15), and a V-shape setup, where the first and last workstation are the fastest,

TABLE 2 Input parameter values of the test set

Input parameter	Values
Number of workstations	4, 6, 8
Mean uptimes	{5,5,5,5,...}, {5,2.5,5,2.5,...}, {10,10,10,10,...}, {10,5,10,5,...}, {20,20,20,20,...}, {20,10,20,10,...}
Mean downtimes	{0.5,0.5,0.5,0.5,...}, {0.5,0.25,0.5,0.25,...}, {1,1,1,1,...}, {1,0.5,1,0.5,...}, {2,2,2,2,...}, {2,1,2,1,...}, {4,4,4,4,...}, {4,2,4,2,...}
Number of servers per workstation	{2,2,2,2,...}, {3,2,3,2,...}, {3,3,3,3,...}, {5,3,5,3,...}, {5,5,5,5,...}
Workstation maximum speeds	{10,10,10,10,...}, {15,10,15,10,...}, {15,...,10,...,15}
Buffer sizes	{1,1,1,1,...}, {5,5,5,5,...}, {10,10,10,10,...}, {25,25,25,25,...}

TABLE 3 Results for production lines of different lengths

Line	Error (%) in the throughput							Error (%) in avg. buffer content						
length	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
4	2.69	57	19	11	5	3	4	5.08	44	14	11	7	6	18
6	4.64	43	15	12	9	6	14	5.03	47	17	9	5	4	17
8	5.86	37	15	11	9	8	20	5.68	42	20	9	6	5	19

speeds decrease linearly in the first part of the production line, and speeds increase linearly in the second part of the production line. By making all possible combinations of parameter settings in Table 2, we obtain a test set of $3 \times 6 \times 8 \times 5 \times 3 \times 4 = 8{,}640$ cases.

To test the quality of the method, we focus on two variables: throughput and average total buffer content. We compare the quantities obtained from the approximation method to those of a discrete-event simulation model, the 95 %confidence intervals of which have a width of at most 0.5%. Tables 3–8 show the results of this comparison. The columns "Avg" provide the average (absolute) relative error over all cases that satisfy the property in the first column. For instance, the 2.50% in Table 3 is the average error for all cases with 4 workstations in the line. The columns "0–2," "2–4," "4–6," "6–8," "8–10," and ">10," provide the percentage of cases satisfying the property in the first column that fall into the specified error range.

The approximation method typically overestimates the throughput: in approximately 96.3% of the cases, the throughput estimates exceed the throughput values obtained by simulation. This effect may be understood by the following example. Consider a four-workstation production line with three servers per workstation. The speed of each server is 5. We decompose the line into subsystems L_1, L_2, and L_3 as illustrated in Figure 3. When looking at the arrival server of subsystem L_2, one of the parameters is $v_{A,3}^{(2)}$, the average speed at which workstation A_2 produces when all three servers are up. In reality, W_2^a can have three possible speeds when all servers are up: 15 when B_1 is nonempty or when B_1 is empty and all three servers of W_1 are up, 10 when B_1 is empty and two servers of W_1 are up, and 5 when B_1 is empty and one server of W_1 is up. In the iterative algorithm, we calculate $v_{A,3}^{(2)}$ using equation (1), which is a weighted average over the three possible speeds. Aggregating these speeds into a single value (thus making it less variable) leads to overestimates. It is possible to not aggregate these speeds, but then the state space explodes for longer production lines.

Table 3 shows that, not surprisingly, the most accurate throughput estimates are obtained for shorter lines. The error in average total buffer content seems to be less sensitive to the number of workstations in the line. Tables 4 and 5 show that the errors in throughput increase as the mean uptimes

TABLE 4 Results for production lines with different mean uptimes

Mean	Error (%) in the throughput							Error (%) in avg. buffer content						
uptimes	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
5.00,2.50,5.00,2.50,...	5.82	34	17	13	10	7	19	3.33	49	24	11	5	3	7
5.00,5.00,5.00,5.00,...	5.14	38	17	12	9	8	15	3.96	50	18	11	6	5	11
10.00,5.00,10.00,5.00,...	4.94	42	16	12	8	7	15	4.50	44	21	11	6	5	14
10.00,10.00,10.00,10.00,...	4.01	48	17	11	7	6	11	5.68	45	13	9	6	6	21
20.00,10.00,20.00,10.00,...	3.67	53	16	10	7	4	10	6.25	38	17	9	7	6	24
20.00,20.00,20.00,20.00,...	2.79	60	16	8	5	4	6	7.87	41	11	6	5	6	31

decrease or the mean downtimes increase. In these cases, workstations have to adjust their speeds more frequently, and so, by following the arguments in the previous paragraph, larger errors (overestimates) in throughput can be expected. Table 6 does not show a clear relation between errors in performance estimates and the number of servers per workstation. Table 7 shows that the estimates for production lines with a Vshape speed set-up perform are significantly better than for production lines with a homogeneous or alternating speed profile. The conclusion from Table 8 is clear: the larger the buffer, the smaller the errors. In production lines with small buffers, the effect of starvation, blocking and speed adaptation is larger, yielding larger errors in performance estimates.

6.2. Comparison

In this section, we compare the current approximation method to existing ones from the literature. The approximation method in Ref.[5] is able to analyze production lines with multiple up- and multiple down-states in each

TABLE 5 Results for production lines with different mean downtimes

Mean	Error (%) in the throughput							Error (%) in avg. buffer content						
downtimes	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
0.50,0.25,0.50,0.25,...	0.84	87	8	3	1	0	0	8.25	48	9	4	2	3	34
0.50,0.50,0.50,0.50,...	1.70	73	14	6	3	2	2	6.56	44	15	5	3	4	29
1.00,0.50,1.00,0.50,...	2.11	67	16	8	4	2	3	5.90	51	7	6	6	5	24
1.00,1.00,1.00,1.00,...	3.61	45	22	12	7	6	8	4.98	46	18	7	5	6	18
2.00,1.00,2.00,1.00,...	4.22	37	22	15	10	6	10	4.59	49	13	10	7	7	15
2.00,2.00,2.00,2.00,...	6.20	22	18	16	13	11	20	3.71	47	25	10	6	4	8
4.00,2.00,4.00,2.00,...	7.02	19	17	16	13	10	24	4.53	32	23	18	11	6	9
4.00,4.00,4.00,4.00,...	9.48	16	14	12	10	11	36	3.60	40	26	16	7	4	6

TABLE 6 Results for production lines with different number of servers per workstation

Number of	Error (%) in the throughput							Error (%) in avg. buffer content						
servers	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
2.00,2.00,2.00,2.00,...	4.80	35	17	14	10	9	14	4.82	46	16	9	7	6	16
3.00,2.00,3.00,2.00,...	3.67	46	20	12	8	6	8	5.33	47	14	10	6	4	19
3.00,3.00,3.00,3.00,...	5.86	41	14	10	7	6	21	5.57	38	20	11	6	6	19
5.00,3.00,5.00,3.00,...	3.02	55	18	10	7	5	6	5.08	49	17	7	5	4	18
5.00,5.00,5.00,5.00,...	4.64	52	14	9	6	4	15	5.53	42	18	10	6	4	19

workstation. This includes multi-server lines. Memory (and also time) complexity of this method, however, is more demanding than for the current method. This is due to the level of detail at which subsystems are modeled: the approximation method in Ref.[5] employs detailed subsystem models with a large state space, whereas the current method uses aggregation to reduce the size of the state space. To explain the difference in the size of the state space, recall from Sections 5.1 and 5.2 that the number of states $k^{(i)}$ of subsystem $i = 1, ..., N - 1$ is given by

$$k^{(i)} = k_A^{(i)} k_D^{(i)},$$

where

$$k_A^{(i)} = 2s_i + 1,$$
$$k_D^{(i)} = 2s_{i+1} + 1.$$

The state space of the subsystem model in Ref.[5] is larger: For W_i^a or W_i^d having to adjust their speed, they need to keep track of which workstation causes this speed reduction and how many servers of this workstation are

TABLE 7 Results for production lines with different maximum workstation speeds

Number of	Error (%) in the throughput							Error (%) in avg. buffer content						
servers	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
10.00,10.00,10.00,10.00,...	5.19	36	20	13	9	6	16	3.53	47	28	10	5	3	7
15.00,10.00,15.00,10.00,...	5.86	31	18	14	10	9	19	10.75	10	12	12	10	11	46
15.00,...,10.00,...,15.00	2.14	71	12	6	4	3	4	1.52	77	12	6	3	1	1

TABLE 8 Results for production lines with different buffer sizes

Buffer	Error (%) in the throughput							Error (%) in avg. buffer content						
sizes	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**	**Avg.**	**0–2**	**2–4**	**4–6**	**6–8**	**8–10**	**>10**
1	7.40	22	16	13	11	9	27	7.44	29	16	12	9	7	27
5	4.79	40	18	12	8	7	14	4.27	48	19	8	6	6	14
10	3.44	52	17	11	7	5	8	4.16	51	17	10	5	5	13
25	1.96	68	15	8	4	2	2	5.19	50	17	8	4	3	18

active and producing. For this model, $k_A^{(i)}$ and $k_D^{(i)}$ are given by

$$rclk_A^{(i)} = 1 + s_i + \sum_{l=1}^{i-1} \left(1 + \sum_{j=1}^{s_i} \sum_{j'=1}^{s_l} x_{l,j,j'} \right),$$

$$k_D^{(i)} = 1 + s_{i+1} + \sum_{l=i+2}^{N} \left(1 + \sum_{j=1}^{s_{i+1}} \sum_{j'=1}^{s_l} y_{l,j,j'} \right),$$

where

$$x_{l,j,j'} = \begin{cases} 1 & \text{if } jv_i > j'v_l \\ 0 & \text{otherwise} \end{cases}$$

$$y_{l,j,j'} = \begin{cases} 1 & \text{if } jv_{i+1} > j'v_l \\ 0 & \text{otherwise} \end{cases}$$

TABLE 9 Parameters for test cases

Case	Mean uptimes	Mean downtimes	Number of servers	Maximum workstation speeds	Buffer sizes
1	(100,100,100)	(10,10,10)	(1,2,1)	(1,2,1)	(10,10)
2	(100,100,100)	(10,10,10)	(1,2,1)	(1,1,1)	(10,10)
3	(100,8.33,100)	(10,10,10)	(1,2,1)	(1,2,1)	(10,10)
4	(100,100,100)	(10,10,10)	(1,5,1)	(1,5,1)	(10,10)
5	(100,100,100)	(10,10,10)	(1,5,1)	(1,1,1)	(10,10)
6	(100,100,100)	(10,10,10)	(1,2,1)	(1,2,1)	(1,1)
7	(100,100,100)	(10,10,10)	(1,2,1)	(1,1,1)	(1,1)
8	(100,8.33,100)	(10,10,10)	(1,2,1)	(1,2,1)	(1,1)

TABLE 10 Throughput results for test cases in Table 9

	Current method				PW-method		Bur-method		CG-method	
Case	**Sim**	**App**	**Dif**	**Sim**	**App**	**Dif**	**App**	**Dif**	**App**	**Dif**
1	0.8695	0.8795	1.04%	0.872	0.863	−1.03%	0.861	−1.26%	0.8740	0.23%
2	0.8287	0.8295	0.10%	0.830	0.832	0.24%	0.830	0.00%	0.8300	0.00%
3	0.7429	0.7443	0.19%	0.756	0.680	−10.05%	0.728	−3.70%	0.7528	−0.42%
4	0.8712	0.8808	1.10%	0.884	0.873	−1.24%	—	—	0.8758	−0.93%
5	0.8339	0.8345	0.07%	0.847	0.838	−1.06%	—	—	0.8381	−1.05%
6	0.8353	0.8360	0.08%	0.838	—	—	0.811	−3.22%	0.8376	−0.05%
7	0.7748	0.7750	0.03%	0.781	—	—	0.778	−0.38%	0.7793	−0.21%
8	0.6432	0.6432	0.00%	0.676	—	—	0.590	−12.72%	0.6710	−0.74%

For instance, for the eight-workstation production line from the previous section, with 5 servers per workstation and an alternating speed setup, the number of states of subsystem L_4 is 121 for the current approximation method and 2,745 in Ref.[5].

We compare the current approximation method in terms of accuracy with three other methods from the literature[20,4,5]. We test all four methods on eight three-workstation cases, in which the second workstation is multi-server. Table 9 provides the input parameters for the test cases. Cases 1–5 in Table 9 are cases 1, 3, 5, 7, and 9 reported in Ref.[20], and cases 6–8 in Table 9 are cases 46, 52, and 58 reported in Ref.[4]. Note that in these cases, only the buffer sizes and the parameters for the second workstation are varied, the other parameters are kept constant. It should be mentioned that the current method applies to a model that is slightly different from the one considered in Refs.[20,4,5]: it is assumed in Refs.[20,4,5] that the exponential rates of the uptimes are adapted *proportionally* to changes in the machine speed, whereas we assume that these rates are *independent* of the machine speed. So we compare the results of the current method to simulation results of the model with parameters as in Table 9, but with uptimes being independent of the machines speeds. For each case, the number of simulation runs is chosen such that the width of the 95% confidence intervals of the performance measures is less than 0.5%.

TABLE 11 Sensitivity analysis for the production line of Section 2

	Mean uptime ×2	**Mean downtime ÷2**	**Speeds +10%**	**Buffers +25**
W_1	1.12%	1.67%	0.03%	0.23%
W_2	0.86%	0.75%	0.76%	0.33%
W_3	1.94%	1.97%	5.25%	0.32%
W_4	0.71%	0.98%	0.27%	0.20%
W_5	1.10%	1.24%	0.00%	0.10%
W_6	0.22%	0.62%	0.00%	—

TABLE 12 Comparison of the throughput estimates obtained from the approximation method to a discrete-event simulation model for the production line of Section 2

	Mean uptime ×2			**Mean downtime ÷2**			**Speed +10%**			**Buffer +25**		
	Sim	**App**	**Dif**	**Sim**	**App**	**Dif**	**Sim**	**App**	**Dif**	**Sim**	**App**	**Dif**
—	21.94	22.07	0.59%	21.94	22.07	0.59%	21.94	22.07	0.59%	21.94	22.07	0.59%
W_1	22.23	22.31	0.36%	22.34	22.44	0.41%	22.00	22.07	0.34%	21.99	22.12	0.59%
W_2	22.16	22.26	0.46%	22.22	22.23	0.05%	22.11	22.24	0.58%	22.00	22.14	0.61%
W_3	22.36	22.49	0.58%	22.37	22.50	0.60%	22.87	23.23	1.54%	22.01	22.14	0.58%
W_4	22.11	22.22	0.53%	22.17	22.28	0.53%	21.97	22.13	0.71%	21.99	22.11	0.54%
W_5	22.20	22.31	0.48%	22.24	22.34	0.45%	21.95	22.07	0.52%	21.98	22.09	0.52%
W_6	22.01	22.12	0.47%	22.05	22.20	0.71%	21.95	22.07	0.54%	—	—	—

In Table 10, "PW-method" shows the results from Ref.[20], "Bur-method" is the approximation method in Ref.[4], and "CG-method" is the method in Ref.[5]. The columns "Sim" show the throughput obtained by simulation, "App" is the approximate throughput, and "Dif" is the relative difference between the approximation and simulation of the throughput. The results show that the current method performs overall better than the PW-method and the Bur-method, and its performance is similar to that of the CG-method.

6.3. Application

In this section we perform a sensitivity analysis of the production line of Section 2. Table 11 provides the results of this sensitivity analysis, and it is based on throughput estimates obtained by the approximation method. The columns list the change in throughput when doubling the mean uptime, dividing the mean downtime by two, increasing the speed by 10%, and increasing the size of the downstream buffer by 25 products, for each of the workstations (one at a time). The rows indicate which workstation is changed. The sensitivity analysis indicates that workstation W_3 is the bottleneck: changing its parameters seems to have the strongest impact on the throughput.

In Table 12, we compare the throughput estimates of the approximation method to those of a discrete-event simulation model, the 95%confidence intervals of which have a width of at most 0.5%. The extra row (−) is the reference case with no change in workstation (or buffer). The columns "Sim" list the throughput obtained by simulation, "App" is the approximate throughput, and "Dif" is the relative difference between the approximation and simulation of the throughput. The errors are in the order of 1%.

7. CONCLUDING REMARKS

In this article, we construct an approximation method for multi-server production lines with finite buffers and exponential up- and downtimes. The distinguishing feature of this method is the use of aggregation: this is crucial to keep the subsystem models manageable. The approximation shows an overall good performance on a broad test set. More specifically, it seems to perform better for short production lines, for production lines having small downtimes with respect to the uptimes, and for production lines with larger buffers.

REFERENCES

1. Bierbooms, R. *Performance analysis of production lines*. PhD thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, **2012**.
2. Bierbooms, R.; Adan, I.; van Vuuren, M. Performance analysis of exponential production lines with fluid flow and finite buffers. *IIE Trans.*, **2012**, *44*, 1132–1144.
3. Bierbooms, R.; Adan, I.; van Vuuren, M. Approximate performance analysis of production lineswith continuous material flows and finite buffers. *Stochastic Models*, **2013**, *29*, 1–30.
4. Burman, M. *New results in flow line analysis*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, **1995**.
5. Colledani, M.; Gershwin, S. A decomposition method for approximate evaluation of continuous flow multi-stage lines with general Markovian machines. *Annals Oper. Res.*, **2013**, *209*, 5–40.
6. Colledani, M.; Tolio, T. Performance evaluation of transfer lines with general repair times and multiple failure modes. *Annals Oper. Res.*, **2011**, *182*, 31–65.
7. Dallery, Y.; Frein, Y. On decomposition methods for tandem queueing networks with blocking. *Oper. Res.*, **1993**, *41*, 386–399.
8. Dallery, Y.; Gershwin, S. B. Manufacturing flow line systems: A review of models and analytical results. *Queueing Syst. Theory Appl.*, **1992**, *12*, 3–94.
9. da Silva Soares, A.; Latouche, G. A matrix-analytic approach to fluid queues with feedback control. *Int. J. Simulation:*, **2005**, *6*, 4–12.
10. da Silva Soares, A.; Latouche, G. Matrix-analytic methods for fluid queues with finite buffers. *Perf. Eval.*, **2006**, *63*, 295–314.
11. Dallery, Y.; Bihan, H. L. A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers. *Annals Oper. Res.*, **2000**, *93*, 265–297.
12. Dallery, Y.; David, R.; Xie., X., An efficient algorithm for analysis of transfer lines with unreliable machines and random processing times. *IIE Trans.*, **1988**, *20*, 280–283.
13. Dallery, Y.; David, R.; Xie., X. Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Trans. Automatic Control*, **1989**, *34*, 943–953.
14. de Koster, M. Estimation of line efficiency by aggregation. *Int. J. Prod. Res.*, **1987**, *25*, 615–626.
15. de Koster, M. *Capacity oriented design and analysis of production systems*. PhD thesis, Eindhoven University of Technology, Department of Industrial Engineering and Management Science, **1988**.
16. Di Mascolo, M. Méthode analytique dévaluation des performances dune ligne dassemblage. *Rapport de DEA, Laboratoire dAutomatique de Grenoble*, **1988**.
17. Gershwin, S. An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking. *Oper. Res.*, **1987**, *35*, 291–305.
18. Levantesi, R.; Matta, A.; Tolio, T. Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes. *Perf. Eval.*, **2003**, *51*, 247–268.
19. Liu, X.; Buzacott, J.; Approximate models of assembly systems with finite banks. *Eur. J. Oper. Res.*, **1990**, *45*, 143–154.
20. Patchong, A.; Willaeys, D. Modeling and analysis of an unreliable flow line composed of parallel-machine stages. *IIE Transactions*, **2001**, *33*, 559–568.

21. Ramaswami, V. Matrix analytic methods for stochastic fluid flows. *Teletraffic Engineering in a Competitive World (Proceedings of the 16th International Teletraffic Congress)*, Elsevier Science, B.V., EdinburghUK, 1019–1023, 1999.
22. Tan, B.; Yeralan, S. A decomposition model for continuous materials flow production systems. *Int. J. Prod. Res.*, **1997**, *3*, 2759–2772.
23. Xie, X.-L. An efficient algorithm for performance analysis of transfer lines and its convergence. Working paper, INRIA-Lorraine, France, 1989.