IIE Transactions (2010) **42**, 422–434

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170903394348

Split and merge production systems: performance analysis and structural properties

YANG LIU and JINGSHAN LI*

Department of Electrical and Computer Engineering and Center for Manufacturing, University of Kentucky, Lexington, KY 40506, USA
E-mail: jingshan@engr.uky.edu

Received September 2008 and accepted July 2009

Production split and merge operations are widely used in many manufacturing systems to increase production capacity and variety, improve product quality, and carry out scheduling and control activities. This article presents analytical methods to analyze split and merge production systems with exponential machine reliability models, operating under circulate, strictly circulate, priority, and percentage split/merge policies. In addition to developing the recursive procedures for performance analysis, the structural properties of the systems and the impacts of routing policies on system performance are investigated.

Keywords: Split, merge, throughput, percentage, priority, circulate, strictly circulate

1. Introduction

In modern manufacturing systems, split and merge structures are often used to increase production capacity and variety, improve product quality, and carry out scheduling and control activities. The routing policies at the split or merge stations play an important role in such systems since they directly control the flow of parts. Therefore, to design and manage such systems more efficiently, the analysis of split and merge systems with different routing policies is important.

In practice, two types of split or merge are often encountered. One is known as *assembly merge* (or *disassembly split*). In this case, the assembly machine needs to take parts from all its upstream branches simultaneously and assemble them into a single product (respectively, disassemble a single part into many components and dispatch to all downstream branches simultaneously). Another type is referred to as *production merge* (or *production split*), where the merge station only takes a part from one of its upstream buffers each time (or sends a part to one of its downstream buffers). In this article, the latter type is considered. Specifically, we consider split and merge systems with exponential reliability of machines. Four frequently used split and merge policies are addressed: *circulate*, *strictly circulate*, *priority*, and *percentage*. In the circulate policy, the split machine sends the part to downstream branches in circulation by ignoring blocked branches. A similar scenario occurs in merge operations, where the merge station takes parts from all available upstream branches circularly. In the strictly circulate policy, routing is similar to the circulate policy except that blocked or starved branches cannot be ignored. In the priority policy, parts are always dispatched to the branch with the highest priority unless it is blocked. Lower-priority branches receive parts only when the split machine is blocked by a higher-priority branch. Similarly, the merge machine always takes parts from the highest available priority upstream branch. In the percentage policy, parts are dispatched to downstream branches or loaded from upstream ones following given percentages.

In recent years, several performance analysis methods have been developed to study split and merge systems (Li *et al.*, 2009). For example, Helber (2000) introduced a geometric machine reliability model for a transfer line with split operations based on a percentage routing policy. A three-station merge system with continuous material flow and exponential machine up- and downtimes was studied by Tan (2001). The merge station was assumed to have a higher processing rate than the combined processing rates of the upstream stations; however, no explicit discussion of merge policies was presented. Helber and Mehrtens (2003) considered a similar system but allowed arbitrary rates in all stations and assumed a priority merge policy for such transitions. A discrete three-machine merge system with geometric reliability of machines was studied by Diamantidis *et al.* (2004) and Diamantidis and Papadopoulos (2006). A

*Corresponding author

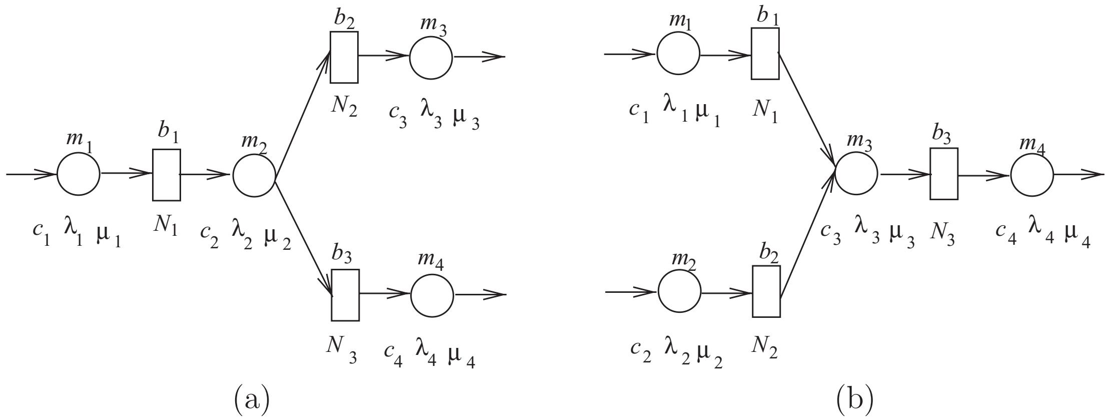


Fig. 1. (a) A split system and (b) a merge system.

priority merge policy was assumed in both papers when the shared buffer was in transit to full state. Multiple products systems were studied by Li and Huang (2005), Colledani *et al.* (2005), and Colledani *et al.* (2008), with exponential and geometric machine reliability models being considered, respectively. In these models, different products were processed separately at the dedicated machines or buffers. Another direction of study related to split and merge systems focuses on rework loops (Li 2004b, 2004c; Biller *et al.*, 2010). In such systems, repaired parts are typically assumed to have a higher priority at merge station to avoid deadlock. Parallel systems were investigated in Li (2004a) with parallel lanes being split from a common buffer and then merged into another shared buffer. Equal probabilities for all branches in split and merge were assumed in this study.

In spite of the above work, split and merge systems with different policies have not been studied thoroughly. In particular, an in-depth study to quantify and compare the results of routing policies and investigate the structural properties of the system is needed. Such a study can enable us to understand the impact of different policies and finally provide guidance to control the flow of parts in operations. A preliminary study on Bernoulli split and merge systems with several routing policies was carried out by Liu and Li (2009). This article is intended to extend that work to more general cases, namely, exponential machine reliability models with more routing policies. The main contribution of this article is in developing analytical models to study the split and merge systems with different routing policies, comparing their impacts and investigating structural properties.

The remainder of this article is structured as follows: Section 2 formulates the problem to be addressed. The modeling and analysis methods are introduced in Section 3. Section 4 is devoted to structural property discussions and policy comparisons. Section 5 extends the study to larger split and merge systems. Finally, conclusions are given in Section 6. All the proofs are provided in the Appendix.

2. Problem formulation

In this article, we consider a typical four-machine split (or merge) system, whose layout is shown in Fig. 1(a) (respectively, Fig. 1(b)). Here the circles represent the machines and the rectangles are the buffers. The following assumptions address the machines, the buffers and their interactions.

1. Each machine m_i, $i = 1, \ldots, 4$, has two states: up and down. When it is up, it is capable of processing parts with capacity c_i parts/unit of time. When the machine is down, no production takes place.
2. The up- and downtimes of machine m_i are random variables exponentially distributed with parameters λ_i and μ_i, respectively; i.e., λ_i and μ_i are failure and repair rates, respectively.
3. Each buffer b_k, $k = 1, 2, 3$, has capacity N_k, $0 < N_k < \infty$.
4. A machine is blocked at time t if it is up, its downstream buffer is full, and the downstream machine fails to take any work from the buffer at time t. Machines m_3 and m_4 are never blocked in a split system, while m_4 is never blocked in a merge system.
5. A machine is starved at time t if it is up, its upstream buffer is empty, and the upstream machine fails to put any work into the buffer at time t. In a split system, machine m_1 is never starved, and in a merge system, machines m_1 and m_2 are never starved.
6. Machine m_2 in a split system (or m_3 in a merge system) will send parts to downstream buffers b_2 and b_3 (respectively, take parts from upstream buffers b_1 and b_2) based on:
   6.1. *Priority policy.* Buffer b_2 has a higher priority; i.e., m_2 will keep sending parts to b_2 whenever it has

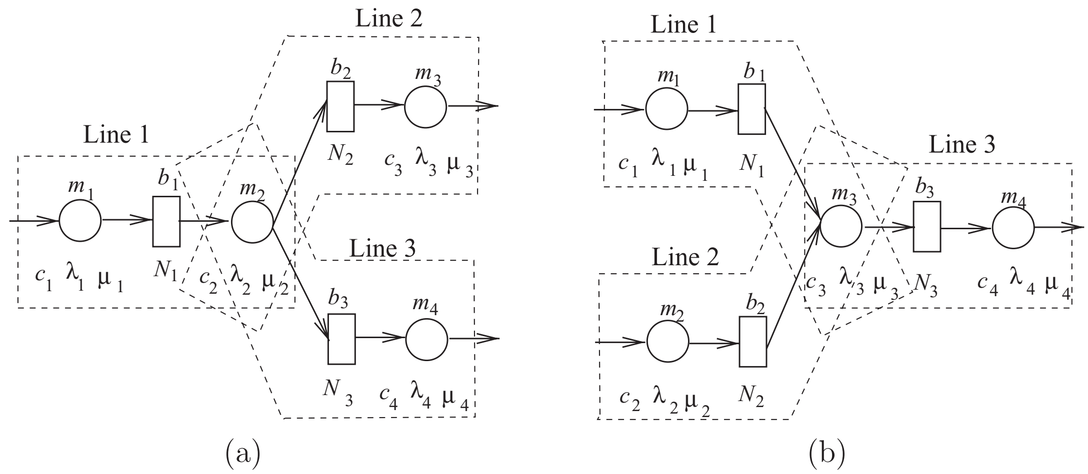


Fig. 2. Overlapping decomposition of (a) split and (b) merge systems.

space (respectively, buffer b_1 has a higher priority, and m_3 takes parts from b_1 if it has available parts). m_2 sends parts to b_3 only when it is blocked by b_2 (respectively, m_3 takes parts from b_2 only when it is starved by b_1).

6.2. *Circulate policy.* Machine m_2 will send a part to buffers b_2 and b_3 circularly if it is not blocked by both buffers (respectively, m_3 takes parts from b_1 and b_2 circularly when it is not starved by both). If it is blocked by one buffer, m_2, will send the part to another buffer (respectively, m_3 will take a part from another buffer if it is starved by a given buffer).

6.3. *Strictly circulate policy.* Machine m_2 will send a part to buffers b_2 and b_3 (respectively, m_3 takes parts from b_1 and b_2) circularly. If it is blocked by a given buffer, m_2, will wait until that buffer is available (respectively, m_3 will wait until the buffer has available parts if it is starved by it).

6.4. *Percentage policy (split only).* Machine m_2 will send a part to buffers b_1 and b_2 based on predesigned percentages, $\alpha \times 100\%$ to b_1 and $(1 - \alpha) \times 100\%$ to b_2.

Remark 1. In practice, priority, circulate, and strictly circulate policies are relatively easy to implement and are used more often than other policies. For example, in rework and re-entrant lines, the priority policy is typical, while in parallel operations, the circulate policy is often used. The strictly circulate policy is usually adopted to keep the desired sequence. In the literature, the percentage policy is also studied. However, it is less popular in practice due to implementation difficulties. In addition, since the percentage merge is less frequently encountered, only the percentage split policy is discussed in this work. The performance measure for a merge systems can be obtained using a reversibility property.

Let *TP* be the throughput of the split (or merge) system; i.e., the average number of parts produced by the last machines m_3 and m_4 (respectively, m_4 in the merge case) per unit of time. The problem addressed in this article is as follows: *Given the production system of assumptions 1 to 6 develop a method to evaluate the system throughput as a function of the system parameters and investigate its structural properties.*

A solution to the problem is presented in the next two sections. Extensions to systems with multiple-branch split/merge machines and longer lines are provided in Section 5.

3. Performance analysis

3.1. *Overlapping decomposition*

Since the split (or merge) machine has to allocate its capacity to downstream (respectively, upstream) branches and all machines and buffers in the system interfere with each other, exact analysis seems impossible. Therefore, an approximation method based on *overlapping decomposition* (Li, 2005) is pursued (see Fig. 2 for illustration).

Consider the split system illustrated in Fig. 2(a). We decompose the system into three overlapped serial lines, where m_2 is the overlapping machine. Assume the probabilities that machine m_2 is blocked by b_2 and b_3 are known, modify m_2 to m_2' to take into account these effects, we obtain the first overlapped serial line, denoted as Line 1 (m_1, b_1, and m_2'). Then, the probability that m_2 is starved by b_1 can be calculated. Next, we modify m_2 to include this starvation probability. In addition, the capacity of m_2 needs to be modified to represent the capacity that is only dedicated to branch b_2, and m_3. Thus, we obtain m_2'' and the second overlapped line, referred to as Line 2 (m_2'', b_2, and m_3). Again, the probability that m_2 is blocked by b_2 can be calculated. Analogously, taking into account the starvation probability from b_1 and only including m_2's capacity allocated to branch b_3 and m_4, we modify m_2 to

m_2''' and obtain Line 3 (m_2''', b_3, and m_4). Then, the probability that m_2 is blocked by b_3 can be computed. Next, using these blockage probabilities, we repeat the analysis to obtain m_2' and Line 1 again, then continue to Lines 2 and 3. The blockage and starvation probabilities are updated and the procedure is repeated anew. When the procedure is convergent, the production rates of Lines 1 to 3 are obtained.

For the merge system in Fig. 2(b), a similar idea can be applied but with m_3 being the overlapping machine. Line 1 consists of m_1, b_1, and modified machine m_3', which takes into account the effect of the blockage of b_3 and capacity allocation to branch b_1 and m_1 only. Line 2 has m_2, b_2, and pseudomachine m_3'', which considers the blockage of b_3 and capacity distributed to branch b_2 and m_2. Including starvation probabilities from b_1 and b_2, we modify m_3 into m_3''' and obtain Line 3 (m_3''', b_3, and m_4). Then, iteration procedures are introduced to update the blockage and starvation probabilities of m_3 recursively.

3.2. *Priority policy*

3.2.1. *Recursive procedures*

Split system: Consider the split system in Fig. 2(a). Assume that buffer b_2 has a higher priority than b_3. In Line 2, m_2 is always available to b_2 when it is not starved, then m_2's capacity c_2 is multiplied by the probability that b_1 is not empty. However, in Line 3, m_2 is available to b_3 only when it is not starved by b_1 but blocked by b_2; thus, the probabilities that b_2 is full and b_1 is not empty are used to modify c_2. Finally, in Line 1, c_2 is multiplied by the probabilities that b_2 and b_3 are not full simultaneously. Let $TP(\tilde{c}_1, \tilde{\lambda}_1, \tilde{\mu}_1, \tilde{c}_2, \tilde{\lambda}_2, \tilde{\mu}_2, \tilde{N}_1)$ denote the operator to calculate the throughput of a two-machine line, where $\tilde{c}_i$, $\tilde{\lambda}_i$, and $\tilde{\mu}_i$, $i = 1, 2$, are the capacity, failure, and repair rates of machine i, respectively, and $\tilde{N}_1$ represents the buffer capacity (see Appendix A for formulas of $TP(\cdot)$). Then, the recursive procedure is introduced as

Procedure 1.

Line 1
$$\begin{aligned} c_2'(s+1) &= c_2(1 - \widehat{X}_{2N_2}(s)\widehat{X}_{3N_3}(s)), \\ \widehat{TP}_{\mathrm{s,p,1}}(s+1) &= TP(c_1, \lambda_1, \mu_1, c_2'(s+1), \\ &\qquad \lambda_2, \mu_2, N_1), \\ \widehat{X}_{10}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s,p,1}}(s+1)}{c_2'(s+1)e_2}, \end{aligned} \tag{1}$$

Line 2
$$\begin{aligned} c_2''(s+1) &= c_2(1 - \widehat{X}_{10}(s+1)), \\ \widehat{TP}_{\mathrm{s,p,2}}(s+1) &= TP(c_2''(s+1), \\ &\qquad \lambda_2, \mu_2, c_3, \lambda_3, \mu_3, N_2), \\ \widehat{X}_{2N_2}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s,p,2}}(s+1)}{c_2''(s+1)e_2}, \end{aligned} \tag{2}$$

Line 3
$$\begin{aligned} c_2'''(s+1) &= c_2\widehat{X}_{2N_2}(s+1)(1 - \widehat{X}_{10}(s+1)), \\ \widehat{TP}_{\mathrm{s,p,3}}(s+1) &= TP(c_2'''(s+1), \\ &\qquad \lambda_2, \mu_2, c_4, \lambda_4, \mu_4, N_3), \\ \widehat{X}_{3N_3}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s,p,3}}(s+1)}{c_2'''(s+1)e_2}, \\ s &= 0, 1, 2, \ldots, \\ \widehat{X}_{2N_2}(0) &= \widehat{X}_{3N_3}(0) = 0, \end{aligned} \tag{3}$$

where $\widehat{X}_{10}(s)$, $\widehat{X}_{2N_2}(s)$, $\widehat{X}_{3N_3}(s)$ denote the estimates of the probabilities that b_1 is empty, b_2 and b_3 are full at iteration s, respectively,

$$e_i = \frac{\mu_i}{\lambda_i + \mu_i}, \quad i = 1, 2, 3,$$

and $\widehat{TP}_{\mathrm{s,p},i}(s)$ is the throughput estimate of line i in split system with priority policy at the sth iteration.

Merge system: Assuming that buffer b_1 has a higher priority than b_2. Then, m_3 in Line 1 is available when it is not blocked; i.e., the probability that b_3 is not full is used to modify c_3. In Line 2, the additional probability of b_1 being empty is used to multiply c_3 due to a lower priority of b_2. Finally, m_3 is modified to exclude the case in which buffers b_1 and b_2 are empty in Line 3. Let $\widehat{TP}_{\mathrm{m,p},i}(s)$ be the throughput estimate of line i in merge system with priority policy at iteration s, we have:

Procedure 2.

Line 1
$$\begin{aligned} c_3'(s+1) &= c_3(1 - \widehat{X}_{3N_3}(s)), \\ \widehat{TP}_{\mathrm{m,p,1}}(s+1) &= TP(c_1, \lambda_1, \mu_1, c_3'(s+1), \\ &\qquad \lambda_3, \mu_3, N_1), \\ \widehat{X}_{10}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,p,1}}(s+1)}{c_3'(s+1)e_3}, \end{aligned} \tag{4}$$

Line 2
$$\begin{aligned} c_3''(s+1) &= c_3\widehat{X}_{10}(s+1)(1 - \widehat{X}_{3N_3}(s)), \\ \widehat{TP}_{\mathrm{m,p,2}}(s+1) &= TP(c_2, \lambda_2, \mu_2, c_3''(s+1), \\ &\qquad \lambda_3, \mu_3, N_2), \\ \widehat{X}_{20}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,p,2}}(s+1)}{c_3''(s+1)e_3}, \end{aligned} \tag{5}$$

Line 3
$$\begin{aligned} c_3'''(s+1) &= c_3(1 - \widehat{X}_{10}(s+1)\widehat{X}_{20}(s+1)), \\ \widehat{TP}_{\mathrm{m,p,3}}(s+1) &= TP(c_3'''(s+1), \\ &\qquad \lambda_3, \mu_3, c_4, \lambda_4, \mu_4, N_3), \\ \widehat{X}_{3N_3}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,p,3}}(s+1)}{c_3'''(s+1)e_3}, \\ s &= 0, 1, 2, \ldots, \\ \widehat{X}_{3N_3}(0) &= 0. \end{aligned} \tag{6}$$

3.2.2. *Convergence*

Let $\widehat{TP}_{\mathrm{s,p},i}$, $\widehat{TP}_{\mathrm{m,p},i}$, $i = 1, 2, 3$, denote the throughputs obtained for Line i if Procedures 1 and 2 are convergent, respectively. It is shown below that these procedures lead to convergent results.

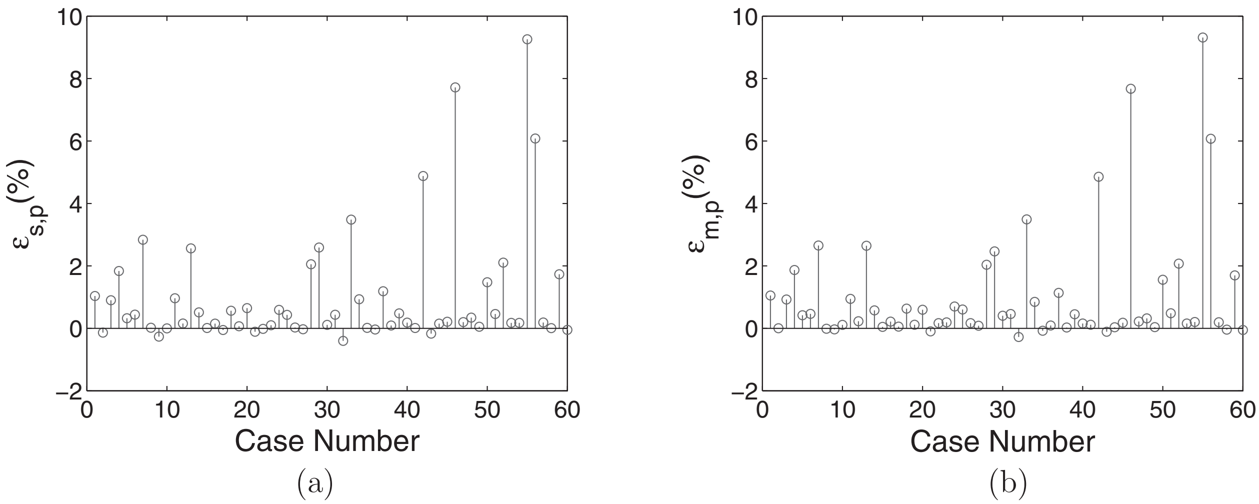


Fig. 3. Accuracy of Procedures 1 and 2 for (a) the split system and (b) the merge system.

Theorem 1. *Under assumptions* 1 *to* 6, *Procedures* 1 *and* 2 *are convergent*; *therefore*,

$$\lim_{s\to\infty} \widehat{TP}_{s,p,i}(s) := \widehat{TP}_{s,p,i},$$
$$\lim_{s\to\infty} \widehat{TP}_{m,p,i}(s) := \widehat{TP}_{m,p,i},\ i = 1, 2, 3. \quad (7)$$

In addition, *the steady-state equations have unique solutions.*

Proof. See Appendix B. ■

Therefore, we obtain the estimates of the throughput, $\widehat{TP}_{s,p}$ and $\widehat{TP}_{m,p}$, for split and merge systems with priority policy, respectively, where:

$$\widehat{TP}_{s,p} = \widehat{TP}_{s,p,2} + \widehat{TP}_{s,p,3},\ \ \widehat{TP}_{m,p} = \widehat{TP}_{m,p,3}. \quad (8)$$

3.2.3. *Accuracy*

The accuracy of the estimation is investigated numerically. Specifically, we randomly and equiprobably select machine and buffer parameters from the following sets, and construct 60 split systems and then 60 merge systems by reversing the lines.

$$\begin{aligned} &e_i \in [0.75, 0.95], \quad i = 1, \ldots, 4,\\ &T_{\text{down},i} \in [2, 10], \quad i = 1, \ldots, 4,\\ &c_i \in [1, 1.2], \quad i = 1, 2 \text{ in split systems},\\ &\qquad i = 3, 4 \text{ in merge systems},\\ &c_i \in [0.6, 0.8], \quad i = 3, 4 \text{ in split systems},\\ &\qquad i = 1, 2 \text{ in merge systems},\\ &N_i \in [1, 3] \times T_{\text{down},i}, \quad i = 1, \ldots, 4, \end{aligned} \quad (9)$$

where $T_{\text{down},i} = 1/\mu_i$ is machine m_i's average downtime, and λ_i can be calculated from e_i.

Both an analytical method using Procedures 1 and 2 and a simulation approach using *Simul8* (Haige and Paige, 2001) are pursued to evaluate the throughput of each line. Ten thousand time units of warmup period are assumed, and the next 100 000 units are used to collect steady-state statistics. Twenty replications are carried out to obtain the average production rate, with 95% confidence intervals typically ranging around ±0.001. The computation time for Procedures 1 and 2 (and all subsequent analytical procedures) is less than a fraction of second, and for simulation it is around 5 minutes on a PC with a 3.4 GHz processor and a 2 GB RAM. The differences between analytical and simulation results are evaluated as

$$\epsilon_{s,p} = \frac{\widehat{TP}_{s,p} - TP_{s,p}}{TP_{s,p}} \times 100\%,$$
$$\epsilon_{m,p} = \frac{\widehat{TP}_{m,p} - TP_{m,p}}{TP_{m,p}} \times 100\%, \quad (10)$$

where $TP_{s,p}$ and $TP_{m,p}$ are the throughputs obtained by simulation for a priority policy in split and merge systems, respectively.

The results of this investigation are illustrated in Fig. 3. The 0.25, 0.5, and 0.75 quantiles are 0.11, 0.3, and 1%, respectively. It is shown that in most cases we studied, the errors are within 3%, with a few exceptions that can reach up to 9%. Even for systems with extremely long downtimes and small buffers, the errors do not increase significantly (with the largest one less than 10%). Therefore, Procedures 1 and 2 provide a relatively accurate approximation for system throughputs.

Remark 2. In additional to throughput, the accuracy for work-in-process is also investigated. The accuracy of the estimates is evaluated by

$$\epsilon_{WIP_i} = \frac{|\widehat{WIP}_i - WIP_i|}{N_i} \times 100\%, \quad i = 1, 2, 3,$$

where WIP_i and $\widehat{WIP}_i$ are the work-in-process measures obtained by simulation and analytical procedures, respectively, and N_i is the buffer capacity. It is observed that the accuracy is close to that of throughput estimation (with an average accuracy within 2%, and the largest one up to 10%). Similar results are were also obtained for subsequent policies.

3.3. *Circulate policy*

Split system: The rationale behind the modification of m_2 is that, in Line 1, m_2 is available to b_1 if it is not blocked by either b_2 or b_3. In Line 2, when m_2 is not starved, it is available to b_2 50% of the time if b_3 is not full and 100% of the time otherwise. A similar argument also applies to Line 3. Thus, c_2 is modified by the sum of 0.5 Prob{b_3 is not full} and Prob{b_3 is full}. Then, we have:

Procedure 3.

$$\begin{aligned}
\text{Line 1} \quad & c_2'(s+1) = c_2(1 - \widehat{X}_{2N_2}(s)\widehat{X}_{3N_3}(s)),\\
& \widehat{TP}_{\text{s,c,1}}(s+1) = TP(c_1, \lambda_1, \mu_1, c_2'(s+1), \lambda_2, \mu_2, N_1), \qquad (11)\\
& \widehat{X}_{10}(s+1) = 1 - \frac{\widehat{TP}_{\text{s,c,1}}(s+1)}{c_2'(s+1)e_2},\\
\text{Line 2} \quad & c_2''(s+1) = c_2(0.5(1 - \widehat{X}_{3N_3}(s)) + \widehat{X}_{3N_3}(s))(1 - \widehat{X}_{10}(s+1)),\\
& \widehat{TP}_{\text{s,c,2}}(s+1) = TP(c_2''(s+1), \lambda_2, \mu_2, c_3, \lambda_3, \mu_3, N_2), \qquad (12)\\
& \widehat{X}_{2N_2}(s+1) = 1 - \frac{\widehat{TP}_{\text{s,c,2}}(s+1)}{c_2''(s+1)e_2},\\
\text{Line 3} \quad & c_2'''(s+1) = c_2(0.5(1 - \widehat{X}_{2N_2}(s+1)) + \widehat{X}_{2N_2}(s+1))(1 - \widehat{X}_{10}(s+1)),\\
& \widehat{TP}_{\text{s,c,3}}(s+1) = TP([c_2'''(s+1), \lambda_2, \mu_2, c_4, \lambda_4, \mu_4, N_3), \qquad (13)\\
& \widehat{X}_{3N_3}(s+1) = 1 - \frac{\widehat{TP}_{\text{s,c,3}}(s+1)}{c_2'''(s+1)e_2},\\
& s = 0, 1, 2, \ldots,\\
& \widehat{X}_{2N_2}(0) = \widehat{X}_{3N_3}(0) = 0.
\end{aligned}$$

Merge system: A similar procedure can be developed for merge systems by considering the starvation probabilities of m_3. Thus, we obtain:

Procedure 4.

$$\begin{aligned}
\text{Line 1} \quad & c_3'(s+1) = c_3(0.5(1 - \widehat{X}_{20}(s)) + \widehat{X}_{20}(s))(1 - \widehat{X}_{3N_3}(s)),\\
& \widehat{TP}_{\text{m,c,1}}(s+1) = TP(c_1, \lambda_1, \mu_1, c_3'(s+1), \lambda_3, \mu_3, N_1), \qquad (14)\\
& \widehat{X}_{10}(s+1) = 1 - \frac{\widehat{TP}_{\text{m,p,1}}(s+1)}{c_3'(s+1)e_3},\\
\text{Line 2} \quad & c_3''(s+1) = c_3(0.5(1 - \widehat{X}_{10}(s+1)) + \widehat{X}_{10}(s+1))(1 - \widehat{X}_{3N_3}(s)),\\
& \widehat{TP}_{\text{m,c,2}}(s+1) = TP(c_2, \lambda_2, \mu_2, c_3''(s+1), \lambda_3, \mu_3, N_2), \qquad (15)\\
& \widehat{X}_{20}(s+1) = 1 - \frac{\widehat{TP}_{\text{m,p,2}}(s+1)}{c_3''(s+1)e_3},\\
\text{Line 3} \quad & c_3'''(s+1) = c_3(1 - \widehat{X}_{10}(s+1)\widehat{X}_{20}(s+1)),\\
& \widehat{TP}_{\text{m,c,3}}(s+1) = TP(c_3'''(s+1), \lambda_3, \mu_3, c_4, \lambda_4, \mu_4, N_3), \qquad (16)\\
& \widehat{X}_{3N_3}(s+1) = 1 - \frac{\widehat{TP}_{\text{m,p,3}}(s+1)}{c_3'''(s+1)e_3},\\
& s = 0, 1, 2, \ldots,\\
& \widehat{X}_{20}(0) = 0, \quad \widehat{X}_{3N_3}(0) = 1.
\end{aligned}$$

Again, both procedures lead to convergent results. Let $\widehat{TP}_{\text{s,c,i}}$, $\widehat{TP}_{\text{m,c,i}}$, $i = 1, 2, 3$, denote the throughput estimates for Line i when Procedures 3 and 4 are convergent, respectively.

Theorem 2. *Under assumptions* 1 *to* 6, *Procedures* 3 *and* 4 *are convergent; that is,*

$$\lim_{s\to\infty} \widehat{TP}_{\text{s,c,i}}(s) := \widehat{TP}_{\text{s,c,i}},$$
$$\lim_{s\to\infty} \widehat{TP}_{\text{m,c,i}}(s) := \widehat{TP}_{\text{m,c,i}}, \; i = 1, 2, 3. \qquad (17)$$

Moreover, the steady-state equations have unique solutions.

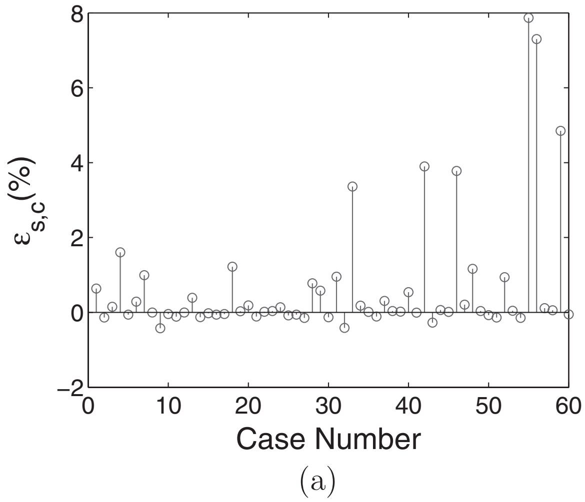
Proof. See Appendix B. ∎

Therefore, the throughput estimates, $\widehat{TP}_{\text{s,c}}$, $\widehat{TP}_{\text{m,c}}$ for split and merge systems under a circulate policy, respectively, can be calculated as

$$\widehat{TP}_{\text{s,c}} = \widehat{TP}_{\text{s,c,2}} + \widehat{TP}_{\text{s,c,3}},$$
$$\widehat{TP}_{\text{m,c}} = \widehat{TP}_{\text{m,c,3}}. \qquad (18)$$

The accuracy of estimation (18) was also investigated numerically. The same split and merge systems as in Section 3.2 are adopted for accuracy analysis using both Procedures 3 and 4 and simulations. The differences between analytical and simulation results are introduced as

$$\epsilon_{\text{s,c}} = \frac{\widehat{TP}_{\text{s,c}} - TP_{\text{s,c}}}{TP_{\text{s,c}}} \times 100\%,$$
$$\epsilon_{\text{m,c}} = \frac{\widehat{TP}_{\text{m,c}} - TP_{\text{m,c}}}{TP_{\text{m,c}}} \times 100\%, \qquad (19)$$



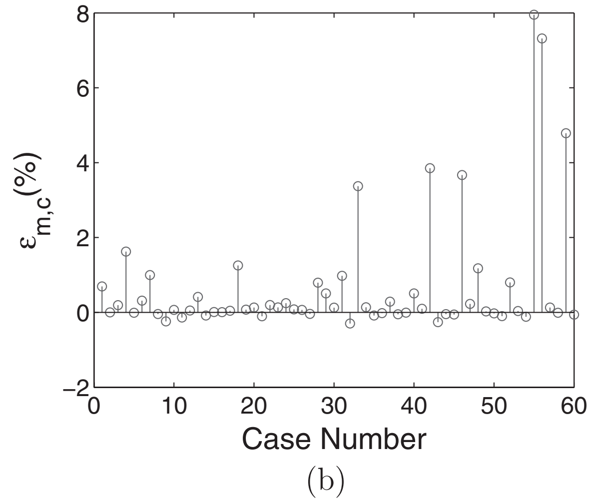


Fig. 4. Accuracy of Procedures 3 and 4 for (a) the split system and (b) the merge system.

where $TP_{\mathrm{s,c}}$ and $TP_{\mathrm{m,c}}$ are the throughputs obtained by simulation for the circulate policy in split and merge systems, respectively.

Figure 4 illustrates the results of these tests. The 0.25, 0.5, and 0.75 quantiles are 0.05, 0.13, and 0.51%, respectively. In most cases we studied, the errors are less than 2%, with a few cases reaching up to 8%. Therefore, Procedures 1 and 2 provide relatively precise estimates.

3.4. *Percentage policy*

The percentage policy implies that, in a split system (Fig. 1(a)), the parts flow into different downstream branches based on given percentages. However, due to possible blockages, the split station may need to wait until the downstream buffer has available space. Therefore, if we directly allocate the capacity of the split station based on the given percentage, the final percentage of parts flowing into different branches may not be the same as expected. To ensure that the final products consist of $100 \times \alpha\%$ parts produced by m_3 and $100 \times (1-\alpha)\%$ by m_4, which agrees with the expectation of the percentage policy, a new percentage of capacity allocation needs to be determined. Assume that $\beta \times 100\%$ of parts are intended to be sent to buffer b_2 and $(1-\beta) \times 100\%$ to b_3 by machine m_2, this implies that m_2 has a probability β to be available on parts to buffer b_2, and probability $1-\beta$ on parts to b_3. Then, after possible blockages, the actual probability of sending parts to b_2 and b_3 will be α and $1-\alpha$, respectively. Therefore, we need:

$$\beta(1-\widehat{X}_{10}) = \alpha(1-\beta\widehat{X}_{2N_2} - (1-\beta)\widehat{X}_{3N_3})(1-\widehat{X}_{10}),$$

i.e., products to branch b_2 are $\alpha \times 100\%$ of the products going through b_1 (when it is not blocked), which leads to

$$\beta = \frac{(1-\widehat{X}_{3N_3})\alpha}{1+\widehat{X}_{2N_2}\alpha - \widehat{X}_{3N_3}\alpha}. \tag{20}$$

Then, c_2 is allocated to Lines 2 and 3 with probabilities β and $1-\beta$, respectively. The recursive procedure for the percentage split approach is introduced as follows:

Procedure 5.

$$\beta(s+1) = \frac{(1-\widehat{X}_{3N_3}(s))\alpha}{1+\widehat{X}_{2N_2}(s)\alpha - \widehat{X}_{3N_3}(s)\alpha},$$

Line 1

$$\begin{aligned} c_2'(s+1) &= c_2(1-\beta(s+1)\widehat{X}_{2N_2}(s) \\ &\quad -(1-\beta(s+1))\widehat{X}_{3N_3}(s)), \\ \widehat{TP}_{\mathrm{s},\%,1}(s+1) &= TP(c_1, \lambda_1, \mu_1, c_2'(s+1), \\ &\quad \lambda_2, \mu_2, N_1), \\ \widehat{X}_{10}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s},\%,1}(s+1)}{c_2'(s+1)e_2}, \end{aligned} \tag{21}$$

Line 2

$$\begin{aligned} c_2''(s+1) &= \beta(s+1)c_2(1-\widehat{X}_{10}(s+1)), \\ \widehat{TP}_{\mathrm{s},\%,2}(s+1) &= TP(c_2''(s+1), \\ &\quad \lambda_2, \mu_2, c_3, \lambda_3, \mu_3, N_2), \\ \widehat{X}_{2N_2}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s},\%,2}(s+1)}{c_2''(s+1)e_2}, \end{aligned} \tag{22}$$

Line 3

$$\begin{aligned} c_2'''(s+1) &= (1-\beta(s+1)) \\ &\quad c_2(1-\widehat{X}_{10}(s+1)), \\ \widehat{TP}_{\mathrm{s},\%,3}(s+1) &= TP([c_2'''(s+1), \\ &\quad \lambda_2, \mu_2, c_4, \lambda_4, \mu_4, N_3), \\ \widehat{X}_{3N_3}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{s},\%,3}(s+1)}{c_2'''(s+1)e_2}, \\ s &= 0, 1, 2, \ldots, \\ \widehat{X}_{2N_2}(0) &= \widehat{X}_{3N_3}(0) = 0. \end{aligned} \tag{23}$$

However, unlike the priority and circulate cases, an analytical proof of the convergence of Procedure 5 is not currently available. Therefore, we justify the convergence numerically. In all the examples we tested, the procedure converges. Thus, we formulate:

Table 1. Accuracy (in percent) of Procedure 5

	Quantile		
α (%)	*0.25*	*0.5*	*0.75*
10	0.08	0.26	1.20
30	0.77	2.30	3.49
50	0.55	2.01	4.74

Numerical Fact 1. Under assumptions 1 to 6, Procedure 5 is convergent. Thus,

$$\lim_{s\to\infty} \widehat{TP}_{\mathrm{s},\%,\mathrm{i}}(s) := \widehat{TP}_{\mathrm{s},\%,\mathrm{i}}, \qquad i = 1, 2, 3. \tag{24}$$

Then, under this fact, the steady-state equations have unique solutions. Therefore, the estimate of throughput for percentage policy, $\widehat{TP}_{\mathrm{s},\%}$, for the split system in steady-state conditions can be obtained as

$$\widehat{TP}_{\mathrm{s},\%} = \widehat{TP}_{s,\%,1} = \widehat{TP}_{s,\%,2} + \widehat{TP}_{s,\%,3}. \tag{25}$$

Similar to the priority and circulate policies, define $TP_{\mathrm{s},\%}$ as the throughput by simulation, and let

$$\epsilon_{\mathrm{s},\%} = \frac{\widehat{TP}_{\mathrm{s},\%} - TP_{\mathrm{s},\%}}{TP_{\mathrm{s},\%}} \cdot 100\%. \tag{26}$$

Numerical experiments were carried out to investigate the accuracy of Procedure 5 with α taken to be 10, 30, and 50%. The 60 lines defined in previous subsections were used in the tests. The results are shown in Fig. 5 and Table 1. As before, most cases result in errors less than 5%, but there exist a few cases where the errors can reach as high are 15%. Considering that the data collected on the factory floor may be subject to 5–10% error, in general, Procedure 5 presents an acceptable accuracy.

3.5. *Strictly circulate policy*

Split system: The strictly circulate policy mandates that the split machine sends parts to downstream branches in a circular mode without ignoring the blocked ones, which implies that a machine waits until the downstream buffer is available. Such a policy is similar to a percentage policy with $\alpha = 0.5$. However, whereas percentage policy one may have the possibility that two consecutive parts will be sent to the same branch, in the strictly circulate policy, this will never happen, and the final distribution of part flows will be identical. Therefore, we use Procedure 5 to estimate the throughput of the split system by assuming $\alpha = 0.5$. Denote $\widehat{TP}_{\mathrm{s,sc},i}(s)$ as the throughput of Line i under a strictly circulate policy at iteration i.

Merge system: While the percentage policy approved is seldom used for production merge, the strictly circulate policy is used in many merge systems. For example, if a desired production sequence needs to be followed, a strictly circulate policy can be adopted. Similar to a split system, a new allocation of capacity (rather than 0.5) needs to be computed in order to ensure the strict circulation. Again, the throughput estimate of Line i under a strictly circulate merge policy at the sth iteration is denoted as $\widehat{TP}_{\mathrm{m,sc},i}(s)$.

Procedure 6.

$$\beta(s+1) = \frac{0.5(1-\widehat{X}_{20}(s))}{1+0.5\widehat{X}_{10}(s)-0.5\widehat{X}_{20}(s)},$$

Line 1

$$\begin{aligned} c_3'(s+1) &= \beta(s+1)c_3(1-\widehat{X}_{3N_3}(s+1)),\\ \widehat{TP}_{\mathrm{m,sc},1}(s+1) &= TP(c_1, \lambda_1, \mu_1, c_3'(s+1),\\ &\quad \lambda_3, \mu_3, N_1), \qquad (27)\\ \widehat{X}_{10}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,sc},1}(s+1)}{c_3'(s+1)e_3},\end{aligned}$$

Line 2

$$\begin{aligned} c_3''(s+1) &= (1-\beta(s+1))\\ &\quad c_3(1-\widehat{X}_{3N_3}(s+1)),\\ \widehat{TP}_{\mathrm{m,c},2}(s+1) &= TP(c_2, \lambda_2, \mu_2, c_3''(s+1),\\ &\quad \lambda_3, \mu_3, N_2), \qquad (28)\\ \widehat{X}_{20}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,sc},2}(s+1)}{c_3''(s+1)e_3},\end{aligned}$$

Line 3

$$\begin{aligned} c_3'''(s+1) &= c_3(1-\beta(s+1)\widehat{X}_{10}(s)\\ &\quad -(1-\beta(s+1))\widehat{X}_{20}(s)),\\ \widehat{TP}_{\mathrm{m,sc},3}(s+1) &= TP([c_3'''(s+1),\\ &\quad \lambda_3, \mu_3, c_4, \lambda_4, \mu_4, N_3), \qquad (29)\\ \widehat{X}_{3N_3}(s+1) &= 1 - \frac{\widehat{TP}_{\mathrm{m,sc},3}(s+1)}{c_3'''(s+1)e_3},\\ s &= 0, 1, 2, \ldots,\\ \widehat{X}_{10}(0) &= \widehat{X}_{20}(0) = 0.\end{aligned}$$

The convergence of Procedures 5 ($\alpha = 0.5$) and 6 is justified through Numerical Fact 1 and the uniqueness of the solution follows. Then,

$$\begin{aligned} &\lim_{s\to\infty} \widehat{TP}_{\mathrm{s,sc},i}(s) = \widehat{TP}_{\mathrm{s,sc},i},\\ &\lim_{s\to\infty} \widehat{TP}_{\mathrm{m,sc},i}(s) = \widehat{TP}_{\mathrm{m,sc},i}, i = 1, 2, 3. \end{aligned} \tag{30}$$

The estimates of system throughput are

$$\widehat{TP}_{\mathrm{s,sc}} = \widehat{TP}_{\mathrm{s,sc},2} + \widehat{TP}_{\mathrm{s,sc},3}, \quad \widehat{TP}_{\mathrm{m,sc}} = \widehat{TP}_{\mathrm{m,sc},1} + \widehat{TP}_{\mathrm{m,sc},2}. \tag{31}$$

Let $TP_{\mathrm{s,sc}}$ and $TP_{\mathrm{m,sc}}$ be the throughputs of the split and merge systems using simulations, respectively. Define the accuracy of the estimates as

$$\begin{aligned} \epsilon_{\mathrm{s,sc}} &= \frac{\widehat{TP}_{\mathrm{s,sc}} - TP_{\mathrm{s,sc}}}{TP_{\mathrm{s,sc}}} \times 100\%,\\ \epsilon_{\mathrm{m,sc}} &= \frac{\widehat{TP}_{\mathrm{m,sc}} - TP_{\mathrm{m,sc}}}{TP_{\mathrm{m,sc}}} \times 100\%. \end{aligned} \tag{32}$$

The same 60 lines used before were tested for accuracy. Figure 6 illustrates the results. The 0.25, 0.5, and 0.75 quantiles are 0.33, 0.96, and 2.19%, respectively. Again, in most

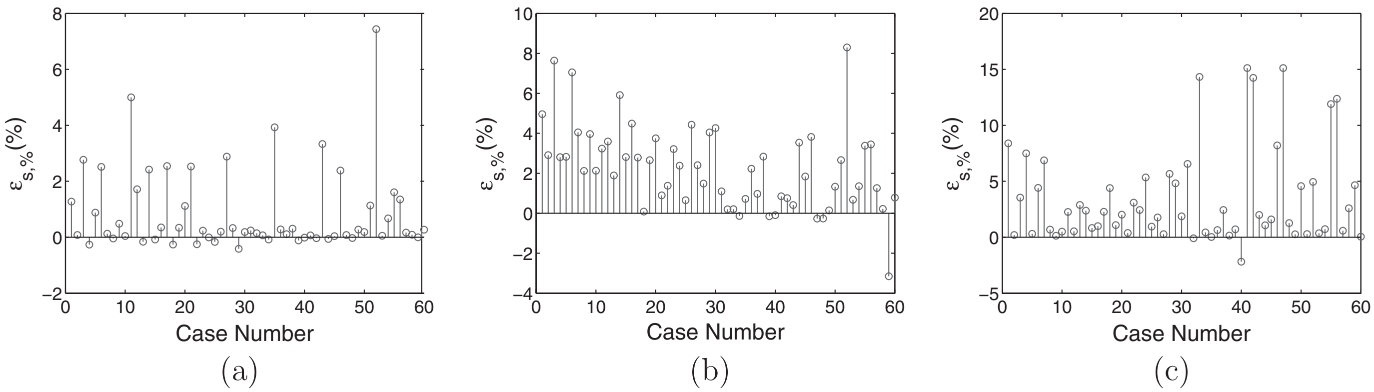


Fig. 5. Accuracy of Procedure 5 for (a) $\alpha = 10\%$, (b) $\alpha = 30\%$, and (c) $\alpha = 50\%$.

cases, the accuracy is within 4%, with a few exceptional cases reaching up to 15%. Therefore, we conclude that both procedures can be used for performance estimation and subsequent analysis of split and merge system with strictly circulate policies.

4. Structural properties

Using the above methods, we now investigate the structural properties of split and merge systems.

4.1. *Conservation of flow*

Corollary 1. *Under assumptions* 1 *to* 6, *the throughputs of Lines* 1 *to* 3 *in split and merge systems satisfy the following property:*

$$\widehat{TP}_{\mathrm{s},k,1} = \widehat{TP}_{\mathrm{s},k,2} + \widehat{TP}_{\mathrm{s},k,3}, \qquad k = \mathrm{p, c, sc, \%}, \quad (33)$$

$$\widehat{TP}_{\mathrm{m,k,3}} = \widehat{TP}_{\mathrm{m,k,1}} + \widehat{TP}_{\mathrm{m,k,2}}, \qquad k = \mathrm{p, c, sc}. \quad (34)$$

In particular, for the percentage split policy:

$$\widehat{TP}_{\mathrm{s},\%,2} = \alpha \widehat{TP}_{\mathrm{s},\%,1}, \qquad \widehat{TP}_{\mathrm{s},\%,3} = (1-\alpha)\widehat{TP}_{s,\%,1}, \quad (35)$$

and for the strictly circulate policy:

$$\widehat{TP}_{\mathrm{s,sc,2}} = \widehat{TP}_{\mathrm{s,sc,3}} = 0.5\widehat{TP}_{\mathrm{s,sc,1}},\ \widehat{TP}_{\mathrm{m,sc,1}} = \widehat{TP}_{\mathrm{m,sc,2}} = 0.5\widehat{TP}_{\mathrm{m,sc,3}}. \quad (36)$$

Proof. See Appendix B. ■

4.2. *Monotonicity*

It has been shown that monotonicity holds in serial lines and assembly systems (Li and Meerkov, 2009); i.e., improving machine reliability and/or increasing buffer capacity can lead to the improvement of the system production rate. Similar properties are observed in split and merge systems for all policies.

Corollary 2. *Under assumptions* 1 *to* 6, *the system throughputs in split and merge systems are monotonically increasing with respect to* c_i, μ_i, $i = 1, \ldots, 4$, *and* N_i, $i = 1, 2, 3$, *and decreasing with respect to* λ_i, $i = 1, \ldots, 4$.

Proof. See Appendix B. ■

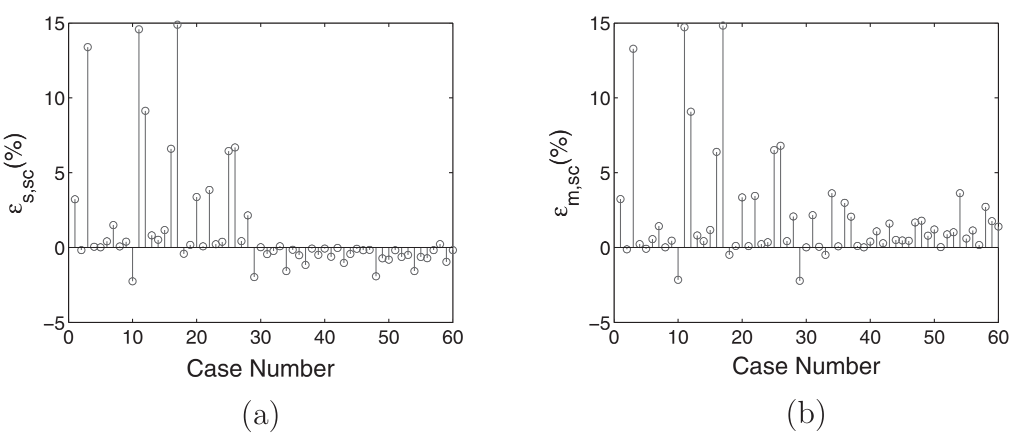


Fig. 6. Accuracy of Procedures 5 ($\alpha = 0.5$) and 6 for (a) the split system and (b) the merge system.

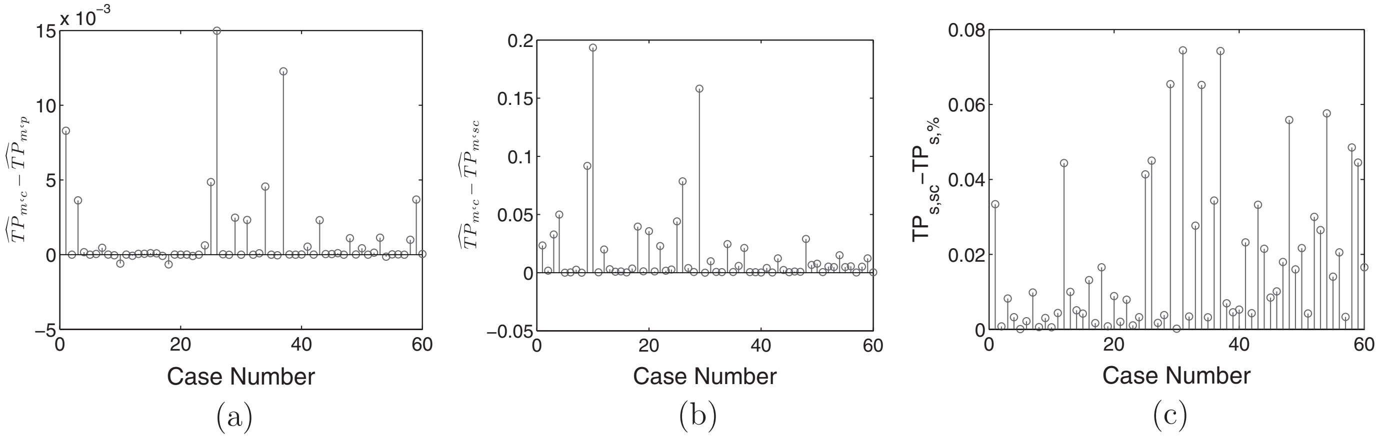


Fig. 7. Policy comparisons: (a) $\widehat{TP}_{\rm m,c} - \widehat{TP}_{\rm m,p}$, (b) $\widehat{TP}_{\rm m,c} - \widehat{TP}_{\rm m,sc}$, and (c) $\widehat{TP}_{\rm s,sc} - \widehat{TP}_{\rm s,\%}$

4.3. *Reversibility*

It is also known that the reversibility property exists in serial production lines (Li and Meerkov, 2009). For the split and merge systems with circulate, strictly circulate and priority policies considered in this article, this property still holds. To illustrate this, denote the machines and buffers in Fig. 1(a) as $m_i^{\rm s}$ and $b_i^{\rm s}$, and in Fig. 1 (b) as $m_i^{\rm m}$ and $b_i^{\rm m}$. We have the following equivalence relations:

$$\begin{aligned} &m_1^{\rm s} \Leftrightarrow m_4^{\rm m}, m_2^{\rm s} \Leftrightarrow m_3^{\rm m}, m_3^{\rm s} \Leftrightarrow m_1^{\rm m}, m_4^{\rm s} \Leftrightarrow m_2^{\rm m}, \\ &b_1^{\rm s} \Leftrightarrow b_3^{\rm m}, b_2^{\rm s} \Leftrightarrow b_1^{\rm m}, b_3^{\rm s} \Leftrightarrow b_2^{\rm m}. \end{aligned} \tag{37}$$

Corollary 3. *Under assumptions* 1 *to* 6 *and condition* (37)*, the system throughputs in split and merge systems with priority, circulate, and strictly circulate policies are identical. In other words:*

$$\widehat{TP}_{\rm m,p} = \widehat{TP}_{\rm s,p}, \quad \widehat{TP}_{\rm m,c} = \widehat{TP}_{\rm s,c}, \quad \widehat{TP}_{\rm m,sc} = \widehat{TP}_{\rm s,sc}. \tag{38}$$

Proof. See Appendix B. ■

4.4. *Policy comparison*

First, we compare the impacts on system throughput between the circulate and priority policies. The results, illustrated in Fig. 7(a), show that the differences in system throughputs between merge systems with circulate and priority policies are typically small. Due to reversibility, the

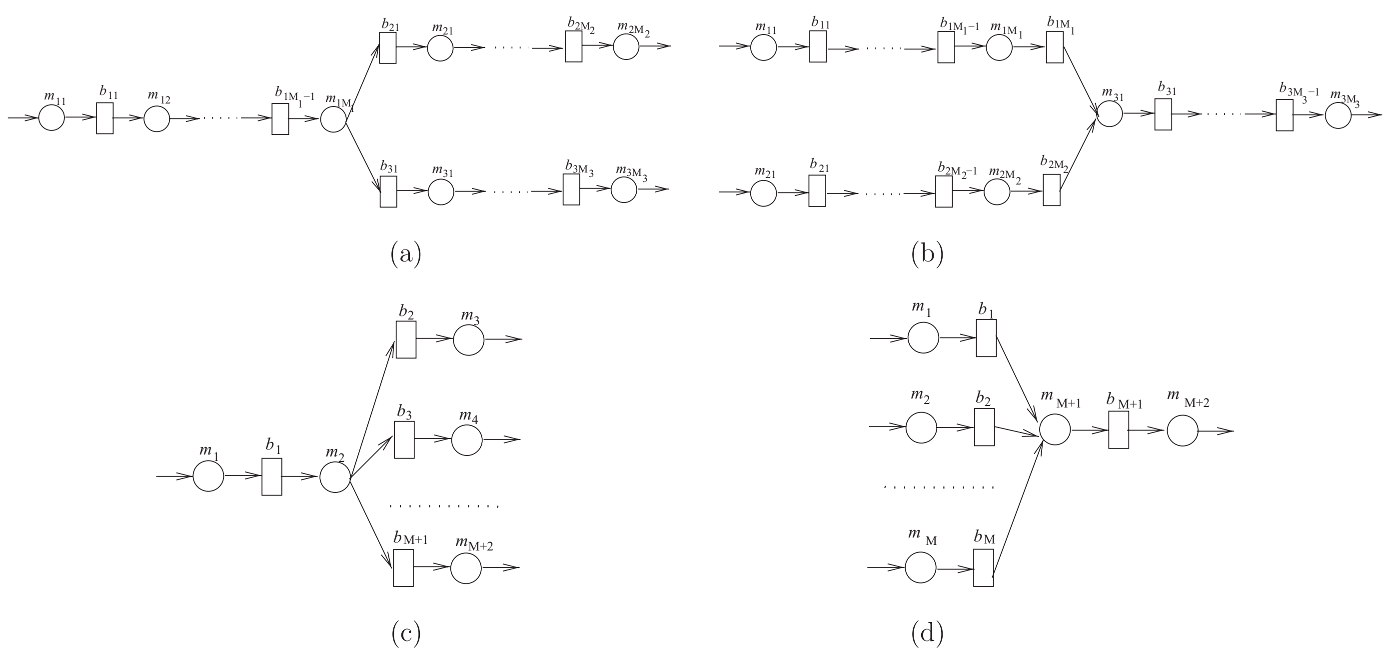


Fig. 8. Long and multiple split and merge lines: (a) long split system, (b) long merge system, (c) multiple-branch split system, and (d) multiple-branch merge system.

same differences will also be obtained for split systems. When the priority policy is implemented, the numerical results suggest that it is always beneficial to assign a higher priority to the branch with the higher throughput machine and larger level of buffering.

Next, the strictly circulate policy is compared with the circulate policy. It is shown in Fig. 7(b) that $\widehat{TP}_{\mathrm{m,c}}$ is always larger than $\widehat{TP}_{\mathrm{m,sc}}$. This is due to the fact that the strictly circulate policy requires that the machine needs to wait if the upstream buffer is not available to recieve parts.

Finally, a 50% split policy is compared with the strictly circulate policy in split systems. It is observed that the differences between $TP_{\mathrm{s,sc}}$ and $TP_{\mathrm{s,\%}}$ are always positive (Fig. 7(c)), but such differences are typically small.

5. Extensions

The proposed methods can be easily extended to split and merge systems with longer lines and multiple branches (Fig. 8). Again, the overlapping decomposition method can be applied to develop recursive procedures to evaluate system performance. For example, for long split lines, overlapped Lines 1 to 3 become $(m_{11}, \ldots, m'_{1M_1}, b_{11}, \ldots, b_{1M_1-1})$, $(m''_{1M_1}, m_{21}, \ldots, m_{2M_2}, b_{21}, \ldots, b_{2M_2})$ and $(m'''_{1M_1}, m_{31}, \ldots, m_{3M_3}, b_{31}, \ldots, b_{3M_3})$, respectively. In this case, the operator $TP(\cdot)$ needs to be replaced by the analysis of long asynchronous serial lines (Li and Meerkov, 2009). For multiple-branch split lines, $M+1$ lines are introduced, $(m_1, m'_2, b_1), (m''_2, m_3, b_2), \ldots, (m_2^{\prime\cdots\prime}, m_{M+2}, b_{M+1})$. Note that in circulate and strictly circulate policies, the coefficient of 0.5 will be changed to $1/M$. Also, the method has been applied to systems with multiple-branch split/merge and longer lines simultaneously.

Numerical results suggest that the proposed methods still achieve an acceptable accuracy in throughput estimation. In all the examples we tested (using the parameters selected from Equation (9)), the errors are typically within 5%. In addition, all structural properties hold for such systems as well.

6. Conclusions

Production split and merge operations are widely used in many manufacturing systems. This article has presented analytical methods to approximate the throughputs of split and merge systems with exponential machine reliability models. Priority, circulate, strictly circulate, and percentage policies have been discussed. Monotonicity, reversibility, and other system properties have been investigated. The impacts on system throughput of using the different policies have been compared. The proposed approach provides production engineers and managers with a quantitative tool for design and continuous improvement of complex production systems.

References

Biller, S., Li, J., Meerkov, S.M. and Zhang, L. (2010) Bottlenecks in Bernoulli serial lines with rework. *IEEE Transactions on Automation Science and Engineering*. DOI 10.1109/TASE.2009.2023463

Chiang, S.-Y., Kuo, C.-T. and Meerkov, S.M. (2001) c-Bottlenecks in serial production lines: identification and application. *Mathematical Problems in Engineering*, **7**, 543–578.

Colledani, M., Gandola, F., Matta, A. and Tolio, T. (2008) Performance evaluation of linear and non-linear multi-product multi-stage lines with unreliable machines and finite homogeneous buffers. *IIE Transactions*, **40**, 612–626.

Colledani, M., Matta, A. and Tolio, T. (2005) Performance evaluation of production lines with finite buffer capacity producing two different products. *OR Spectrum*, **27**, 243–263.

Diamantidis, A.C. and Papadopoulos, C.T. (2006) Markovian analysis of a discrete material manufacturing system with merge operations, operation-dependent and idleness failures. *Computer & Industrial Engineering*, **50**, 466–487.

Diamantidis, A.C., Papadopoulos, C.T. and Vidalis, M. (2004) Exact analysis of a discrete material three-station one-buffer merge system with unreliable machines. *International Journal of Production Research*, **42**, 651–675.

Haige, J.W. and Paige, K.N. (2001) *Learning Simul8: the Complete Guide*. Plain Vu Publishing, Bellingham, WA.

Helber, S. (2000) Approximate analysis of unreliable transfer lines with splits in the flow of material. *Annals of Operations Research*, **93**, 217–243.

Helber, S. and Mehrtens, N. (2003) Exact analysis of continuous material merge system with limited buffer capacity and three stations, in *Analysis and Modeling of Manufacturing Systems*, Gershwin, S., Dallery, Y., Papadopoulos, C.T., and Smith, J.M. (eds), Kluwer Academic Publisher, Norwell, MA, pp. 85–121.

Li, J. (2004a) Modeling and analysis of manufacturing systems with parallel lines. *IEEE Transactions on Automatic Control*, **49**, 1824–1829.

Li, J. (2004b) Performance analysis of production systems with rework loops. *IIE Transactions*, **36**, 755–765.

Li, J. (2004c) Throughput analysis in automotive paint shops: a case study. *IEEE Transactions on Automation Sciences and Engineering*, **1**, 90–98.

Li, J. (2005) Overlapping decomposition: a system-theoretic method for modeling and analysis of complex production systems. *IEEE Transactions on Automation Sciences and Engineering*, **2**, 40–53.

Li, J., Blumenfeld, D.E., Huang, N. and Alden, J.M. (2009) Throughput analysis in production systems: recent advances and future topics. *International Journal of Production Research*, **47**, 3823–3851.

Li, J. and Huang, N. (2005) Modeling and analysis of a multiple product manufacturing system with split and merge. *International Journal of Production Research*, **43**, 4049–4066.

Li, J. and Meerkov, S.M. (2009) *Production Systems Engineering*, Springer, New York, NY.

Liu, Y. and Li, J. (2009) Modeling and analysis of split and merge systems with bernoulli reliability machines. *International Journal of Production Research*, **47**, 4373–4397.

Tan, B. (2001) A three-station merge system with unreliable stations and a shared buffer. *Mathematical and Computer Modeling*, **33**, 1011–1026.

Appendices

Appendix A: Operator $TP(\cdot)$

Introduce $TP(c_1, \lambda_1, \mu_1, c_2, \lambda_2, \mu_2, N_1)$ as an operator to calculate the throughput of a two-machine line, where c_i, λ_i, and μ_i, $i = 1, 2$, are the capacity, failure, and repair rates of machine i, respectively, and N_1 represents the buffer capacity. Then, the line throughput can be calculated as (Chiang *et al.*, 2001; Li and Meerkov, 2009):
when $c_1 < c_2$:

$$TP = \frac{c_2 e_2 A e^{k_1 N_1} + c_1 e_1 B e^{k_2 N_1} + c_1 e_1 C_1 e^{-k_2 N_1}}{A e^{k_1 N_1} + B e^{k_2 N_1} + C_1 e^{-k_2 N_1}}, \quad \text{(A1)}$$

where

$$\begin{aligned} e_i &= \frac{\mu_i}{\lambda_i + \mu_i}, i = 1, 2, \\ k_1 &= \big[\mu_1 c_1^2(\mu_1 + \mu_2 + \lambda_2) - c_1 c_2\big[(\mu_1 + \mu_2)^2 \\ &\quad + (\mu_1 + \mu_2)(\lambda_1 + \lambda_2) \\ &\quad + (\mu_1\lambda_2 + \mu_2\lambda_1)\big] + \mu_2 c_2^2(\mu_1 + \mu_2 + \lambda_1)\big]/ \\ &\quad [2c_1 c_2(\mu_1 + \mu_2)(c_1 - c_2)], \\ k_2 &= \frac{(c_1\mu_1 + c_2\mu_2)R}{2c_1 c_2(\mu_1 + \mu_2)(c_2 - c_1)}, \\ A &= \mu_1 R^2 + \mu_1 R[c_1(\mu_1 + \mu_2 + \lambda_2) \\ &\quad - c_2(\mu_1 + \mu_2 + \lambda_1)], \qquad \text{(A2)} \\ B &= \mu_2\lambda_1 c_2[(c_1 - c_2)(\mu_1 - \mu_2) - (c_2\lambda_1 + c_1\lambda_2) - R], \\ R &= \sqrt{[c_1(\mu_1 + \mu_2 + \lambda_2) - c_2(\mu_1 + \mu_2 + \lambda_1)]^2} \\ &\quad + 4c_1 c_2\lambda_1\lambda_2, \\ C_1 &= \frac{e_2(c_2 - c_1 e_1)A + c_1 e_1(1 - e_2)B}{c_1 e_1(e_2 - 1)}. \end{aligned}$$

when $c_1 = c_2$:

$$\begin{aligned} TP &= c_2 e_2[1 - Q(\lambda_1, \mu_1, \lambda_2, \mu_2, N_1)] \\ &= c_1 e_1[1 - Q(\lambda_2, \mu_2, \lambda_1, \mu_1, N_1)], \qquad \text{(A3)} \end{aligned}$$

where

$$Q(\lambda_1, \mu_1, \lambda_2, \mu_2, N_1) = \begin{cases} \dfrac{(1 - e_1)(1 - \phi)}{1 - \phi e^{-\beta N_1}}, & \text{if } \dfrac{\lambda_1}{\mu_1} \neq \dfrac{\lambda_2}{\mu_2}, \\ \dfrac{\lambda_1(\lambda_1 + \lambda_2)(\mu_1 + \mu_2)}{(\lambda_1 + \mu_1)[)\lambda_1 + \lambda_2)(\mu_1 + \mu_2) + \lambda_2\mu_1(\lambda_1 + \lambda_2 + \mu_1 + \mu_2)N_1]}, & \text{if } \dfrac{\lambda_1}{\mu_1} = \dfrac{\lambda_2}{\mu_2}, \end{cases} \quad \text{(A4)}$$

$$\phi = \frac{e_1(1 - e_2)}{e_2(1 - e_1)}, \quad \beta = \frac{(\lambda_1 + \lambda_2 + \mu_1 + \mu_2)(\lambda_1\mu_2 - \lambda_2\mu_1)}{(\lambda_1 + \lambda_2)(\mu_1 + \mu_2)}.$$

When $c_1 > c_2$: By reversibility.

Appendix B: Proofs

The proof of Theorem 1 requires the following lemmas:

Lemma A1. *Consider sequences $\widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s)$ defined in Procedure 1. If $\widehat{X}_{2N_2}(s) > \widehat{X}_{2N_2}(s-1)$ and $\widehat{X}_{3N_3}(s) > \widehat{X}_{3N_3}(s-1)$, then $\widehat{X}_{2N_2}(s+1) > \widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s+1) > \widehat{X}_{3N_3}(s)$.*

Proof. If

$$\widehat{X}_{2N_2}(s) > \widehat{X}_{2N_2}(s-1), \qquad \widehat{X}_{3N_3}(s) > \widehat{X}_{3N_3}(s-1),$$

from Equation (1), and monotonicity in serial line (Li and Meerkov, 2009) we obtain:

$$c_2'(s+1) < c_2'(s), \quad \widehat{TP}_{\text{s,p},1}(s+1) < \widehat{TP}_{\text{s,p},1}(s), \quad \widehat{X}_{10}(s+1) < \widehat{X}_{10}(s).$$

It follows from Equation (2) that:

$$c_2''(s+1) > c_2''(s), \quad \widehat{TP}_{\text{s,p},2}(s+1) > \widehat{TP}_{\text{s,p},2}(s), \quad \widehat{X}_{2N_2}(s+1) > \widehat{X}_{2N_2}(s).$$

Analogously, due to Equation (3), we obtain:

$$c_2'''(s+1) > c_2'''(s), \quad \widehat{TP}_{\text{s,p},3}(s+1) > \widehat{TP}_{\text{s,p},3}(s), \quad \widehat{X}_{3N_3}(s+1) > \widehat{X}_{3N_3}(s).$$

■

Lemma A2. *Sequences $\widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s)$ in Procedure 1 are monotonically increasing.*

Proof. By induction. For $s = 0$, and initial condition $\widehat{X}_{2N_2}(0) = \widehat{X}_{2N_2}(0) = 0$, it follows from Equation (1) that:

$$c_2'(1) = c_2, \quad \widehat{TP}_{\text{s,p},1}(1) > 0, \quad \widehat{X}_{10}(1) < 1.$$

By Equation (2) we have:

$$c_2''(1) = c_2(1 - \widehat{X}_{10}(1)), \quad \widehat{X}_{2N_2}(1) > 0.$$

It follows from Equation (3) that:

$$c_2'''(1) = c_2\widehat{X}_{2N_2}(1)(1 - \widehat{X}_{10}(1)), \quad \widehat{X}_{3N_3}(1) > 0.$$

Thus,

$$\widehat{X}_{2N_2}(1) > \widehat{X}_{2N_2}(0), \quad \widehat{X}_{3N_3}(1) > \widehat{X}_{3N_3}(0).$$

The base case is proved.

Now assume that for $s > 0$, $\widehat{X}_{2N_2}(s) > \widehat{X}_{2N_2}(s-1)$. From Lemma A1, we obtain $\widehat{X}_{2N_2}(s+1) > \widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s+1) > \widehat{X}_{3N_3}(s)$. Therefore, the sequences $\widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s)$ are monotonically increasing. ■

Proof of Theorem 1. Since the sequences $\widehat{X}_{2N_2}(s)$ and $\widehat{X}_{3N_3}(s)$ defined in Procedure 1 are monotonically increasing (Lemma A2) and are bounded between zero and one (Li and Meerkov, 2009), they are convergent. Therefore, the limits of $\widehat{TP}_{\text{s,p},i}(s)$, $i = 1, 2, 3$, exist. This proves the first part of the theorem.

For the second part, the steady-state equations of Procedure 1 are

$$\begin{aligned}
c_2' &= c_2(1 - \widehat{X}_{2N_2}\widehat{X}_{3N_3}),\\
\widehat{TP}_{\mathrm{s,p},1} &= TP(c_1, \lambda_1, \mu_1, c_2', \lambda_2, \mu_2, N_1), \qquad \text{(A5)}\\
\widehat{X}_{10} &= 1 - \frac{\widehat{TP}_{\mathrm{s,p},1}}{c_2' e_2},\\
c_2'' &= c_2(1 - \widehat{X}_{10}),\\
\widehat{TP}_{\mathrm{s,p},2} &= TP(c_2'', \lambda_2, \mu_2, c_3, \lambda_3, \mu_3, N_2), \qquad \text{(A6)}\\
\widehat{X}_{2N_2} &= 1 - \frac{\widehat{TP}_{\mathrm{s,p},2}}{c_2'' e_2},\\
c_2''' &= c_2 \widehat{X}_{2N_2}(1 - \widehat{X}_{10}),\\
\widehat{TP}_{\mathrm{s,p},3} &= TP([c_2''', \lambda_2, \mu_2, c_4, \lambda_4, \mu_4, N_3), \qquad \text{(A7)}\\
\widehat{X}_{3N_3} &= 1 - \frac{\widehat{TP}_{\mathrm{s,p},3}}{c_2''' e_2}.
\end{aligned}$$

Assume there exists another solution, $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} \neq \widehat{TP}_{\mathrm{s,p},1}$. If $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} > \widehat{TP}_{\mathrm{s,p},1}$, then from Equation (A5) we must have $\tilde{c}_2' > c_2'$ and $\widetilde{\widehat{X}}_{10} > \widehat{X}_{10}$. It follows from Equation (A6) that $\tilde{c}_2'' > c_2''$, $\widetilde{\widehat{TP}}_{\mathrm{s,p},2} < \widehat{TP}_{\mathrm{s,p},2}$, and $\widetilde{\widehat{X}}_{2N} < \widehat{X}_{2N}$. In addition, from Equation (A7) we obtain $\tilde{c}_2''' < c_2'''$ and $\widetilde{\widehat{TP}}_{\mathrm{s,p},3} < \widehat{TP}_{\mathrm{s,p},3}$. This implies that $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} < \widehat{TP}_{\mathrm{s,p},1}$, which is a contradiction. Similarly, we can obtain another contradiction if $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} < \widehat{TP}_{\mathrm{s,p},1}$. Thus, the only possibility is $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} = \widehat{TP}_{\mathrm{s,p},1}$; i.e., there is a unique solution.

The proofs for the merge system can be obtained similarly. ■

Proof of Theorem 2. Similar to the proof of Theorem 1. ■

Proof of Corollary 1. Using steady-state Equations (A5) to (A7), we obtain:

$$\begin{aligned}
\widehat{TP}_{\mathrm{s,p},1} &= c_2(1 - \widehat{X}_{2N_2}\widehat{X}_{3N_3})e_2(1 - \widehat{X}_{10}),\\
\widehat{TP}_{\mathrm{s,p},2} &= c_2(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{2N_2}),\\
\widehat{TP}_{\mathrm{s,p},3} &= c_2\widehat{X}_{2N_2}(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{3N_3}).
\end{aligned}$$

It follows that:

$$\begin{aligned}
\widehat{TP}_{\mathrm{s,p},2} + \widehat{TP}_{\mathrm{s,p},3} &= c_2(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{2N_2})\\
&\quad + c_2\widehat{X}_{2N_2}(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{3N_3})\\
&= c_2(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{2N_2} + \widehat{X}_{2N_2} - \widehat{X}_{2N_2}\widehat{X}_{3N_3})\\
&= c_2(1 - \widehat{X}_{10})e_2(1 - \widehat{X}_{2N_2}\widehat{X}_{3N_3}) = \widehat{TP}_{\mathrm{s,p},1}.
\end{aligned}$$

Similar proofs can be obtained for circulate, percentage, and strictly circulate policies. ■

Proof of Corollary 2. By contradiction. First, consider the split system with priority policy. Assume that if $\tilde{\lambda}_1 > \lambda_1$, we have $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} \geq \widehat{TP}_{\mathrm{s,p},1}$. Then, from Equation (A5) we have:

$$\tilde{c}_2' > c_2', \quad \widetilde{\widehat{X}}_{2N_2}\widetilde{\widehat{X}}_{3N_3} < \widehat{X}_{2N_2}\widehat{X}_{3N_3}, \quad \widetilde{\widehat{X}}_{10} > \widehat{X}_{10}.$$

By Equation (A6),

$$\tilde{c}_2'' < c_2'', \quad \widetilde{\widehat{TP}}_{\mathrm{s,p},2} < \widehat{TP}_{\mathrm{s,p},2}, \quad \widetilde{\widehat{X}}_{2N_2} < \widehat{X}_{2N_2}.$$

Then, Equation (A7) results in

$$\tilde{c}_2''' < c_2''', \quad \widetilde{\widehat{TP}}_{\mathrm{s,p},3} < \widehat{TP}_{\mathrm{s,p},3}, \quad \widetilde{\widehat{X}}_{3N_3} < \widehat{X}_{3N_3}.$$

This leads to $\widetilde{\widehat{TP}}_{\mathrm{s,p},2} + \widetilde{\widehat{TP}}_{\mathrm{s,p},3} < \widehat{TP}_{\mathrm{s,p},2} + \widehat{TP}_{\mathrm{s,p},3}$; i.e., $\widetilde{\widehat{TP}}_{\mathrm{s,p},1} < \widehat{TP}_{\mathrm{s,p},1}$, which is a contradiction. Thus, the throughput is monotonically decreasing with respect to λ_1.

The monotonicity of $TP_{\mathrm{s,p},1}$ with respect to other λ_i, μ_i, c_i, and N_i can be proved analogously. Moreover, similar logics can be applied to a merge system and other policies. ■

Proof of Corollary 3. Consider the split and merge systems with priority policy. In Procedures 1 and 2, $\widehat{TP}_{\mathrm{s,p}} = \widehat{TP}_{\mathrm{s,p},1}$, $\widehat{TP}_{\mathrm{m,p}} = \widehat{TP}_{\mathrm{m,p},3}$. From condition (37), rewrite Procedure 2. By renaming the variables as

$$\begin{aligned}
&c_3'^{\mathrm{m}} \to c_2'^{\mathrm{s}} \quad c_3''^{\mathrm{m}} \to c_2''^{\mathrm{s}}, \quad c_3'''^{\mathrm{m}} \to c_2'''^{\mathrm{s}},\\
&\widehat{X}_{10}^{m} \to \widehat{X}_{20}^{s}, \quad \widehat{X}_{20}^{\mathrm{m}} \to \widehat{X}_{30}^{s}, \quad \widehat{X}_{3N_3}^{\mathrm{m}} \to \widehat{X}_{1N_1}^{s},
\end{aligned}$$

we obtain the same procedure as Procedure 1. Therefore, we can conclude that $\widehat{TP}_{\mathrm{s,p}} = \widehat{TP}_{\mathrm{m,p}}$.

Similarly, we can prove other equations. ■

Biographies

Yang Liu received his B.S. and M.S. from the Department of Automation, Tsinghua University, Beijing, China, in 2004 and 2006, respectively. He is now a Ph.D. candidate in the Department of Electrical and Computer Engineering and Center for Manufacturing at the University of Kentucky. He received the Commonwealth Research Award in 2008. His main research interests are in performance analysis of complex manufacturing systems.

Jingshan Li is an Assistant Professor in Department of Electrical and Computer Engineering and Center for Manufacturing, University of Kentucky, Lexington, KY. Before joining the University of Kentucky in 2006, he was a Staff Research Engineer in Manufacturing Systems Research Lab, General Motors Research & Development Center, Warren, MI. He received a bachelor's degree from the Department of Automation, Tsinghua University, Beijing, China, and a master's degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, and a Ph.D. in Electrical Engineering Systems, University of Michigan, Ann Arbor, MI in 1989, 1992 and 2000, respectively. He has published over 30 refereed journal articles in *IIE Transactions*, *IEEE Transactions*, *Annals of Operations Research*, *International Journal of Production Research*, etc. He is also the Associate Editor of *IEEE Transactions on Automation Science and Engineering* and *Mathematical Problems in Engineering*. He received the 2009 IIE Transactions Best Application Paper Award—Design and Manufacturing, 2005 IEEE Transactions on Automation Science and Engineering Best Paper Award, 2006 IEEE Early Industry/Government Career Award in Robotics and Automation, and he was also a finalist of the Best Paper Award of 2009 IEEE Conference on Automation Science and Engineering and the Best Automation Paper Award of 2005 IEEE International Conference on Robotics and Automation. His primary research interests are in modeling, analysis, and control of manufacturing and service systems. He is a senior member of IEEE and IIE.