2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Performance Analysis of Split and Merge Production Systems

Yang Liu and Jingshan Li

Abstract— **Production split and merge are widely used in many manufacturing systems to increase production capacity and variety, improve product quality, and carry out scheduling and control activities. In this paper, we present analytical methods to analyze such systems with exponential machine reliability models, operating under circulate, strictly circulate, priority and percentage split/merge policies.**

I. INTRODUCTION

Performance analysis of production systems has received tremendous amount of research focus in last five decades (see reviews [1], [2] and monographs [3]-[5]). In modern manufacturing systems, to improve productivity, quality and flexibility, split and merge structures are often used to increase production capacity, improve product quality, and carry out scheduling and control policies. In addition, routing policies at the split or merge stations play an important role in such systems since they directly control the parts flow. Therefore, to design and manage such systems more efficiently, modeling and analysis of split and merge systems with different production control policies are of significant importance.

In practice, two types of split or merge are encountered. One is known as *assembly merge* (or *disassembly split*). In this case, the assembly machine needs to take parts from all its upstream buffers and assemble them into a single product (respectively, disassemble a single part into many ones to all downstream buffers). Another type is referred to as *production merge* (or *production split*), where the merge station will only take one part from one of its upstream buffers each time (or send one part to one of its downstream buffers). In this paper, the latter case is considered. Specifically, we consider split and merge systems with exponential reliability machines. Four frequently used split and merge policies are addressed: *circulate, strictly circulate, priority* and *percentage*. In circulate policy, the split machine sends the part to downstream branches in circulation when it is not blocked by any branch. However, if a branch blocks the split machine, it will be ignored and part will flow to subsequent branch. Similar scenario occurs in merge operations, where the merge station takes part from all upstream branches circularly if it is not starved, and the empty buffer branch will be ignored. In strictly circulate policy, routing is similar to circulate policy except that blocked or starved branches cannot be ignored, parts need to wait until the branch is available. In priority policy, parts will be dispatched to the branch with higher priority unless it is blocked. Only when the split machine is blocked by the higher priority branch, it will send parts to the lower priority one. Similarly, the merge machine always takes parts from higher priority upstream branch if it is available. In percentage policy, parts are dispatched to downstream branches or loaded from upstream one following a given percentage.

In recent years, several performance analysis methods have been developed to study split and merge systems. For example, paper [6] introduces a geometric machine reliability model for a transfer line with split operations based on percentage routing policy. Merge systems with a shared buffer are discussed in [7] and [8]. Priority merge policy is assumed in these papers when the shared buffer is transit to full state. Multiple product systems have been studied in [9], where different products are processed separately at the dedicated machines or buffers. Another direction of study focuses on rework loops ([10], [11]). In such systems, repaired parts are typically assumed to have higher priority at merge station to avoid deadlock. Parallel systems are investigated in [12] where parallel lanes are split from a common buffer and then merge into another shared buffer. Paper [13] presents a general method to model complex production systems, referred to as *overlapping decomposition*.

In spite of these, the production split and merge systems with different policies have not been studied thoroughly. In particular, the systems under different split and merge policies need to be analyzed in details. Such a study can enable us to understand the impact of different policies and provide guidance to control the part flow in operations. A preliminary study on Bernoulli split and merge systems with several routing policies has been carried out in [14]. This paper is intended to extend this work to more general cases, exponential machine reliability models with more routing policies. The main contribution of this paper is in development of analytical models to study split and merge systems with different routing policies. Recursive procedures are proposed to analyze system performance. The convergence of the procedures and uniqueness of the solutions are justified, and accuracy are validated with acceptable precision.

The remaining of the paper is structured as follows: the problem to be addressed is formulated in Section II. The modeling and analysis methods for split and merge systems with different routing policies are introduced in Section III. Conclusions are given in Section IV. Due to space limitation, all proofs are omitted and can be found in [15].

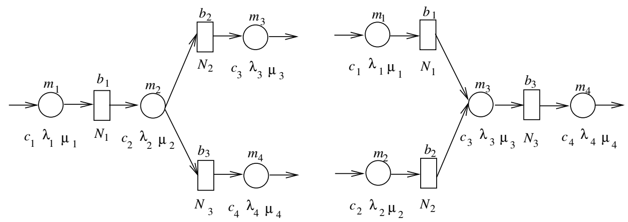
The work is partially supported by University of Kentucky Faculty Research Grant.

Y. Liu and J. Li are with Dept. of Electrical and Computer Engineering and Center for Manufacturing, University of Kentucky, Lexington, KY 40506, USA. yang.liu@uky,edu, jingshan@engr.uky.edu

Please send all correspondences to Prof. Jingshan Li.

II. SYSTEM DESCRIPTION AND PROBLEM FORMULATION

In this paper, we consider a typical four-machine split (or merge) system, whose layout is shown in Figure 1(a) (respectively, Figure 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.



(a) Split system (b) Merge system

Fig. 1. Split and merge systems

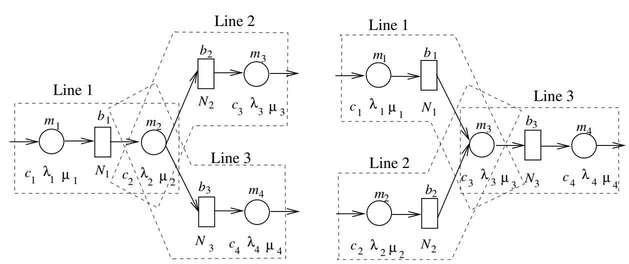
1) Each machine m_i, $i = 1, \ldots, 4$, has two states: up and down. When it is up, it is capable of processing parts with capacity c_i parts/unit of time. When the machine is down, no production takes place.
2) The up- and downtimes of machine m_i are random variables exponentially distributed with parameters λ_i and μ_i, respectively. In other words, 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 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 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 split system (or m_3 in merge system) will send part to downstream buffers b_2 and b_3 (respectively, take material from upstream buffers b_1 and b_2) based on the following policies:
   - *Priority policy.* Buffer b_2 has higher priority, i.e., m_2 will keep sending parts to b_2 whenever it has space (respectively, buffer b_1 has higher priority, and m_3 takes part 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).
   - *Circulate policy.* Machine m_2 will send part to buffers b_2 and b_3 circularly if it is not blocked by both buffers (respectively, m_3 takes part 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 part from another buffer if it is starved by one).
   - *Strictly circulate policy.* Machine m_2 will send part to buffers b_2 and b_3 (respectively, m_3 takes part from b_1 and b_2) circularly. If it is blocked by one buffer, m_2 will wait until the buffer is available (respectively, m_3 will wait until the buffer has available part if it is starved).
   - *Percentage policy (split only).* Machine m_2 will send a part to buffers b_1 and b_2 based on pre-designed percentage, $\alpha \cdot 100\%$ to b_1 and $(1-\alpha) \cdot 100\%$ to b_2.

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 merge case) per unit of time. The problem addressed in this paper is as follows: *Given production system 1)-6), develop a method to evaluate the system throughput as a function of the system parameters.*

III. PERFORMANCE ANALYSIS

A. Overlapping Decomposition

Since the split (or merge) machine has to allocate its capacity to different downstream (respectively, upstream) branches and all machines and buffers in the system interfere with each other and impact such allocation, the exact analysis seems impossible. Therefore, approximation method is pursued here. The idea of the approximation is based on overlapping decomposition ([13]), and is illustrated below (Figure 2(a) for split system and (b) for merge system):



(a) Split system (b) Merge system

Fig. 2. Overlapping decomposition of split and merge systems

Consider the split system illustrated in Figure 2(a), decompose the system into three overlapped serial lines, where m_2 is the overlapping machine. Specifically, modify machine m_2 as m_2' to take into account the effects that m_2 is blocked by b_2 and b_3, 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 using a two-machine throughput analysis formula ([5]). Using this probability, consider machine m_2 with capacity allocated to buffer b_2 and m_3 only, modify m_2 to include this starvation and capacity allocation, we obtain m_2'' and the second overlapped serial 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 the capacity allocated only to b_3 and m_4, we modify m_2 to m_2''' and obtain Line 3 (m_2''', b_3 and m_4). The probability that m_2 is blocked by b_3 can be computed. Next, using these blockage probabilities, we carry out the analysis for Line 1 again, and the procedure is repeated anew. When the procedure is convergent, the production rates of Lines 1-3 are obtained.

For the merge system in Figure 2(b), a similar idea can be applied but with m_3 as the overlapping machine. Here Line 1 consists of m_1, b_1 and a modified machine m_3', which takes into account the effect of blockage of buffer b_3 and capacity allocation to branch b_1 and m_1 only. Line 2 is composed of m_2, b_2 and pseudo machine m_3'', which considers the blockage of b_3 and capacity allocated to b_2 and m_2. Finally, 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). The recursive procedure is again introduced to update the blockage and starvation probabilities of machine m_3 until it is convergent.

B. Operator TP

To implement the approximation procedure introduced above, an operator TP to calculate the throughput of two-machine line is needed. We denote such an operator as $TP(c_1,\lambda_1,\mu_1,c_2,\lambda_2,\mu_2,N_1)$. Then the line throughput can be calculated as (see [5] for details).

- $c_1 < c_2$

$$TP = \frac{c_2e_2Ae^{k_1N_1}+c_1e_1Be^{k_2N_1}+c_1e_1Ce^{-k_2N_1}}{Ae^{k_1N_1}+Be^{k_2N_1}+C_1e^{-k_2N_1}}, \quad (1)$$

where

$$\begin{aligned}
e_i &= \frac{\mu_i}{\lambda_i+\mu_i}, i=1,2\\
k_1 &= \frac{1}{2c_1c_2(\mu_1+\mu_2)(c_1-c_2)}[\mu_1c_1^2(\mu_1+\mu_2+\lambda_2)\\
&\quad -c_1c_2[(\mu_1+\mu_2)^2+(\mu_1\lambda_2+\mu_2\lambda_1)\\
&\quad +(\mu_1+\mu_2)(\lambda_1+\lambda_2)]+\mu_2c_2^2(\mu_1+\mu_2+\lambda_1)],\\
k_2 &= \frac{(c_1\mu_1+c_2\mu_2)R}{2c_1c_2(\mu_1+\mu_2)(c_2-c_1)},\\
A &= \mu_1R^2+\mu_1R[c_1(\mu_1+\mu_2+\lambda_2) \qquad (2)\\
&\quad -c_2(\mu_1+\mu_2+\lambda_1)],\\
B &= \mu_2\lambda_1c_2[(c_1-c_2)(\mu_1-\mu_2)-(c_2\lambda_1+c_1\lambda_2)-R],\\
R &= \{[c_1(\mu_1+\mu_2+\lambda_2)-c_2(\mu_1+\mu_2+\lambda_1)]^2\\
&\quad +4c_1c_2\lambda_1\lambda_2\}^{1/2},\\
C_1 &= \frac{e_2(c_2-c_1e_1)A+c_1e_1(1-e_2)B}{c_1e_1(e_2-1)}.
\end{aligned}$$

- $c_1 = c_2$

$$\begin{aligned}
TP &= c_2e_2[1-Q(\lambda_1,\mu_1,\lambda_2,\mu_2,N_1)]\\
&= c_1e_1[1-Q(\lambda_2,\mu_2,\lambda_1,\mu_1,N_1)], \qquad (3)
\end{aligned}$$

where

$$Q(\lambda_1,\mu_1,\lambda_2,\mu_2,N_1) = \begin{cases} \frac{(1-e_1)(1-\phi)}{1-\phi e^{-\beta N_1}}, & \text{if } \frac{\lambda_1}{\mu_1} \neq \frac{\lambda_2}{\mu_2},\\ \frac{\lambda_1(\lambda_1+\lambda_2)(\mu_1+\mu_2)}{(\lambda_1+\mu_1)[)\lambda_1+\lambda_2)(\mu_1+\mu_2)+\lambda_2\mu_1(\lambda_1+\lambda_2+\mu_1+\mu_2)N_1]}, & \text{if } \frac{\lambda_1}{\mu_1} = \frac{\lambda_2}{\mu_2}, \end{cases} \quad (4)$$

$$\begin{aligned}
e_i &= \frac{\mu_i}{\lambda_i+\mu_i}, \qquad i=1,2,\\
\phi &= \frac{e_1(1-e_2)}{e_2(1-e_1)}, \qquad (5)\\
\beta &= \frac{(\lambda_1+\lambda_2+\mu_1+\mu_2)(\lambda_1\mu_2-\lambda_2\mu_1)}{(\lambda_1+\lambda_2)(\mu_1+\mu_2)}.
\end{aligned}$$

- $c_1 > c_2$. By reversibility.

Using this operator, recursive procedures are developed to analyze split and merge systems with different routing policies. The specific policy will be taken into account when modifications of m_2 in split system or m_3 in merge system are carried out. Below, details of these modifications are introduced.

C. Priority Policy

1) Recursive procedures:

a) Split system: Consider the split system in Figure 2(a). The modification of m_2 for priority policy is carried out as follows: Assume buffer b_2 has higher priority than b_3. Then, in Line 2, m_2 is always available to b_2 when it is not starved, thus c_2 is multiplied by the probability that b_1 is not empty. However, m_2 is available to b_3 only when it is not starved by b_1, but blocked by b_2. Therefore, c_2 is modified by these two probabilities. Then the recursive procedure is introduced as:

Procedure 1:

$$\begin{aligned}
&\text{Line 1}\\
c_2'(s+1) &= c_2(1-\widehat{X}_{2N_2}(s)\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{s,p,1}(s+1) &= TP(c_1,\lambda_1,\mu_1,c_2'(s+1),\lambda_2,\mu_2,N_1), \qquad (6)\\
\widehat{X}_{10}(s+1) &= 1-\frac{\widehat{TP}_{s,p,1}(s+1)}{c_2'(s+1)e_2},\\
&\text{Line 2}\\
c_2''(s+1) &= c_2(1-\widehat{X}_{10}(s+1)),\\
\widehat{TP}_{s,p,2}(s+1) &= TP(c_2''(s+1),\lambda_2,\mu_2,c_3,\lambda_3,\mu_3,N_2), \qquad (7)\\
\widehat{X}_{2N_2}(s+1) &= 1-\frac{\widehat{TP}_{s,p,2}(s+1)}{c_2''(s+1)e_2},\\
&\text{Line 3}\\
c_2'''(s+1) &= c_2\widehat{X}_{2N_2}(s+1)(1-\widehat{X}_{10}(s+1)),\\
\widehat{TP}_{s,p,3}(s+1) &= TP(c_2'''(s+1),\lambda_2,\mu_2,c_4,\lambda_4,\mu_4,N_3), \qquad (8)\\
\widehat{X}_{3N_3}(s+1) &= 1-\frac{\widehat{TP}_{s,p,3}(s+1)}{c_2'''(s+1)e_2},\\
s &= 0,1,2,\ldots,\\
\widehat{X}_{2N_2}(0) &= \widehat{X}_{3N_3}(0)=0,
\end{aligned}$$

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, and $\widehat{TP}_{s,p,i}(s)$ is the throughput of line i in split system with priority policy at the s-th iteration. Similar notations are used in subsequent procedures.

b) Merge system: Assuming buffer b_1 has higher priority than b_2. Analogously to Procedure 1, we have

Procedure 2:

$$\begin{aligned}
&\text{Line 1}\\
c_3'(s+1) &= c_3(1-\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{m,p,1}(s+1) &= TP(c_1,\lambda_1,\mu_1,c_3'(s+1),\lambda_3,\mu_3,N_1), \qquad (9)\\
\widehat{X}_{10}(s+1) &= 1-\frac{\widehat{TP}_{m,p,1}(s+1)}{c_3'(s+1)e_3},\\
&\text{Line 2}\\
c_3''(s+1) &= c_3\widehat{X}_{10}(s+1)(1-\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{m,p,2}(s+1) &= TP(c_2,\lambda_2,\mu_2,c_3''(s+1),\lambda_3,\mu_3,N_2), \qquad (10)\\
\widehat{X}_{20}(s+1) &= 1-\frac{\widehat{TP}_{m,p,2}(s+1)}{c_3''(s+1)e_3},\\
&\text{Line 3}
\end{aligned}$$

$$
\begin{aligned}
c_3'''(s+1) &= c_3(1-\widehat{X}_{10}(s+1)\widehat{X}_{20}(s+1)),\\
\widehat{TP}_{m,p,3}(s+1) &= TP(c_3'''(s+1),\lambda_3,\mu_3,c_4,\lambda_4,\mu_4,N_3), \quad (11)\\
\widehat{X}_{3N_3}(s+1) &= 1-\frac{\widehat{TP}_{m,p,3}(s+1)}{c_3'''(s+1)e_3},\\
s &= 0,1,2,\dots,\\
\widehat{X}_{3N_3}(0) &= 0.
\end{aligned}
$$

2) Convergence: Let $\widehat{TP}_{s,p,i}$, $\widehat{TP}_{m,p,i}$, $i=1,2,3$, 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.

Theorem 1: Under assumptions 1)-6), Procedures 1 and 2 are convergent, therefore, the following limits exist:

$$\lim_{s\to\infty}\widehat{TP}_{s,p,i}(s)=\widehat{TP}_{s,p,i}, \lim_{s\to\infty}\widehat{TP}_{m,p,i}(s)=\widehat{TP}_{m,p,i},\ i=1,2,3. \quad (12)$$

In addition, the steady state equations of Procedures 1 and 2 have unique solutions.

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

$$\widehat{TP}_{s,p}=\widehat{TP}_{s,p,2}+\widehat{TP}_{s,p,3}, \qquad \widehat{TP}_{m,p}=\widehat{TP}_{m,p,3}. \quad (13)$$

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 and 60 merge systems by reversing the lines.

$$
\begin{aligned}
e_i &\in [0.75,0.95], i=1,\dots,4, \text{ where } e_i=\frac{\mu_i}{\lambda_i+\mu_i},\\
T_{down,i} &\in [2,10], i=1,\dots,4, \text{ where } \mu_i=1/T_{down,i},\\
c_i &\in [1,1.2], i=1,2 \text{ for split}, i=3,4 \text{ for merge}, \quad (14)\\
c_i &\in [0.6,0.8], i=3,4 \text{ for split}, i=1,2 \text{ for merge},\\
N_i &\in [1,3]\cdot T_{down,i}, i=1,\dots,4.
\end{aligned}
$$

Both analytical method using Procedures 1 and 2 and simulation approach using *Simul8* ([16]) are pursued to evaluate the throughput of each line. 10,000 time units of warmup period are assumed, and the next 100,000 units are used to collect steady state statistics. 20 replications are carried out to obtain the average production rate, with 95% confidence intervals typically ranging around ± 0.001. The differences between analytical and simulation results are evaluated as

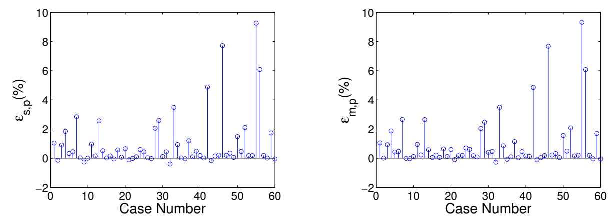
$$\varepsilon_{s,p}=\frac{\widehat{TP}_{s,p}-TP_{s,p}}{TP_{s,p}}\cdot 100\%, \quad \varepsilon_{m,p}=\frac{\widehat{TP}_{m,p}-TP_{m,p}}{TP_{m,p}}\cdot 100\%, \quad (15)$$

where $TP_{s,p}$ and $TPR_{m,p}$ are throughputs obtained by simulation for priority policy in split and merge systems, respectively (similar notations are used for subsequent procedures as well).

The results of this investigation are illustrated in Figure 3 for Procedures 1 and 2. It is shown that in most cases we studied, the error is within 3-5%, with a few exceptions up to 9%. Therefore, Procedures 1 and 2 provide a relative accurate approximation for system throughputs.

D. Circulate Policy

a) 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 b_2 and b_3. In Line 2, when m_2 is not starved, it is available to b_2 50% of time if b_3 is not full, and 100%



(a) Split system (b) Merge system

Fig. 3. Accuracy of Procedures 1 and 2

of time otherwise. Similar argument applies to Line 3. Thus, the recursive procedure is introduced as follows:

Procedure 3:

Line 1

$$
\begin{aligned}
c_2'(s+1) &= c_2(1-\widehat{X}_{2N_2}(s)\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{s,c,1}(s+1) &= TP(c_1,\lambda_1,\mu_1,c_2'(s+1),\lambda_2,\mu_2,N_1), \quad (16)\\
\widehat{X}_{10}(s+1) &= 1-\frac{\widehat{TP}_{s,c,1}(s+1)}{c_2'(s+1)e_2},
\end{aligned}
$$

Line 2

$$
\begin{aligned}
c_2''(s+1) &= 0.5c_2(1+\widehat{X}_{3N_3}(s))(1-\widehat{X}_{10}(s+1)),\\
\widehat{TP}_{s,c,2}(s+1) &= TP(c_2''(s+1),\lambda_2,\mu_2,c_3,\lambda_3,\mu_3,N_2), \quad (17)\\
\widehat{X}_{2N_2}(s+1) &= 1-\frac{\widehat{TP}_{s,c,2}(s+1)}{c_2''(s+1)e_2},
\end{aligned}
$$

Line 3

$$
\begin{aligned}
c_2'''(s+1) &= 0.5c_2(1+\widehat{X}_{2N_2}(s+1))(1-\widehat{X}_{10}(s+1)),\\
\widehat{TP}_{s,c,3}(s+1) &= TP(c_2'''(s+1),\lambda_2,\mu_2,c_4,\lambda_4,\mu_4,N_3), \quad (18)\\
\widehat{X}_{3N_3}(s+1) &= 1-\frac{\widehat{TP}_{s,c,3}(s+1)}{c_2'''(s+1)e_2},\\
s &= 0,1,2,\dots,\\
\widehat{X}_{2N_2}(0) &= \widehat{X}_{3N_3}(0)=0.
\end{aligned}
$$

b) Merge system: Similar procedure is developed.

Procedure 4:

Line 1

$$
\begin{aligned}
c_3'(s+1) &= 0.5c_3(1+\widehat{X}_{20}(s))(1-\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{m,c,1}(s+1) &= TP(c_1,\lambda_1,\mu_1,c_3'(s+1),\lambda_3,\mu_3,N_1), \quad (19)\\
\widehat{X}_{10}(s+1) &= 1-\frac{\widehat{TP}_{m,p,1}(s+1)}{c_3'(s+1)e_3},
\end{aligned}
$$

Line 2

$$
\begin{aligned}
c_3''(s+1) &= 0.5c_3(1+\widehat{X}_{10}(s+1))(1-\widehat{X}_{3N_3}(s)),\\
\widehat{TP}_{m,c,2}(s+1) &= TP(c_2,\lambda_2,\mu_2,c_3''(s+1),\lambda_3,\mu_3,N_2), \quad (20)\\
\widehat{X}_{20}(s+1) &= 1-\frac{\widehat{TP}_{m,p,2}(s+1)}{c_3''(s+1)e_3},
\end{aligned}
$$

Line 3

$$
\begin{aligned}
c_3'''(s+1) &= c_3(1-\widehat{X}_{10}(s+1)\widehat{X}_{20}(s+1)),\\
\widehat{TP}_{m,c,3}(s+1) &= TP(c_3'''(s+1),\lambda_3,\mu_3,c_4,\lambda_4,\mu_4,N_3), \quad (21)\\
\widehat{X}_{3N_3}(s+1) &= 1-\frac{\widehat{TP}_{m,p,3}(s+1)}{c_3'''(s+1)e_3},\\
s &= 0,1,2,\dots,\\
\widehat{X}_{20}(0) &= 0, \quad \widehat{X}_{3N_3}(0)=1.
\end{aligned}
$$

Theorem 2: Under assumptions 1)-6), Procedures 3 and 4 are convergent, therefore, the following limits exist:

$$\lim_{s\to\infty}\widehat{TP}_{s,c,i}(s)=\widehat{TP}_{s,c,i}, \lim_{s\to\infty}\widehat{TP}_{m,c,i}(s)=\widehat{TP}_{m,c,i},\ i=1,2,3. \quad (22)$$

In addition, the steady state equations of Procedures 3 and 4 have unique solutions.

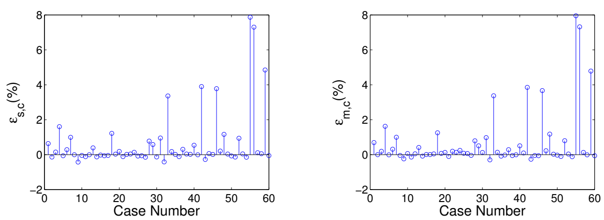
Therefore, throughput estimates, $\widehat{TP}_{s,c}$, $\widehat{TP}_{m,c}$ for split and merge systems with circulate policy, respectively, can be calculated:

$$\widehat{TP}_{s,c} = \widehat{TP}_{s,c,2} + \widehat{TP}_{s,c,3}, \qquad \widehat{TP}_{m,c} = \widehat{TP}_{m,c,3}. \tag{23}$$

Again the accuracy of estimates (23) is investigated numerically. Same split and merge systems as in Subsection III-C are used for accuracy analysis. The differences between analytical and simulation results are introduced as

$$\varepsilon_{s,c} = \frac{\widehat{TP}_{s,c} - TP_{s,c}}{TP_{s,c}} \cdot 100\%, \quad \varepsilon_{m,c} = \frac{\widehat{TP}_{m,c} - TP_{m,c}}{TP_{m,c}} \cdot 100\%. \tag{24}$$

As one can see that in most cases we studied, the errors are less than 2%, with a few cases up to 8%. Therefore, Procedures 1 and 2 again provide relative precise estimates.



(a) Split system (b) Merge system

Fig. 4. Accuracy of Procedures 3 and 4

E. Percentage Policy

Percentage policy has been studied in the literature. However, it is less popular due to implementation difficulty. In addition, since percentage merge is less encountered, only the percentage split policy is discussed in this work. A percentage policy implies that in a split system (Figure 1)(a), the parts flow into different downstream branches based on given percentage. However, due to possible blockages, the split station may need to wait until the downstream buffer has available space. Therefore, the final percentage of parts flow into different branches may not be the same as the capacity allocation on the split machine. To ensure that the final products consisting of parts $100 \cdot \alpha\%$ produced by m_3 and $100 \cdot (1-\alpha)\%$ by m_4, which agrees with the expectation of percentage policy, a new percentage of capacity allocation needs to be determined. Assume that $\beta \cdot 100\%$ of parts are intended to be sent to buffer b_2 and $(1-\beta) \cdot 100\%$ to b_3 by machine m_2. Then after possible blockages, the actual probability sending parts to b_2 and b_3 will be α and $1-\alpha$, respectively. Therefore, we need

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

which leads to

$$\beta = \frac{(1-\widehat{X}_{3N_3})\alpha}{1 + \widehat{X}_{2N_2}\alpha - \widehat{X}_{3N_3}\alpha}. \tag{25}$$

Thus, the recursive procedure for percentage split 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

$$\begin{aligned}
c_2'(s+1) &= c_2(1 - \beta(s+1)\widehat{X}_{2N_2}(s) - \\
&\quad (1-\beta(s+1))\widehat{X}_{3N_3}(s)), \\
\widehat{TP}_{s,\%,1}(s+1) &= TP(c_1, \lambda_1, \mu_1, c_2'(s+1), \lambda_2, \mu_2, N_1), \quad (26) \\
\widehat{X}_{10}(s+1) &= 1 - \frac{\widehat{TP}_{s,\%,1}(s+1)}{c_2'(s+1)e_2},
\end{aligned}$$

Line 2

$$\begin{aligned}
c_2''(s+1) &= \beta(s+1)c_2(1 - \widehat{X}_{10}(s+1)), \\
\widehat{TP}_{s,\%,2}(s+1) &= TP(c_2''(s+1), \lambda_2, \mu_2, c_3, \lambda_3, \mu_3, N_2), \quad (27) \\
\widehat{X}_{2N_2}(s+1) &= 1 - \frac{\widehat{TP}_{s,\%,2}(s+1)}{c_2''(s+1)e_2},
\end{aligned}$$

Line 3

$$\begin{aligned}
c_2'''(s+1) &= (1-\beta(s+1))c_2(1 - \widehat{X}_{10}(s+1)), \\
\widehat{TP}_{s,\%,3}(s+1) &= TP(c_2'''(s+1), \lambda_2, \mu_2, c_4, \lambda_4, \mu_4, N_3), \quad (28) \\
\widehat{X}_{3N_3}(s+1) &= 1 - \frac{\widehat{TP}_{s,\%,3}(s+1)}{c_2'''(s+1)e_2}, \\
s &= 0, 1, 2, \ldots, \\
\widehat{X}_{2N_2}(0) &= \widehat{X}_{3N_3}(0) = 0.
\end{aligned}$$

However, unlike the priority and circulate cases, the analytical proof of the convergence of Procedure 5 is not available now. Therefore, we justify the convergence numerically. It turns out that in all the examples we tested, the procedure converges. Therefore, we formulate it as a numerical fact.

Numerical Fact 1: Under assumptions i)-vi), Procedure 5 is convergent and the following limits exist:

$$\lim_{s\to\infty} \widehat{TP}_{s,\%,i}(s) = \widehat{TP}_{s,\%,i}, \qquad i = 1,2,3. \tag{29}$$

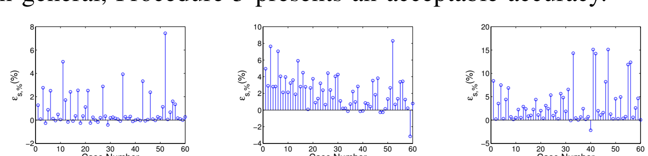
Then, under this Numerical Fact, the steady state equations of Procedure 5 have unique solutions. The estimate of throughput for the split system with percentage policy, $\widehat{TP}_{s,\%}$, in steady state is obtained.

$$\widehat{TP}_{s,\%} = \widehat{TP}_{s,\%,1} = \widehat{TP}_{s,\%,2} + \widehat{TP}_{s,\%,3}. \tag{30}$$

Similarly, the accuracy is defined as

$$\varepsilon_{s,\%} = \frac{\widehat{TP}_{s,\%} - TP_{s,\%}}{TP_{s,\%}} \cdot 100\%. \tag{31}$$

By selecting α as 10%, 30% and 50%, numerical experiments are carried out to investigate the accuracy of Procedure 5. Sixty lines defined in previous subsections are used for tests. The results are shown in Figure 5. As before, most cases result in errors less than 5%, but there exist a few cases where errors go up to 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.



(a) $\alpha = 10\%$ (a) $\alpha = 30\%$ (a) $\alpha = 50\%$

Fig. 5. Accuracy of Procedure 5

F. Strictly Circulate Policy

c) Split system: The strict circulate policy indicates that the split machine sends parts to downstream branches in circular mode without ignoring the blocked branches, which implies the machine waits until the downstream buffer is available. Such policy is similar to a percentage policy in the sense that $\alpha = 0.5$. Although in percentage policy, one may have the possibility that two consecutive parts will be sent to the same branch, while in strict circulate policy, it will never happen, the final results of parts flow distribution will be identical. Therefore, we use the same procedure to estimate the throughput of split system by assuming $\alpha = 0.5$, and denote $\widehat{TP}_{s,sc,i}(s)$ as the throughput of Line i with strictly circulate policy at iteration i.

d) Merge system: Although percentage policy is seldom used for production merge, strictly circulate policy is popular in many merge systems. For example, if a desired production sequence needs to be followed, strictly circulate policy can be adopted. Similar to split system, a new allocation of capacity (rather than 0.5) needs to be computed in order to ensure the strictly circulation.

Procedure 6:

$$\begin{aligned}
\beta(s+1) &= \frac{0.5(1-\widehat{X}_{20}(s))}{1+0.5\widehat{X}_{10}(s)-0.5\widehat{X}_{20}(s)}, \\
&\text{Line 1} \\
c_3'(s+1) &= \beta(s+1)c_3(1-\widehat{X}_{3N_3}(s+1)), \\
\widehat{TP}_{m,sc,1}(s+1) &= TP(c_1,\lambda_1,\mu_1,c_3'(s+1),\lambda_3,\mu_3,N_1), \quad (32) \\
\widehat{X}_{10}(s+1) &= 1-\frac{\widehat{TP}_{m,sc,1}(s+1)}{c_3'(s+1)e_3}, \\
&\text{Line 2} \\
c_3''(s+1) &= (1-\beta(s+1))c_3(1-\widehat{X}_{3N_3}(s+1)), \\
\widehat{TP}_{m,c,2}(s+1) &= TP(c_2,\lambda_2,\mu_2,c_3''(s+1),\lambda_3,\mu_3,N_2), \quad (33) \\
\widehat{X}_{20}(s+1) &= 1-\frac{\widehat{TP}_{m,sc,2}(s+1)}{c_3''(s+1)e_3}, \\
&\text{Line 3} \\
c_3'''(s+1) &= c_3(1-\beta(s+1)\widehat{X}_{10}(s) \\
&\quad -(1-\beta(s+1))\widehat{X}_{20}(s)), \\
\widehat{TP}_{m,sc,3}(s+1) &= TP([c_3'''(s+1),\lambda_3,\mu_3,c_4,\lambda_4,\mu_4,N_3), \quad (34) \\
\widehat{X}_{3N_3}(s+1) &= 1-\frac{\widehat{TP}_{m,sc,3}(s+1)}{c_3'''(s+1)e_3}, \\
s &= 0,1,2,\ldots, \\
\widehat{X}_{10}(0) &= \widehat{X}_{20}(0)=0.
\end{aligned}$$

Again the convergence of Procedures 5 and 6 are justified through Numerical Fact 1 and the uniqueness of the solution follows immediately. Then

$$\lim_{s\to\infty}\widehat{TP}_{s,sc,i}(s)=\widehat{TP}_{s,sc,i}, \lim_{s\to\infty}\widehat{TP}_{m,sc,i}(s)=\widehat{TP}_{m,sc,i}, i=1,2,3. \quad (35)$$

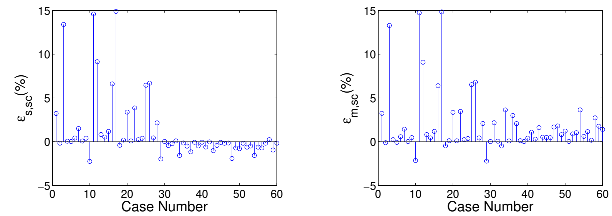
The estimates of system throughput will be

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

Define the accuracy of the estimates

$$\varepsilon_{s,sc}=\frac{\widehat{TP}_{s,sc}-TP_{s,sc}}{TP_{s,sc}}\cdot 100\%, \quad \varepsilon_{m,sc}=\frac{\widehat{TP}_{m,sc}-TP_{m,sc}}{TP_{m,sc}}\cdot 100\%. \quad (37)$$

The same 60 lines are used for accuracy justification. Figure 6 illustrates the results. Again in most cases, the accuracy is within 4%, with a few exceptions up to 15%. Therefore, we conclude that both procedures can be used for performance estimation of split and merge system with strict circulate policies.



(a) Split system (b) Merge system

Fig. 6. Accuracy of Procedures 5 ($\alpha = 0.5$) and 6

IV. CONCLUSIONS

This paper presents analytical methods to approximate the throughput of split and merge systems with exponential reliability machines. Priority, circulate, strictly circulate and percentage policies are discussed. It is shown that these methods provide acceptable accuracy (in most cases less than 5%) for throughput estimation.

REFERENCES

[1] Y. Dallery and S.B. Gershwin, "Manufacturing Flow Line Systems: A Review of Models and Analytical Results," *Queuing Sys.*, vol. 12, pp. 3-94, 1992.

[2] J. Li, D.E. Blumenfeld, N. Huang and J.M. Alden, "Throughput Analysis in Production Systems: Recent Advances and Future Topics," to appear in *Int. J. of Prod. Res.*, 2008.

[3] J.A. Buzacott and J.G. Shantikumar, *Stochastic Models of Manufacturing Systems*, Prentice Hall, 1993.

[4] S.B. Gershwin, *Manufacturing Systems Engineering*, PTR Prentice Hall, 1994.

[5] J. Li and S.M. Meerkov, *Production Systems Engineering*, Preliminary Edition, Third Printing, WingSpan Press, 2008.

[6] S. Helber, "Approximate Analysis of Unreliable Transfer Lines with Splits in the Flow of Material", *Annals of Oper. Res.*, vol. 93, pp. 217-243, 2000.

[7] B. Tan, "A Three-Station Merge System with Unreliable Stations and a Shared Buffer", *Math. & Comp. Mod.*, vol. 33, pp. 1011-1026, 2001.

[8] A.C. Diamantidis and C.T. Papadopoulos, "Markovian Analysis of a Discrete Material Manufacturing System with Merge Operations, Operation-Dependent and Idleness Failures", *Comp. & Ind. Eng.*, vol. 50, pp. 466-487, 2006.

[9] J. Li and N. Huang, "Modeling and Analysis of a Multiple Product Manufacturing System with Split and Merge," *Int. J. of Prod. Res.*, vol. 43, pp. 4049-4066, 2005.

[10] J. Li, "Performance Analysis of Production Systems with Rework Loops," *IIE Trans.*, vol. 36, pp. 755-765, 2004.

[11] J. Li, "Throughput Analysis in Automotive Paint Shops: A Case Study," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 1, pp. 90-98, 2004.

[12] J. Li, "Modeling and Analysis of Manufacturing Systems with Parallel Lines", *IEEE Trans. on Autom. Ctrl.*, vol. 49, pp. 1824-1829, 2004.

[13] J. Li, "Overlapping Decomposition: A System-Theoretic Method for Modeling and Analysis of Complex Production Systems," *IEEE Trans. on Autom. Sci. and Eng.*, vol. 2, pp. 40-53, 2005.

[14] Y. Liu and J. Li, "Modeling and Analysis of Split and Merge Systems with Bernoulli Reliability Machines," to appear in *Int J. of Prod. Res.*, 2008.

[15] Y. Liu and J. Li, "Production Split and Merge Systems: Performance Analysis, Policy Comparison and Structural Properties," *Report PSSL-08-03*, Dept. of ECE, Univ. of Kentucky, Lexington, KY, 2008.

[16] J.W. Haige and K.N. Paige, *Learning Simul8: the Complete Guide*, Plain Vu, 2001.