International Journal of Production Research,
Vol. 43, No. 19, 1 October 2005, 4049–4066

Taylor & Francis
Taylor & Francis Group

Modelling and analysis of a multiple product manufacturing system with split and merge

JINGSHAN LI* and NINGJIAN HUANG

Manufacturing Systems Research Laboratory, General Motors Research & Development Center, Mail Code 480-106-359, 30500 Mound Road, Warren, MI 48090-9055, USA

(*Received September 2004*)

To satisfy a rapidly changing market and various requirements of customer demand, many automotive assembly systems are designed to increase flexibility. One way to achieve this is to have a system making multiple products through split and merge. This paper presents an iterative approach to model such a kind of flexible manufacturing system with common lines and dedicated branches to process different products. The mathematical procedures, associated with the justification of convergence and accuracy, are provided. A case study is presented to illustrate the applicability of the method.

Keywords: Overlapping decomposition; Aggregation; Multiple products; Split; Merge; Flexibility

1. Introduction

Flexibility allows a company to adjust its capacity and resources to make different products in order to satisfy rapid market changes and various customer demand. This has challenged the automotive industry due to many practical constraints that limit flexibility. For example, a high-volume body shop normally is not designed to produce vehicles that differ significantly in size and style. A system that can produce any vehicle at any volume requires very high investment on flexible tooling and fixtures and also very high maintenance costs. As an alternative, automotive companies apply limited flexibility to their body shops to produce several distinguished, yet closely related, models in one system. One approach to achieve this is to use split and merge systems shown in figure 1, where two different products, A and C, are produced. In figure 1, the circles represent the machines and rectangles are the buffers. Dedicated production lines A and C are designed for specific processing of the respective products, while the common processes before the split and after the merge are shared for both products. At stations *Start* to *Split*, and stations *Merge* to *Finish*, part types A and C have identical operations. At *Split* station, A and C type parts will be specifically routed to its own line, for product-specific processing, respectively, i.e., type A parts have to go through operations in Line A, and type C parts follow operations in Line C. At *Merge* station, both A and C type parts will be

*Corresponding author. Email: jingshan.li@gm.com

International Journal of Production Research
ISSN 0020–7543 print/ISSN 1366–588X online © 2005 Taylor & Francis Group Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540500031899

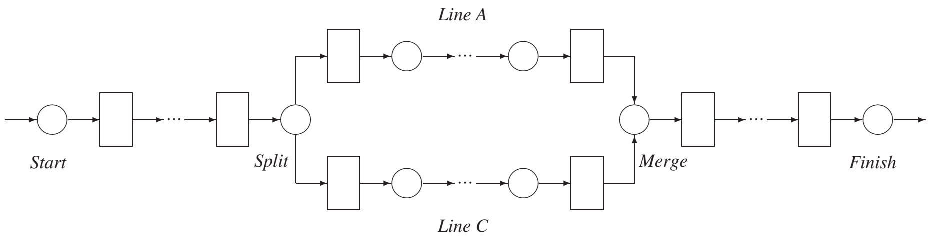

Figure 1. Multiple product split and merge system.

loaded sequentially for remaining processes (which are identical to both *A* and *C* types) until the end of the line.

Such system layouts are used in body shops of different automotive companies. Moreover, in some automotive paint shops, in order to maintain high paint quality, similar system structures are applied where cars or trucks are regrouped by colours and processed at specific painting booths and ovens. Such systems have obvious advantages, for instance, straightforward design, easy floor control, less expense, etc. However, one of the disadvantages is that it is difficult to analyse effectively system performance due to a lack of analytical methods and tools.

Performance analysis of single-product manufacturing systems has attracted tremendous attention (see reviews by Dallery and Gershwin 1992; monographs by Buzacott and Shanthikumar 1993; Gershwin 1994; and representative papers by Dallery *et al.* 1988, 1989, Di Mascolo *et al.* 1991, Jacobs and Meerkov 1995, Chiang *et al.* 2000a, b, 2001, Li 2003, 2004a–c, respectively). However, the study of multiple-product manufacturing systems is quite limited. Nemec (1999) presents a decomposition method to analyse a synchronous transfer line which processes multiple products through dedicated buffers. A similar problem is addressed by Colledani *et al.* (2003) with a different scheduling rule. Diamantidis and Papadopoulos (2003) discuss a discrete unreliable merge system with a finite intermediate buffer, which can be extended to process multiple products. Another decomposition method for a deterministic multiple-part-type multiple-failure-mode line is described by Syrowicz (1999), still dedicated buffers are assigned for different products. In addition, Dallery (1999) studies a split and merge system by transforming it into a disassembly/assembly system under the assumption that failures and repairs of different machines have to occur exactly at the same instant in both models. To our best knowledge, no effective analytical methods are available in the current literature to analyse the performance of the multiple-product manufacturing system illustrated in figure 1.

In order to design an effective and productive multiple-product manufacturing system, quick and accurate analysis of the performance of such systems is necessary and important. The main contribution of this paper is the development of a system-theoretic approach for modelling and analysis of such systems. In this paper, convergence and accuracy of the mathematical models or procedures are studied, and it is shown that the system performance can be effectively analysed with satisfactory accuracy. Moreover, a case study at an automotive body shop is introduced to illustrate the applicability of the method.

The paper is structured as follows. Section 2 formulates the problem. The recursive procedure of performance evaluation is introduced in section 3.

Section 4 outlines a case study at an automotive body assembly system. The conclusions are formulated in section 5. All proofs are given in the Appendix.

2. Problem formulation

Consider a two-product manufacturing system shown in figure 2. The system consists of one common main line (machines m_1 to m_k, m_{k+1} to m_M, buffers B_1 to B_{k-1}, B_{k+1} to B_{M-1}), and two dedicated lines (machines $m_{i,1}$ to m_{i,M_i}, $i = a, c$, buffers $B_{i,1}$ to B_{i,M_i+1}) for part types A and C, respectively. At machines m_1 to m_k, m_{k+1} to m_M, part types A and C have identical operations. However, type A parts have to go through machines $m_{a,1}$ to m_{a,M_a}, and type C parts follow $m_{c,1}$ to m_{c,M_c}. The following assumptions pertaining to the machines, buffers and their interactions are introduced:

(i) Each machine, $m_1, \ldots, m_M$, $m_{j,1}, \ldots, m_{j,M_j}$, $j = a, c$, has two states: up and down. When up, machine m_i, $i = 1, \ldots, M$, or $m_{j,n}$, $j = a, c$, $n = 1, \ldots, M_j$, is capable of producing with the rate S_i or $S_{j,n}$ parts per unit of time (cycle), respectively. When the machine is down, no production takes place.

(ii) The uptime and the downtime of each machine, m_i, $i = 1, \ldots, M$, or $m_{j,n}$, $j = a, c$, $n = 1, \ldots, M_j$, are random variables distributed exponentially with parameters p_i and r_i, or $p_{j,n}$ and $r_{j,n}$, respectively.

Remark 1: Assumption (ii) implies that p_i and r_i (respectively, $p_{j,n}$ and $r_{j,n}$) are the failure and repair rates of machine m_i (respectively, machine $m_{j,n}$), respectively. In other words, m_i's (respectively, $m_{j,n}$'s) average up- and downtime are $1/p_i$ and $1/r_i$ (respectively, $1/p_{j,n}$ and $1/r_{j,n}$), respectively. The exponential assumption is used to simplify the analysis. An investigation of the validation of the exponential assumption in production systems research is presented by Inman (1999). □

(iii) Each buffer, B_i, $i = 1, \ldots, k-1, k+1, \ldots, M-1$, or $B_{j,n}$, $j = a, c$, $n = 1, \ldots, M_j + 1$, is characterized by its capacity N_i, or $N_{j,n}$, respectively. In addition, $0 < N_i < \infty$ or $0 < N_{j,n} < \infty$.

(iv) Part type A consists of $\gamma \cdot 100\%$ of all products, $0 < \gamma < 1$, the rest are type C. At machine m_k, type A parts must be routed to line A through $B_{a,1}$ if it is not full. Type C parts are sent to line C through $B_{c,1}$ if it has available spaces. There is no sequencing requirement for the system.

(v) Machine m_1 has probability γ to load part type A and $1 - \gamma$ loading part type C each cycle.

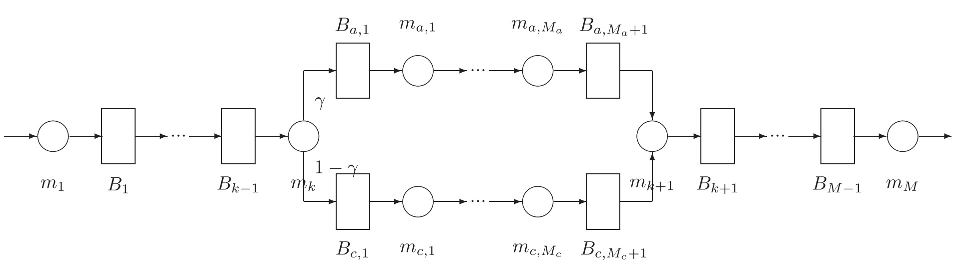

Figure 2. Structure of a two-product manufacturing system.

(vi) Machine m_i is blocked at time t if downstream buffer is full at time t. Machine m_M is never blocked. Machine m_k is blocked by line A if it produces part type A and $B_{a,1}$ is full, while machine m_k is blocked by line C if it produces part type C and $B_{c,1}$ is full.

(vii) Machine m_{k+1} takes parts from B_{a,M_a+1} or B_{c,M_c+1} alternatively if they are not empty (in other words, m_{k+1} loads one part from line A, then another part from line C if both are available). When either buffer is empty, m_{k+1} keeps loading from the other one.

Remark 2: Assumption (vii) implies that if both buffers are not empty, part types A and C will have equal probability to be loaded by machine m_{k+1}. If one of the buffers is empty, the other one has 100% probability being loaded. Such policy is also referred to as circulating in some simulation packages. □

(viii) Machine m_i is starved at time t if upstream buffer is empty at time t. Machine m_{k+1} is starved if both B_{a,M_a+1} and B_{c,M_c+1} are empty. Machine m_1 is never starved.

The problem addressed is as follows:

Problem: *Given production system* (i–viii), *develop a method for evaluating the throughput as a function of the system parameters.*

The solution to the problem described above is given in section 3.

3. Performance evaluation

3.1 *Approach outline*

A system-theoretic method, referred to as *overlapping decomposition*, is presented by Li (2003, 2004a–c) to analyse the performance of complex production systems, for instance, systems with rework loops, parallel lines, assembly/disassembly, feed-forward lines, etc. The main idea of the method is to decompose the complex system into a set of overlapped serial production lines, adjust the overlapping machines to accommodate the effects of machines and buffers in other lines, and introduce a recursive procedure to approximate the system performance. Finally, the system throughput can be calculated when the procedure converges. In this paper, we apply this idea in analysing system (i–viii). Detailed analysis introduced below are designed specifically for this problem. To this end, we decompose the system into four serial lines (figure 3), where machines m_k and m_{k+1} are the overlapping machines:

$$\begin{array}{lll} \text{Line 1:} & m_1, \ldots, m_k, & B_1, \ldots, B_{k-1}; \\ \text{Line 2:} & m_{k+1}, \ldots, m_M, & B_{k+1}, \ldots, B_{M-1}; \\ \text{Line 3:} & m_k, m_{a,1}, \ldots, m_{a,M_a}, m_{k+1}, & B_{a,1}, \ldots, B_{a,M_a+1}; \\ \text{Line 4:} & m_k, m_{c,1}, \ldots, m_{c,M_c}, m_{k+1}, & B_{c,1}, \ldots, B_{c,M_c+1}. \end{array}$$

If we know the throughput of each serial line, the system throughput (which equals to that of line 2) is obtained. To calculate the throughput of each serial line, the first machine's blockage probability and the last machine's starvation probability have to be known. Therefore we introduce *virtual* machines by taking into account these probabilities to replace the first and last machines, respectively, and

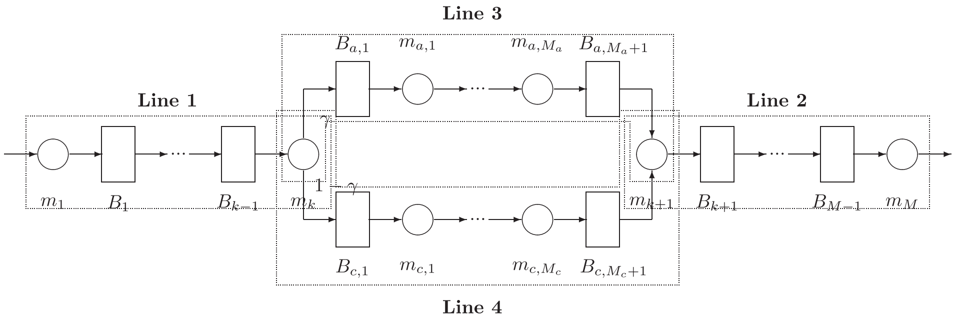

Figure 3. Overlapping decomposition of systems (i–viii).

to calculate the line throughput and the probabilities of blockage and starvation of the first and last machine in each serial line, respectively. Since these probabilities are unknown, we introduce iterations.

First, we consider serial lines 1 and 2. Introduce virtual machine m_k^1 and m_{k+1}^2 (where the superscripts denote the line numbers), which can be 'viewed' as machines that are never blocked and starved, respectively. In other words, we consider the blockage time of m_k and starvation time of m_{k+1} as m_k^1's and m_{k+1}^2's downtime, respectively. More specifically, we define:

$$r_k^1 = r_k[1 - \gamma \text{Prob}\{m_k \text{ is blocked by } B_{a,1}\} - (1-\gamma)\text{Prob}\{m_k \text{ is blocked by } B_{c,1}\}],$$
$$r_{k+1}^2 = r_{k+1}(1 - \text{Prob}\{m_{k+1} \text{ is starved by } B_{a,M_a+1}\}\text{Prob}\{m_{k+1} \text{ is starved by } B_{c,M_c+1}\}).$$

Then we select p_k^1 and p_{k+1}^2 to satisfy the conservation of flow and obtain:

$$p_k^1 = p_k + r_k - r_k^1,$$
$$p_{k+1}^2 = p_{k+1} + r_{k+1} - r_{k+1}^2.$$

Now using a serial line analysis method (which is described in detail in subsection 3.3), we can calculate the throughputs of lines 1 and 2, and the probabilities that m_k is starved and m_{k+1} is blocked, respectively.

Next consider lines 3 and 4, introduce virtual machines m_k^j and m_{k+1}^j, $j = 3, 4$, we have:

$$r_k^3 = r_k \gamma \text{Prob}\{m_k \text{ is not starved}\},$$
$$\begin{aligned} r_{k+1}^3 &= r_{k+1}\text{Prob}\{m_{k+1} \text{ is not blocked}\}(\text{Prob}\{m_{k+1} \text{ is starved by } B_{c,M_c+1}\} \\ &\quad + 0.5\text{Prob}\{m_{k+1} \text{ is not starved by } B_{c,M_c+1}\}) \\ &= 0.5 r_{k+1}\text{Prob}\{m_{k+1} \text{ is not blocked}\}(\text{Prob}\{m_{k+1} \text{ is starved by } B_{c,M_c+1}\} + 1), \end{aligned}$$
$$r_k^4 = r_k(1-\gamma)\text{Prob}\{m_k \text{ is not starved}\},$$
$$\begin{aligned} r_{k+1}^4 &= r_{k+1}\text{Prob}\{m_{k+1} \text{ is not blocked}\}(\text{Prob}\{m_{k+1} \text{ is starved by } B_{a,M_a+1}\} \\ &\quad + 0.5\text{Prob}\{m_{k+1} \text{ is not starved by } B_{a,M_a+1}\}) \\ &= 0.5 r_{k+1}\text{Prob}\{m_{k+1} \text{ is not blocked}\}(\text{Prob}\{m_{k+1} \text{ is starved by } B_{a,M_a+1}\} + 1), \end{aligned}$$
$$p_i^j = p_i + r_i - r_i^j, \quad i = k, k+1, \;\; j = a, c.$$

Note that for line 3, m_{k+1} has 100% probability to load from B_{a,M_a+1} if it is starved by B_{c,M_c+1} and only 50% probability when it is not starved by B_{c,M_c+1}.

Similarly, for line 4, m_{k+1} has 100% and 50% probabilities to load from B_{c,M_c+1} if B_{a,M_a+1} is empty and not empty, respectively. Therefore, Prob$\{m_k$ is blocked by $B_{a,1}\}$, Prob$\{m_{k+1}$ is starved by $B_{a,M_a+1}\}$, and Prob$\{m_k$ is blocked by $B_{c,1}\}$, Prob$\{m_{k+1}$ is starved by $B_{c,M_c+1}\}$ can be calculated. Now use these probabilities, and repeat this process by alternating among lines 1–4 until it is convergent. Finally we obtain the throughput estimation of the system. The recursive procedures implementing this method are introduced next.

3.2 *Recursive procedures*

Introduce the following notation:

$$\begin{aligned} b_{k,j} &= \text{Prob}\{\text{machine } m_k \text{ is blocked by } B_{j,1}\}, \quad j=a,c,\\ s_{k+1,j} &= \text{Prob}\{\text{machine } m_{k+1} \text{ is starved by } B_{j,M_j+1}\}, \quad j=a,c,\\ b_{k+1} &= \text{Prob}\{\text{machine } m_{k+1} \text{ is blocked}\},\\ s_k &= \text{Prob}\{\text{machine } m_k \text{ is starved}\}. \end{aligned}$$

Let $TP(\cdot)$ denote the procedure to calculate the throughput of a serial line, and ρ_i be the throughput of line i. Introduce operators Φ_1 and Φ_2 represent the calculations of the blocking and starving probabilities of the first and last machines in a serial line, respectively. Formally, the recursive procedure is as follows.

Procedure 1:

Line 2

$$\begin{aligned} r^2_{k+1}(n+1) &= r_{k+1}[1-s_{k+1,a}(n)s_{k+1,c}(n)],\\ p^2_{k+1}(n+1) &= p_{k+1}+r_{k+1}-r^2_{k+1}(n+1),\\ \widehat{\rho}_2(n+1) &= TP\Big(p^2_{k+1}(n+1), r^2_{k+1}(n+1), S_{k+1},\ldots,p_M,r_M,S_M,N_{k+1},\ldots,N_{M-1}\Big),\\ b_{k+1}(n+1) &= \Phi_1\Big(p^2_{k+1}(n+1), r^2_{k+1}(n+1), S_{k+1},\ldots,p_M,r_M,S_M,N_{k+1},\ldots,N_{M-1}\Big), \end{aligned} \tag{1}$$

Line 1

$$\begin{aligned} r^1_k(n+1) &= r_k[1-\gamma b_{k,a}(n)-(1-\gamma)b_{k,c}(n)],\\ p^1_k(n+1) &= p_k+r_k-r^1_k(n+1),\\ \widehat{\rho}_1(n+1) &= TP\Big(p_1,r_1,S_1,\ldots,p^1_k(n+1),r^1_k(n+1),S_k,N_1,\ldots,N_{k-1}\Big),\\ s_k(n+1) &= \Phi_2\Big(p_1,r_1,S_1,\ldots,p^1_k(n+1),r^1_k(n+1),S_k,N_1,\ldots,N_{k-1}\Big), \end{aligned} \tag{2}$$

Line 3

$$\begin{aligned} r^3_k(n+1) &= r_k(1-\gamma)[1-s_k(n+1)],\\ p^3_k(n+1) &= p_k+r_k-r^3_k(n+1),\\ r^3_{k+1}(n+1) &= 0.5r_{k+1}[1+s_{k+1,c}(n)][1-b_{k+1}(n+1)],\\ p^3_{k+1}(n+1) &= p_{k+1}+r_{k+1}-r^3_{k+1}(n+1), \end{aligned} \tag{3}$$

$$\widehat{\rho}_3(n+1) = TP\Big(p_k^3(n+1), r_k^3(n+1), S_k, p_{a,1}, r_{a,1}, S_{a,1}, \ldots, p_{a,M_a}, r_{a,M_a}, S_{a,M_a}, p_{k+1}^3(n+1), r_{k+1}^3(n+1), S_{k+1}, N_{a,1}, \ldots, N_{a,M_a+1}\Big),$$

$$b_{k,a}(n+1) = \Phi_1\Big(p_k^3(n+1), r_k^3(n+1), S_k, p_{a,1}, r_{a,1}, S_{a,1}, \ldots, p_{a,M_a}, r_{a,M_a}, S_{a,M_a}, p_{k+1}^3(n+1), r_{k+1}^3(n+1), S_{k+1}, N_{a,1}, \ldots, N_{a,M_a+1}\Big),$$

$$s_{k+1,a}(n+1) = \Phi_2\Big(p_k^3(n+1), r_k^3(n+1), S_k, p_{a,1}, r_{a,1}, S_{a,1}, \ldots, p_{a,M_a}, r_{1,M_a}, S_{a,M_a}, p_{k+1}^3(n+1), r_{k+1}^3(n+1), S_{k+1}, N_{a,1}, \ldots, N_{a,M_1+1}\Big),$$

Line 4

$$
\begin{aligned}
r_k^4(n+1) &= r_k\gamma[1 - s_k(n+1)],\\
p_k^4(n+1) &= p_k + r_k - r_k^4(n+1),\\
r_{k+1}^4(n+1) &= 0.5r_{k+1}[1 + s_{k+1,a}(n+1)][1 - b_{k+1}(n+1)],\\
p_{k+1}^4(n+1) &= p_{k+1} + r_{k+1} - r_{k+1}^4(n+1), \qquad (4)\\
\widehat{\rho}_4(n+1) &= TP\Big(p_k^4(n+1), r_k^4(n+1), S_k, p_{c,1}, r_{c,1}, S_{c,1}, \ldots, p_{c,M_c}, r_{c,M_c}, S_{c,M_c}, p_{k+1}^4(n+1), r_{k+1}^4(n+1), S_{k+1}, N_{c,1}, \ldots, N_{c,M_c+1}\Big),\\
b_{k,c}(n+1) &= \Phi_1\Big(p_k^4(n+1), r_k^4(n+1), S_k, p_{c,1}, r_{c,1}, S_{c,1}, \ldots, p_{c,M_c}, r_{c,M_c}, S_{c,M_c}, p_{k+1}^4(n+1), r_{k+1}^4(n+1), S_{k+1}, N_{c,1}, \ldots, N_{c,M_c+1}\Big),\\
s_{k+1,c}(n+1) &= \Phi_2\Big(p_k^4(n+1), r_k^4(n+1), S_k, p_{c,1}, r_{c,1}, S_{c,1}, \ldots, p_{c,M_c}, r_{c,M_c}, S_{c,M_c}, p_{k+1}^4(n+1), r_{k+1}^4(n+1), S_{k+1}, N_{c,1}, \ldots, N_{c,M_c+1}\Big),
\end{aligned}
$$

$$n = 0, 1, 2, \ldots,$$

with initial conditions

$$s_{k+1,j}(0) = 0, \quad b_{k,j}(0) = 1, \quad j = a, c.$$

Here $TP(\cdot)$, $\Phi_1(\cdot)$ and $\Phi_2(\cdot)$ are calculated through the aggregation procedure developed by Jacobs (1993). A detailed description is presented next.

3.3 *Operators*

Consider a serial production line consisting of M machines with parameters p_1, r_1, $S_1, \ldots, p_M$, r_M, S_M, and $M-1$ in-process buffers with capacities $N_1, \ldots, N_{M-1}$. According to Jacobs (1993), its performance can be analysed using the recursive procedure introduced below.

Basically, the procedure consists of two aggregations: forward aggregation and backward aggregation. In the backward aggregation, the last two machines, m_M and m_{M-1}, are aggregated into a single machine, m_{M-1}^b. Next m_{M-1}^b is aggregated with m_{M-2} to result in m_{M-2}^b, and so on until all M machines are aggregated into a single one m_1^b. Then, in the forward aggregation, the first machine, m_1, is aggregated with m_2^b to generate m_2^f and so on until all machines are again aggregated into a single machine m_M^f. Then the procedure is repeated again. In addition, parameters ρ_i^f, ρ_i^b, ν_i^f and ν_i^b are introduced to characterize the mean and variance of the throughput of the forward and backward aggregated machine, respectively, and b_i^b and s_i^f are blockage and starvation parameters, respectively. Formally, this process can be represented as follows:

Procedure 2:

$$\text{Backward aggregation} \quad i = M-1, \ldots, 1,$$

$$\begin{aligned}
b_i^b(l+1) &= 1 - \frac{TP_2(p_i^f(l), r_i^f(l), S_i^f(l), p_{i+1}^b(l+1), r_{i+1}^b(l+1), S_{i+1}^b(l+1), N_i)}{\rho_i^f(l)},\\
\rho_i^b(l+1) &= \rho_i[1 - b_i^b(l+1)],\\
\nu_i^b(l+1) &= \nu_i[1 - b_i^b(l+1)] + \nu_{i+1}^b(l+1)b_i^b(l+1),\\
S_i^b(l+1) &= \begin{cases} S_i, & \text{if } S_{i+1}^b(l+1) \geq S_i,\\ S_i[1 - b_i^b(l+1)e_i] + S_{i+1}^b(l+1)b_i^b(l+1)e_i, & \text{if } S_{i+1}^b(l+1) < S_i,\end{cases}\\
r_i^b(l+1) &= \frac{2[\rho_i^b(l+1)]^2[S_i^b(l+1) - \rho_i^b(l+1)]}{S_i^b(l+1)\nu_i^b(l+1)},\\
p_i^b(l+1) &= \frac{2\rho_i^b(l+1)[S_i^b(l+1) - \rho_i^b(l+1)]^2}{S_i^b(l+1)\nu_i^b(l+1)},
\end{aligned} \tag{5}$$

$$\text{forward aggregation} \quad i = 1, \ldots, M-1,$$

$$\begin{aligned}
s_{i+1}^f(l+1) &= 1 - \frac{TP_2(p_i^f(l+1), r_i^f(l+1), S_i^f(l+1), p_{i+1}^b(l+1), r_{i+1}^b(l+1), S_{i+1}^b(l+1), N_i)}{\rho_{i+1}^b(l+1)},\\
\rho_{i+1}^f(l+1) &= \rho_{i+1}[1 - s_{i+1}^f(l+1)],\\
\nu_{i+1}^f(l+1) &= \nu_{i+1}[1 - s_{i+1}^f(l+1)] + \nu_i^f(l+1)s_{i+1}^f(l+1),\\
S_{i+1}^f(l+1) &= \begin{cases} S_{i+1}, & \text{if } S_i^f(l+1) \geq S_{i+1},\\ S_{i+1}[1 - s_{i+1}^f(l+1)e_{i+1}] + S_i^f(l+1)s_{i+1}^f(l+1)e_{i+1}, & \text{if } S_i^f(l+1) < S_{i+1},\end{cases}\\
r_{i+1}^f(l+1) &= \frac{2[\rho_{i+1}^f(l+1)]^2[S_{i+1}^f(l+1) - \rho_{i+1}^f(l+1)]}{S_{i+1}^f(l+1)\nu_{i+1}^f(l+1)},\\
p_{i+1}^f(l+1) &= \frac{2\rho_{i+1}^f(l+1)[S_{i+1}^f(l+1) - \rho_{i+1}^f(l+1)]^2}{S_{i+1}^f(l+1)\nu_{i+1}^f(l+1)},
\end{aligned} \tag{6}$$

where

$$e_i = \frac{r_i}{p_i + r_i}, \quad \rho_i = S_i e_i, \quad \nu_i = \frac{2S_i^2 r_i p_i}{(r_i + p_i)^3}, \quad i = 1, \ldots, M, \tag{7}$$

with boundary conditions

$$\begin{aligned}
&p_1^f(l) = p_1, \quad r_1^f(l) = r_1, \quad S_1^f(l) = S_1, \quad \rho_1^f(l) = S_1 e_1, \quad \nu_1^f(l) = \nu_1,\\
&p_M^b(l) = p_M, \quad r_M^b(l) = r_M, \quad S_M^f(l) = S_M, \quad \rho_M^b(l) = S_M e_M, \quad \nu_M^b(l) = \nu_M,
\end{aligned}$$

and initial conditions

$$p_i^f(0)=p_i,\quad r_i^f(0)=r_i,\quad S_i^f(0)=S_i,\quad \rho_i^f(0)=S_ie_i,\quad \nu_i^f(0)=\nu_i,$$
$$p_i^b(0)=p_i,\quad r_i^b(0)=r_i,\quad S_i^f(0)=S_i,\quad \rho_i^b(0)=S_ie_i,\quad \nu_i^b(0)=\nu_i,\quad i=1,\dots,M,$$

and $TP_2(p_1,r_1,S_1,p_2,r_2,S_2,N)$ is defined by (Jacobs 1993, Chiang *et al.* 2000a, 2001).

- *Case 1*: $S_1=S_2=S$

$$\rho=\begin{cases}\dfrac{r_1r_2S[p_1(p_2+r_2)-p_2(p_1+r_1)e^{-\beta N}]}{(p_1+r_1)(p_1+r_2)(p_1r_2-p_2r_1e^{-\beta N})}, & \text{if } \dfrac{p_1}{r_1}\neq\dfrac{p_2}{r_2},\\[2ex] \dfrac{Sr_2^2(r_1+r_2)+Nr_1r_2(p_2+r_2)^2}{(p_2+r_2)[S(r_1+r_2)+Nr_1(p_2+r_2)]}S, & \text{if } \dfrac{p_1}{r_1}=\dfrac{p_2}{r_2},\end{cases}\tag{8}$$

where

$$\beta=\frac{(r_1+r_2+p_1+p_2)(p_1r_2-p_2r_1)}{(r_1+r_2)(p_1+p_2)S}.$$

- *Case 2*: $S_1<S_2$

$$\rho=\frac{S_2e_2Ae^{\gamma_1N}+S_1e_1Be^{\gamma_2N}+S_1e_1Ce^{-\gamma_2N}}{Ae^{\gamma_1N}+Be^{\gamma_2N}+Ce^{-\gamma_2N}},\tag{9}$$

where

$$\begin{aligned}A&=r_1Q^2+r_1Q[S_1(r_1+r_2+p_2)-S_2(r_1+r_2+p_1)],\\ B&=r_2p_1S_2[(S_1-S_2)(r_1-r_2)-(S_2p_1+S_1p_2)-Q],\\ C&=\frac{e_2^2(S_2-S_1e_1)A+S_1e_1(1-S_2)B}{S_1e_1(e_2-1)},\\ \gamma_1&=\frac{1}{2S_1S_2(r_1+r_2)(S_1-S_2)}\Big[r_1S_1^2(r_1+r_2+p_2)+r_2S_2^2(r_1+p_1+r_2)\\ &\qquad -S_1S_2[(r_1+r_2)^2+(r_1+r_2)(p_1+p_2)+(r_1p_2+r_2p_1)]\Big],\\ \gamma_2&=\frac{(S_1r_1+S_2r_2)Q}{2S_1S_2(r_1+r_2)(S_2-S_1)},\\ Q&=\sqrt{[S_1(r_1+r_2+p_2)-S_2(r_1+r_2+p_1)]^2+4S_1S_2p_1p_2}.\end{aligned}\tag{10}$$

- *Case 3*: $S_1>S_2$, by reversibility (Muth 1979, Jacobs 1993)

$$\rho=TP_2(p_2,r_2,S_2,p_1,r_1,S_1,N).\tag{11}$$

In terms of the steady state of the procedure, the throughput of the serial line, $\widehat{\rho}$, is defined through ρ_M^f.

$$\widehat{\rho}=\lim_{l\to\infty}\rho_M^f(l)=\rho_M^f.\tag{12}$$

Remark 3: Numerical experiments conducted by Jacobs (1993) and Li (2004a) using procedure 2 have all resulted in convergent sequences, suggesting that operator Φ_1 is monotonic. In view of this, we introduce the following hypothesis:

Hypothesis 1: *Procedure 2 converges and Operator* $\Phi_1(p_1, r_1, \ldots, p_M, r_M, N_1, \ldots, N_{M-1})$ *decreases monotonically with respect to* p_1*, and increases monotonically with respect to* r_1.

Remark 4: Due to assumptions (vi) and (viii), procedure 2 implies that a blocking before service time-dependent failure model is used in this study. Compared to blocking after service operation-dependent assumption in some simulation software packages, the difference in system throughput is typically small when $N > 0$ (Buzacott and Shanthikumar (1993)). However, when $N = 0$, the difference becomes larger when the number of machines in the system increases. To correct this discrepancy, based on our experience and experimental study, by setting $N = 0.2$ (or a similar small number $0 < N \ll 1$), the difference will be negligible. □

3.4 *Convergence*

By using the results from procedure 2, we obtain the following.

Proposition 1: *Under Hypothesis 1, recursive procedure 1 is convergent, i.e.*

$$\lim_{n\to\infty} s_k(n) = s_k, \quad \lim_{n\to\infty} s_{k+1,j}(n) = s_{k+1,j}, \quad j = a, c,$$
$$\lim_{n\to\infty} b_{k+1}(n) = b_{k+1}, \quad \lim_{n\to\infty} b_{k,j}(n) = b_{k,j}, \quad j = a, c,$$
$$\lim_{n\to\infty} \widehat{\rho}_i(n) = \widehat{\rho}_i \quad i = 1, \ldots, 4. \tag{13}$$

Proof: See Appendix. □

From conservation of flow, it follows immediately that

Corollary 1: *Under assumptions* (i–viii),

$$\widehat{\rho}_1 = \widehat{\rho}_2 = \widehat{\rho}_3 + \widehat{\rho}_4. \tag{14}$$

Using the limits in (13), the system throughput can be evaluated as:

$$\widehat{\rho} = TP(p_{k+1} + r_{k+1}s_{k+1,a}s_{k+1,c}, r_{k+1}(1 - s_{k+1,a}s_{k+1,c}), S_{k+1}, \ldots, p_M, r_M, S_M, N_{k+1}, \ldots, N_{M-1}). \tag{15}$$

3.5 *Accuracy*

In this paper, the accuracy of the estimation is investigated numerically. Dozens of systems defined by the assumptions (i–viii) with various machine and buffer parameter settings are simulated by using the commercial simulation software Simul8 (http://www.simul8.com). Eight of the examples are shown in table 1. In all the experiments we carried out, zero initial occupancy of all buffers and 5000 time units of warm up period have been assumed. The next 50 000 time units of stationary regime have been used to evaluate throughput statistically. The 95% confidence intervals for all statistical estimates have been evaluated with 20 runs. The confidence intervals throughout this paper are consistently less than ±0.0020. In table 1, ρ denotes the throughput obtained by simulation and $\widehat{\rho}$ is the estimate using procedure 1. It is shown in table 1 that the estimates return in an acceptable precision.

Table 1. Accuracy of throughput estimation (err% $=(|\rho - \widehat{\rho}|)/(\rho) \cdot 100\%$).

	Machine and buffer parameter	Upstream common line, m_1–m_k	Downstream common line, m_{k+1}–m_M	Product A dedicated line, $m_{a,i}$	Product C dedicated line, $m_{c,i}$	Split probability and throughput and accuracy
Ex 1	p_i	0.1 0.1 0.1	0.1 0.1 0.1	0.1 0.1	0.1	γ: 0.26
	r_i	0.8 0.8 0.8	0.8 0.8 0.8	0.8 0.8	0.8	ρ: 0.8235
	S_i	1 1 1	1 1 1	1 1	1	$\widehat{\rho}$: 0.8220
	N_i	3 3	3 3	3 3 3	3 3	err%: 0.18
Ex 2	p_i	0.01 0.01 0.02	0.02 0.01 0.01	0.02	0.03	γ: 0.32
	r_i	0.6 0.6 0.55	0.55 0.6 0.6	0.6	0.6	ρ: 0.9584
	S_i	1 1 1	1 1 1	1	1	$\widehat{\rho}$: 0.9327
	N_i	3 3	3 3	2 2	3 3	err%: 2.68
Ex 3	p_i	0.2 0.22 0.25 0.1	0.23 0.24	0.1 0.15	0.14 0.18 0.2	γ: 0.15
	r_i	0.83 0.86 0.85 0.94	0.86 0.84	0.93 0.95	0.87 0.95 0.9	ρ: 0.7285
	S_i	1 1 1 1	1 1	1 1	1 1 1	$\widehat{\rho}$: 0.7173
	N_i	2 2 3	3	2 2 3	4 3 3 2	err%: 1.54
Ex 4	p_i	0.13 0.1	0.09 0.11	0.15 0.12	0.05 0.04	γ: 0.6
	r_i	0.63 0.5	0.49 0.59	0.45 0.5	0.35 0.33	ρ: 0.9116
	S_i	1.6 1.8	1.7 1.5	1.1 0.8	1.2 0.9	$\widehat{\rho}$: 0.8972
	N_i	4	3	3 3 2	4 2 3	err%: 1.58
Ex 5	p_i	0.003 0.04	0.06 0.05	0.14 0.17 0.18	0.1 0.12	γ: 0.7
	r_i	0.8 0.9	0.77 0.84	0.65 0.79 0.68	0.6 0.7	ρ: 0.9088
	S_i	1.3 1.2	1.4 1.5	1.9 1.1	0.9 1.1	$\widehat{\rho}$: 0.9300
	N_i	3	3	2 2 2 2	2 2 2	err%: 2.33
Ex 6	p_i	0.02 0.03	0.04 0.05	0.06 0.07 0.08	0.09 0.11 0.13	γ: 0.45
	r_i	0.8 0.7	0.6 0.5	0.4 0.3 0.2	0.4 0.3 0.2	ρ: 0.9031
	S_i	1.1 1.2	1.6 1.7	1.3 1.4 1.5	1.3 1.4 1.5	$\widehat{\rho}$: 0.9287
	N_i	2	2	3 2 2 3	3 2 2 3	err%: 2.83
Ex 7	p_i	0.05 0.08 0.08	0.06 0.07	0.15 0.13 0.17	0.15 0.12 0.14 0.19	γ: 0.25
	r_i	0.6 0.72 0.65	0.63 0.69	0.57 0.64 0.52	0.66 0.49 0.65 0.68	ρ: 0.7194
	S_i	1.5 1.35 1.2	1.2 1.6	0.85 0.79 0.95	0.8 0.87 0.87 0.99	$\widehat{\rho}$: 0.7283
	N_i	2 3	2	1 3 2 3	3 2 1 2 2	err%: 1.24
Ex 8	p_i	0.1 0.13 0.08	0.11 0.05	0.06 0.09	0.25	γ: 0.8
	r_i	0.5 0.62 0.48	0.65 0.66	0.44 0.45	0.78	ρ: 0.7305
	S_i	1.1 1.25 1	1.15 1.3	1.1 0.95	1.05	$\widehat{\rho}$: 0.7370
	N_i	3 2	2	2 3 3	3 2	err%: 0.89

Remark 5: The calculation of estimate (15) is much faster than simulation. All examples analysed in table 1 using procedure 1 are carried out in less than 1 s on a 600-MHz PC. To compare, simulation takes at least 6 min.

4. Case study

4.1 *Throughput estimation*

The above method has been used for analysis of multiple-product manufacturing systems with split and merge. A case study is described below. Figure 4 illustrates a simplified layout of a body assembly system design (with shortened main line).

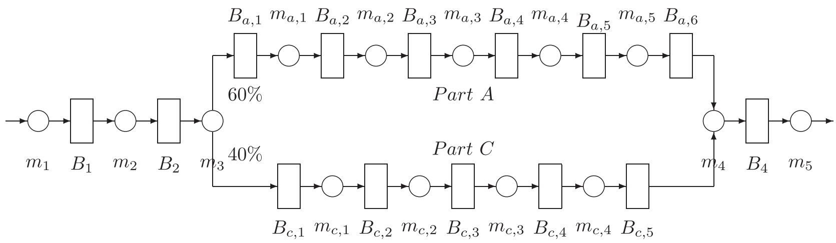

Figure 4. Simplified system layout with shortened shared lines.

Table 2. Design parameters.

m_1–m_5	p_i	0.0023 0.0020 0.0020 0.0537 0.0169	r_i	50.000 0.3636 0.3571 3.3333 0.4348
	S_i	1.2833 1.5333 1.5000 3.1167 1.4000	N_i	2 2, –, 1
$m_{a,1}$–$m_{a,5}$	$p_{a,i}$	0.0045 0.0070 0.0010 0.0083 0.0187	$r_{a,i}$	0.8197 1.3158 0.1754 0.3333 2.0000
	$S_{a,i}$	2.0500 3.1667 1.1000 1.2000 1.2167	$N_{a,i}$	1 1 1 1 1 4
$m_{c,1}$–$m_{c,4}$	$p_{c,i}$	0.0059 0.0131 0.0082 0.0055	$r_{c,i}$	0.8197 0.3704 0.3448 0.5000
	$S_{c,i}$	2.6667 1.0500 1.1833 0.9000	$N_{c,i}$	3 1 1 1 4

Table 3. Sensitivity to machine downtime.

Downtime increment	5%	10%	15%	20%	25%	50%	100%
ρ jobs/(unit of time)	1.2781	1.2775	1.2770	1.2763	1.2756	1.2682	1.2170

The designed system parameters are shown in table 2. They are selected based on the data provided by manufacturers, collected data from similar equipment in existing systems, or the experimental results of new equipment, according to cost, capacity and other requirements.

In addition, 60% of the products are part type A, the rest are part type C. Due to confidentiality, the layout and data have been modified, they are for illustration purposes only.

Using procedure 1, the calculated system throughput is 1.2785 parts/(unit of time). We also use Simul8 to validate the design. The simulated throughput is 1.2628 parts/(unit of time). The difference is 1.24%. Therefore, the model is justified in this case study and can be used for further analysis.

4.2 *Sensitivity study*

Due to the uncertainty of the data used in the design phase, system parameters (i.e., speed, mean time to repair, mean time between failure, etc.) during actual operations usually will deviate from design parameters. To ensure an effective production, a sensitivity study on system parameters is needed and should be an important part of any system design. Below, sensitivity studies with respect to machine downtime, uptime, speed and buffer capacity is introduced and shown in tables 3–7, respectively.

Table 4. Sensitivity to machine uptime.

Uptime decrement	5%	10%	20%	25%	50%	60%	75%
ρ jobs/(unit of time)	1.2783	1.2780	1.2773	1.2769	1.2731	1.2637	1.2010

Table 5. Sensitivity to machine speed.

Speed decrement	1%	2%	3%	5%
ρ jobs/(unit of time)	1.2662	1.2531	1.2417	1.2181

Table 6. Individual machine sensitivity to speed.

	m_1	m_2	m_3	m_4	m_5	$m_{a,1}$	$m_{a,2}$
$(\rho(S_i-\delta)-\rho(S_i))/\delta$	0.992	0.001	0	0	0	0	0
	$m_{a,3}$	$m_{a,4}$	$m_{a,5}$	$m_{c,1}$	$m_{c,2}$	$m_{c,3}$	$m_{c,4}$
$(\rho(S_i-\delta)-\rho(S_i))/\delta$	0	0	0	0	0	0	0

Table 7. Sensitivity to buffer capacity.

	$N_1=1$	$N_2=1$	$N_{a,6}=1$	$N_{c,1}=1$	$N_{c,5}=2$	$N_{c,5}=1$	$N_i=1, \forall i$
ρ jobs/(unit of time)	1.2758	1.2785	1.2785	1.2785	1.2785	1.2754	1.2653

4.2.1 Sensitivity to machine downtime. In table 3, all machine downtimes are increased by 5–100%. The system throughput is evaluated using procedure 1.

From table 3, it is clear that the designed system, relatively, is not sensitive to machine downtime. Even when machine downtimes are increased over 50%, the decrease on throughput is negligible.

4.2.2 Sensitivity to machine uptime. Similar study is conducted for sensitivity with respect to machine uptime. For this study, in table 4, all machine uptimes are decreased by 5–75%. It is shown that the system is also not sensitive to machine uptime. A 60% decrease in machine uptime only has minimum effect on system throughput.

4.2.3 Sensitivity to machine speed. In table 5, all machine speeds are decreased by 1, 2, 3 and 5%. Obviously, although the current design is not sensitive to machine uptime or downtime, it is somewhat sensitive to machine speed.

Since the system is relatively sensitive to machine speed, we investigate next which machine is the most sensitive one. In other words, we need to find out machine i such that

$$\frac{\rho(S_i-\delta)-\rho(S_i)}{\delta} > \frac{\rho(S_j-\delta)-\rho(S_j)}{\delta}, \quad \forall j \neq i, 0 < \delta \ll 1.$$

In Chiang (2001), such machine is referred to as *c*-bottleneck machine. It is shown in table 6 that machine m_1 is the most sensitive machine. Therefore, it is important to verify that all machines operate at the designed speeds, in particular, machine m_1.

4.2.4 Sensitivity to buffer capacity. Finally, sensitivity study with respect to buffer capacity is carried out. Various buffer capacity settings are assigned in table 7. It turns out that reducing buffer capacities will not affect system throughput dramatically. Therefore, more improvement in reducing inventory cost can be obtained in this direction.

5. Conclusions

A system-theoretic approach, referred to as overlapping decomposition, was presented to model and analyse the performance of multiple-product manufacturing systems. The convergence of the iteration procedures was proved analytically and the accuracy of the estimates was evaluated numerically with satisfactory precision. A case study for analysis of a two-product body assembly system design was introduced to illustrate the applicability of the method. Note that the method described is not only applicable to the automotive industry, but also can be used in other manufacturing industries.

In addition to analysing the multiple production system described, the study of fully flexible production lines where each machine is capable of processing different products (in some cases, it is referred to as product or model mix lines) is an important and challenging problem, in particular when many production systems are becoming more flexible. This will be one of the directions in future work.

Appendix: Proofs of subsection 3.2

To prove Proposition 1, the following two Lemmas are needed:

Lemma A.1: *Under Hypothesis 1, operator* $\Phi_2(p_1, r_1, \ldots, p_M, r_M, N_1, \ldots, N_{M-1})$ *is monotonically decreasing with respect to* p_M*, and increasing with respect to* r_M.

Proof: Follows from Hypothesis 1 by reversibility of production lines (Muth 1979, Jacobs 1993). □

Lemma A.2: *Under Hypothesis 1, in recursive procedure 1, if* $s_{k+1,j}(n) > s_{k+1,j}(n-1)$ *and* $b_{k,j}(n) < b_{k,j}(n-1), j = a, c$*, then* $s_{k+1,j}(n+1) > s_{k+1,j}(n)$ *and* $b_{k,j}(n+1) < b_{k,j}(n)$.

Proof: If

$$s_{k+1,j}(n) > s_{k+1,j}(n-1), \quad b_{k,j}(n) < b_{k,j}(n-1), \quad j = a, c, \tag{A.1}$$

from (2), we have

$$r_k^1(n+1) > r_k^1(n), \quad p_k^1(n+1) < p_k^1(n).$$

It follows from Lemma A.1 that

$$s_k(n+1) > s_k(n). \tag{A.2}$$

Similarly, from (1) we have

$$r_{k+1}^2(n+1) < r_{k+1}^2(n), \quad p_{k+1}^2(n+1) > p_{k+1}^2(n).$$

From Hypothesis 1,

$$b_{k+1}(n+1) < b_{k+1}(n). \tag{A.3}$$

For line 3, it implies that

$$\begin{aligned} r_k^3(n+1) &= r_k(1-\gamma)[1 - s_k(n+1)] \\ &< r_k(1-\gamma)[1 - s_k(n)] = r_k^3(n), \\ p_k^3(n+1) &> p_k^3(n), \\ r_{k+1}^3(n+1) &= 0.5r_{k+1}[1 + s_{k+1,c}(n)][1 - b_{k+1}(n+1)] \\ &> 0.5r_{k+1}[1 + s_{k+1,c}(n-1)][1 - b_{k+1}(n)] = r_{k+1}^3(n), \\ p_{k+1}^3(n+1) &< p_{k+1}^3(n). \end{aligned}$$

Then from Hypothesis 1 and Lemma A.1, we have

$$b_{k,a}(n+1) < b_{k,a}(n), \quad s_{k+1,a}(n+1) > s_{k+1,a}(n).$$

Similarly for line 4, we obtain

$$\begin{aligned} r_k^4(n+1) &= r_k\gamma[1 - s_k(n+1)] \\ &< r_k\gamma[1 - s_k(n)] = r_k^4(n), \\ p_k^4(n+1) &> p_k^4(n), \\ r_{k+1}^4(n+1) &= 0.5r_{k+1}[1 + s_{k+1,a}(n)][1 - b_{k+1}(n+1)] \\ &> 0.5r_{k+1}[1 + s_{k+1,a}(n-1)][1 - b_{k+1}(n)] \\ &= r_{k+1}^4(n), \\ p_{k+1}^4(n+1) &< p_{k+1}^4(n), \end{aligned}$$

and

$$b_{k,c}(n+1) < b_{k,c}(n), \quad s_{k+1,c}(n+1) > s_{k+1,c}(n).$$

□

Proof of Proposition 1: By induction. For $n = 0$,

$$s_{k+1,j}(0) = 0, \quad b_{k,j}(0) = 1, \quad j = a, c.$$

For line 2, we have

$$r_{k+1}^2(1) = r_{k+1}, \quad p_{k+1}^2(1) = p_{k+1}.$$

Then,

$$0 < b_{k+1}(1) < 1 = b_{k+1}(0).$$

Similarly for line 1, we have

$$r_k^1(1) = 0, \quad s_k(1) = 0.$$

For line 3, we obtain

$$
\begin{aligned}
r_k^3(1) &= r_k(1-\gamma), \\
p_k^3(1) &= p_k + r_k\gamma, \\
r_{k+1}^3(1) &= 0.5r_{k+1}[1 + s_{k+1,c}(0)][1 - b_{k+1}(1)] \\
&= 0.5r_{k+1}[1 - b_{k+1}(1)] > 0, \\
p_{k+1}^3(1) &> 0.
\end{aligned}
$$

Therefore,

$$0 < b_{k,a}(1) < 1, \quad 1 > s_{k+1,a} > 0.$$

Analogously for line 4, it leads to

$$
\begin{aligned}
r_k^4(1) &= r_k\gamma, \\
p_k^4(1) &= p_k + r_k(1-\gamma), \\
r_{k+1}^4(1) &= 0.5r_{k+1}[1 + s_{k+1,a}(1)][1 - b_{k+1}(1)] \\
&> 0, \\
p_{k+1}^4(1) &> 0.
\end{aligned}
$$

Therefore,

$$0 < b_{k,c}(1) < 1, \quad 1 > s_{k+1,c}(1) > 0.$$

Continue this process for $n = 1$, we obtain

$$r_{k+1}^2(2) < r_{k+1} = r_{k+1}(1), \quad r_k^1(2) > 0 = r_k^1(1),$$

it follows from Hypothesis 1 and Lemma A.1 that

$$b_{k+1}(2) < b_{k+1}, \quad s_k(2) > 0 = s_k(1).$$

In addition,

$$
\begin{aligned}
r_k^3(2) &= r_k(1-\gamma)[1 - s_k(2)] \\
&< r_k(1-\gamma) = r_k^3(2), \\
r_{k+1}^3(2) &= 0.5r_{k+1}[1 + s_{k+1,c}(1)][1 - b_{k+1}(2)] \\
&> 0.5r_{k+1}[1 - b_{k+1}(1)] = r_{k+1}^3(1).
\end{aligned}
$$

It implies that

$$b_{k,a}(2) < b_{k,a}(1), \quad s_{k+1,a}(2) > s_{k+1,a}(1).$$

Moreover,

$$
\begin{aligned}
r_k^4(2) &= r_k\gamma[1 - s_k(2)] \\
&< r_k\gamma = r_k^4(1), \\
r_{k+1}^4(2) &= 0.5r_{k+1}[1 + s_{k+1,a}(2)][1 - b_{k+1}(2)] \\
&> 0.5r_{k+1}[1 - b_{k+1}(1)] = r_{k+1}^4(1),
\end{aligned}
$$

and

$$b_{k,c}(2) < b_{k,c}(1), \quad s_{k+1,c}(2) > s_{k+1,c}(1).$$

The base case is proved. Assume now $n > 0$,

$$s_{k+1,j}(n) > s_{k+1,j}(n-1), \quad b_{k,j}(n) < b_{k,j}(n-1), \quad j = a, c.$$

Then from Lemma A.2 we obtain

$$s_{k+1,j}(n+1) > s_{k+1,j}(n), \quad b_{k,j}(n+1) < b_{k,j}(n), \quad j = a, c.$$

Therefore, $s_{k+1,j}(n)$ and $b_{k,j}(n)$, $j = a, c$, are monotonically increasing or decreasing, respectively. Since they are bounded by 0 and 1 (Chiang *et al*. 2000a), they are convergent. □

Acknowledgement

The authors thank Dr Dennis E. Blumenfeld and Dr Samuel P. Marin of General Motors Research & Development Center, and the anonymous reviewers for valuable comments and suggestions.

References

Buzacott, J.A. and Shantikumar, J.G., *Stochastic Models of Manufacturing Systems*, 1993 (Prentice Hall: Englewood Cliffs).

Chiang, S.-Y., Kuo, C.-T. and Meerkov, S.M., DT-bottlenecks in serial production lines: theory and application. *IEEE Trans. Robot. Automat*., 2000a, **16**, 567–580.

Chiang, S.-Y., Kuo, C.-T., Lim, J.-T. and Meerkov, S.M., Improvability theory of assembly systems I: problem formulation and performance evaluation. *Math. Prob. Eng*., 2000b, **6**, 321–357.

Chiang, S.-Y., Kuo, C.-T. and Meerkov, S.M., c-Bottlenecks in serial production lines: identification and application. *Math. Prob. Eng*., 2001, **6**, 543–578.

Colledani, M., Matta, A. and Tolio, T., Performance evaluation of production lines with finite buffer capacity producing two different products, in *Proceedings of the 4th Aegean Internation Conference on Analysis of Manufacturing Systems*, Samos Island, Greece, 2003, pp. 231–240.

Dallery, Y., Extending the scope of analytical methods for performance evaluation of manufacturing flow systems, in *2nd Internationl Conference on Analysis and Modeling of Manufacturing Systems*, Tinos Island, Greece, 1999.

Dallery, Y. and Gershwin, S.B., Manufacturing flow line systems: a review of models and analytical results. *Queueing Syst*., 1992, **12**, 3–94.

Dallery, Y., David, R. and Xie, X.-L., An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers. *IIE Transactions*, 1988, **20**, 280–283.

Dallery, Y., David, R. and Xie, X.-L., Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Trans. Automat. Contr*., 1989, **34**, 943–953.

Diamantidis, A.C. and Papdopoulos, C.T., Analysis of a discrete material unreliable merge system with a finite intermediate buffer and three stations with operational-dependent and idleness failures, in *Proceedings of the 4th Aegean Internation Conference on Analysis of Manufacturing Systems*, Samos Island, Greece, 2003, pp. 103–112.

Di Mascolo, M., David, R. and Dallery, Y., Modeling and analysis of assembly systems with unreliable machines and finite buffers. *IIE Trans*., 1991, **23**, 315–330.

Gershwin, S.B., *Manufacturing Systems Engineering*, 1994 (Prentice Hall: Englewood Cliffs).

Inman, R.R., Empirical evaluation of exponential and independence assumptions in queueing models of manufacturing systems. *Prod. Oper. Manag*. 1999, **4**, 409–432.

Jacobs, D.A., *Improvability in production systems: theory and case studies*. PhD thesis, Department of EECS, University of Michigan, Ann Arbor, 1993.

Jacobs, D.A. and Meerkov, S.M., A system-theoretic property of serial production lines: improvability. *Int. J. Sys. Sci.* 1995, **26**, 95–137.

Li, J., Modeling and analysis of complex production systems, in *Proceedings of the fourth Aegean International Conference on Analysis of Manufacturing Systems*, Samos Island, Greece, 2003, pp. 203–212.

Li, J., Throughput analysis in automotive paint shops: a case study. *IEEE Trans. Automat. Sci. Eng.*, 2004a, **1**, 90–98.

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

Li, J., Modeling and analysis of manufacturing systems with parallel lines. *IEEE Trans. Automat. Contr.*, 2004c, **49**, pp. 1824–1829.

Muth, E., The reversibility property of production lines. *Manag. Sci.*, 1979, **25**, 152–158.

Nemec, J.E., Diffusion and decomposition approximations of stochastic models of multiclass processing networks. PhD thesis, Sloan School of Management, MIT, Cambridge, MA, 1999.

Syrowicz, D., Decomposition analysis of a deterministic, multiple-part-type, multiple-failure-mode production lines. SM thesis, Department of EECS, MIT, Cambridge, MA, 1999.