Modeling and Analysis of Manufacturing Systems With Parallel Lines

Jingshan Li

***Abstract*—This note is devoted to the study of manufacturing systems with parallel production lines. By using overlapping decomposition method, a convergent recursive procedure is presented to estimate system production rate with high accuracy. The results obtained are illustrated by a case study at an automotive assembly plant.**

***Index Terms*—Aggregation, overlapping decomposition, parallel lines, production rate.**

I. Introduction

Parallel lines are often used to increase production capacity in many manufacturing systems. A production line can be split into several lines in parallel, each of them consisting of one or more machines to process identical or different jobs. At the end, all parallel lines are merged into one single line. To design, operate and improve the performance of such systems, accurate performance analysis is necessary and important.

Analytical methods to evaluate the production rate (PR, i.e., number of parts produced by the last machine per unit of time) of parallel systems are limited. References [1]–[6] study the series-parallel type of production lines where each stage consists of multiple parallel machines. The multiple machines are replaced by an equivalent single machine and system performance is analyzed using the equivalent machine. A similar idea has been used in [7] for assembly/disassembly systems. In all these studies, each parallel line only consists of one machine. However, it is common in most parallel systems that there are multiple machines in each parallel line. Except for simulation study, there are no analytical methods available in the current literature to analyze the performance of such systems. Therefore, an effective method to study the parallel systems with multiple machines in each line is needed. In this note, we present a recursive method, which decomposes the parallel system into overlapped serial lines, to approximate the performance of such systems. We prove the convergence of the procedure and uniqueness of the solution analytically and justify the accuracy numerically.

The remainder of this note is structured as follows: Section II formulates the problem. A recursive procedure to estimate parallel system performance is presented in Section III. Section IV introduces a case study at an automotive assembly plant. The conclusions are formulated in Section V. Due to page limitation, all proofs are omitted and can be found in [8].

II. Problem Formulation

The parallel manufacturing system studied in this note is shown in Fig. 1. The circles represent the machines and the rectangles are the buffers. The following assumptions pertaining to machines, buffers, and their interactions are introduced in this note.

i) The system consists of a main line (machines m_1, m_2, buffers B_1, B_2) and k parallel lines (machines $m_{i,1}, \dots, m_{i,M_i}$, $i = 1, \dots, k$, buffers $B_{i,1}, \dots, B_{i,M_i-1}$). The main line splits into k parallel lines at B_1, while the parallel lines merge into the main line at B_2.

ii) All machines, m_1, m_2, and $m_{i,j}$, $i = 1, \dots, k$, $j = 1, \dots, M_i$, have two states: up and down. When up, machines m_1, m_2, and $m_{i,j}$ are capable of producing with the rate S_1, S_2 or $S_{i,j}$ parts per unit of time, respectively; when the machine is down, no production takes place. For simplicity, it is assumed that all machines in parallel line i, $i = 1, \dots, k$, have identical speed, i.e., $S_{i,j_1} = S_{i,j_2}, \forall j_1, j_2 \in \{1, \dots, M_i\}$.

iii) The up- and the downtime of each machine m_j, $j = 1, 2$, or $m_{i,j}$, $i = 1, \dots, k$, $j = 1, \dots, M_i$, are random variables distributed exponentially with parameters p_j and r_j, or $p_{i,j}$ and $r_{i,j}$, respectively.

Remark 1: Assumption iii) implies that p_j and r_j (respectively, $p_{i,j}$ and $r_{i,j}$) are failure and repair rates, respectively. In other words, $1/p_j$ and $1/r_j$ (respectively, $1/p_{i,j}$ and $1/r_{i,j}$) are average up- and downtime of m_j (respectively, $m_{i,j}$), respectively.

iv) Each buffer B_j, $j = 1, 2$, or $B_{i,j}$, $i = 1, \dots, k$, $j = 1, \dots, M_i - 1$, has a finite capacity N_j, $0 \le N_j < \infty$, or $N_{i,j}$, $0 \le N_{i,j} < \infty$, respectively.

v) A machine is starved at time t if its upstream buffer is empty at time t. Machine m_1 is never starved.

vi) A machine is blocked at time t if downstream buffer is full at time t. Machine m_2 is never blocked.

vii) The first machine in each parallel line, $m_{i,1}$, $i = 1, \dots, k$, has equal probability to take the last part (more rigorously, a fraction of part in the continuous model) in buffer B_1, if it is not blocked. Similarly, buffer B_2 receives the last fraction of part (to make it full) from any unstarved machines m_{i,M_i}, $i = 1, \dots, k$, with equal probabilities.

The problem addressed in this note is as follows: *Given the parallel production system i)–vii), develop a method for evaluating the production rate as a function of the system parameters.*

A solution to the problem is given in Section III.

III. Performance Evaluation

A. Idea of Overlapping Decomposition Method

Due to the complexity of parallel systems, exact analysis is intractable. Therefore, an approximation method is needed. The idea of the approximation is to decompose the production system i)–vii) into a set of serial lines, with the first or last machines in one serial line overlapped with another line, and to modify the overlapping machines appropriately to accommodate the effects of other lines. More specifically, we decompose the system into $k + 2$ serial lines (see Fig. 2), where parallel lines 1 to k are referred to as lines 1 to k, respectively. Machine m_1 and the aggregation of all first machines, $m_{i,1}$, in each parallel line construct line $k + 1$. Finally, line $k + 2$ consists of the aggregation of all last machines, m_{i,M_i}, and machine m_2. Since machines $m_{i,1}$ and m_{i,M_i}, $i = 1, \dots, k$, have been used for both line i, and lines $k + 1$ or $k + 2$, respectively, this method involves overlapping and is referred to as *overlapping decomposition* [9], [10]. In the following, the procedure for implementing the method is outlined.

Consider parallel line i, $i = 1, \dots, k$, consisting of machines $m_{i1}, \dots, m_{i,M_i}$, and buffers $B_{i,1}, \dots, B_{i,M_i-1}$. If we know the production rate of each line i, the PR of the whole system is readily obtained. A convergent recursive procedure for evaluating PR of a serial line has been developed in [9] and is outlined in Section III-C.

Manuscript received September 27, 2002; revised April 25, 2003, April 8, 2004, and May 27, 2004. Recommended by Associate Editor A. Giua.

The author is with the Manufacturing Systems Research Laboratory, General Motors Research and Development Center, Warren, MI 48090-9055 USA (e-mail: jingshan.li@gm.com).

Digital Object Identifier 10.1109/TAC.2004.835584

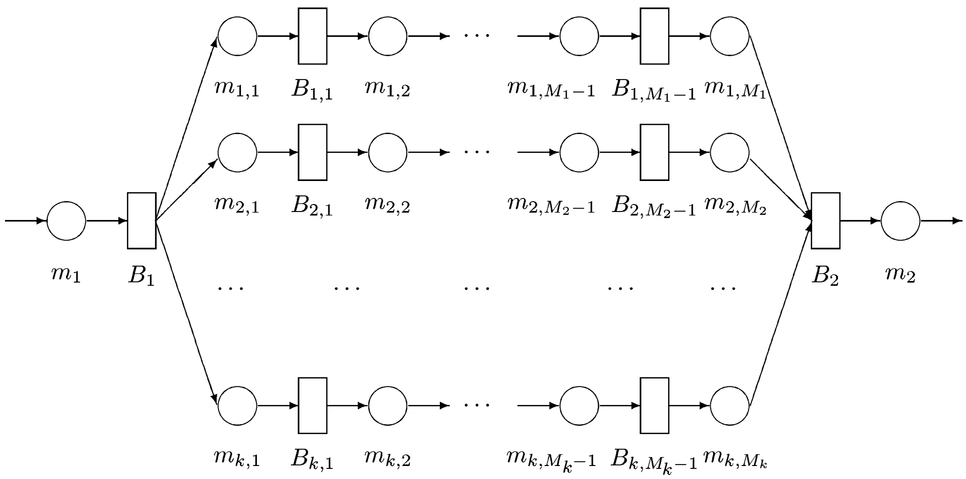

Fig. 1. Parallel production lines.

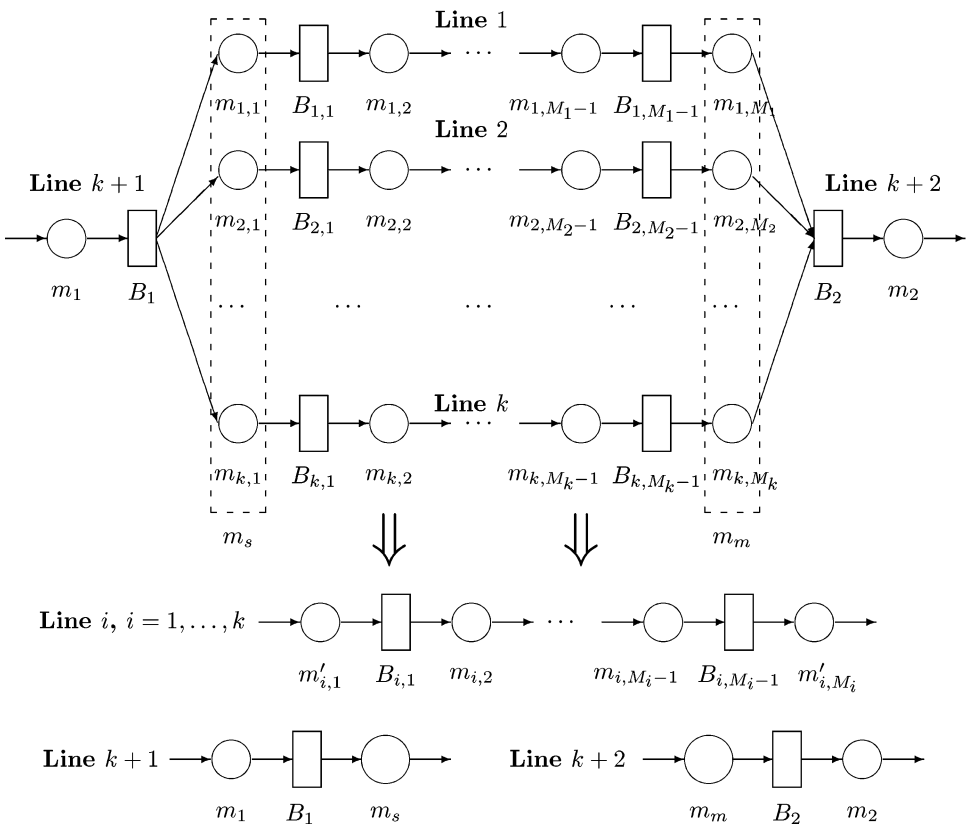

Fig. 2. Overlapping decomposition.

In order to use this procedure, $m_{i,1}$ and m_{i,M_i} should not be starved or blocked, respectively. Therefore, we introduce fictitious machines $m'_{i,1}$ and m'_{i,M_i} defined by parameters $p'_{i,1}$, $r'_{i,1}$, and p'_{i,M_i}, r'_{i,M_i}, $i = 1, \dots, k$, respectively, where

$$\begin{aligned} r'_{i,1} &= r_{i,1}\mathrm{Prob}\{m_{i,1} \text{ is not starved}\} \\ p'_{i,1} &= p_{i,1} + r_{i,1} - r'_{i,1} \\ r'_{i,M_i} &= r_{i,M_i}\mathrm{Prob}\{m_{i,M_i} \text{ is not blocked}\} \\ p'_{i,M_i} &= p_{i,M_i} + r_{i,M_i} - r'_{i,M_i}. \end{aligned}$$

Remark 2: Here, $r'_{i,1}$ is adjusted by considering the starving time of machine $m_{i,1}$ as its downtime (from the point of view of the machines downstream of $m_{i,1}$). Therefore, fictitious machine $m'_{i,1}$'s average downtime $1/r'_{i,1}$ can be approximated by $1/(r_{i,1}\mathrm{Prob}\{m_{i,1}$ is not starved$\})$. Based on conservation of flow, $p'_{i,1}$ is selected such that $r'_{i,1}/(p'_{i,1} + r'_{i,1}) = (r_{i,1}/(p_{i,1} + r_{i,1}))\mathrm{Prob}\{m_{i,1}$ is not starved$\}$. Analogously, the blocking time of m_{i,M_i} can also be considered as m_{i,M_i}'s downtime from the point of view of the machines upstream of m_{i,M_i}. Similar modifications are used for all the overlapping machines $m_{i,1}$ and m_{i,M_i}, $i = 1, \dots, k$.

If $\mathrm{Prob}\{m_{i,1}$ is not starved$\}$ and $\mathrm{Prob}\{m_{i,M_i}$ is not blocked$\}$ were known, we could calculate the production rate of parallel line i. Since these probabilities are unknown, we introduce an iterative procedure described as follows.

Step 1) Assume $\text{Prob}\{m_{i,1}$ is not starved$\}$ and $\text{Prob}\{m_{i,M_i}$ is not blocked$\}$ are known. Modify machines $m_{i,1}$ and m_{i,M_i} to $m'_{i,1}$ and m'_{i,M_i}, $i = 1,\ldots,k$, respectively, and calculate $r'_{i,1}$, $p'_{i,1}$ and r'_{i,M_i}, p'_{i,M_i}. Using the serial line evaluation procedure, we calculate $\text{Prob}\{m_{i,1}$ is blocked$\}$ and $\text{Prob}\{m_{i,M_i}$ is starved$\}$, $i = 1,\ldots,k$. These calculations are denoted as operators Φ_1 and Φ_2, respectively, and are described in Section III-C.

Step 2) Using these probabilities, introduce again fictitious machines $m''_{i,1}$, $i = 1,\ldots,k$, where

$$r''_{i,1} = r_{i,1}\text{Prob}\{m_{i,1} \text{ is not blocked}\}$$
$$p''_{i,1} = p_{i,1} + r_{i,1} - r''_{i,1}.$$

Now aggregate the parallel machines $m''_{i,1}$, $i = 1,\ldots,k$, into an equivalent single machine m_s (where s denotes split). The aggregation formula is provided in Section III-B. Next, consider a two-machine line with machines m_1, m_s and buffer B_1. We can calculate the probability that buffer B_1 is empty (i.e., m_s is starved). Due to assumption vii), it is equivalent to the probability that machine $m_{i,1}$, $i = 1,\ldots,k$, is starved.

Step 3) Similarly, introduce again fictitious machines m''_{i,M_i}, $i = 1,\ldots,k$, where

$$r''_{i,M_i} = r_{i,M_i}\text{Prob}\{m_{i,M_i} \text{ is not starved}\}$$
$$p''_{i,M_i} = p_{i,M_i} + r_{i,M_i} - r''_{i,M_i}.$$

Aggregate machines m''_{i,M_i}, $i = 1,\ldots,k$, into another equivalent single machine m_m (where m denotes merge), and construct another two-machine line of machines m_m, m_2 and buffer B_2. Calculate the probability that m_m is blocked (i.e., probability that any m_{i,M_i}, $i = 1,\ldots,k$, is blocked).

Step 4) Using now these probabilities for the second iteration in analysis, modify machines $m_{i,1}$ and m_{i,M_i} again to $m'_{i,1}$ and m'_{i,M_i}, respectively, and repeat the process [back to Step 1)] until it is convergent. As shown below, the iterations are convergent and result in the estimate of system production rate.

B. Aggregation of Multiple Parallel Machines

To adopt the method previously described, an aggregation of multiple parallel machines into one single machine with the same production rate is needed. Existing aggregation approaches are either developed for special cases, or based on heuristics only. In this section, a new aggregation formula which preserves the system PR is derived analytically.

Consider the following multiple machine system with machines $m_1,\ldots,m_k$ in parallel (Fig. 3), where machine m_j, $j = 1,\ldots,k$, is described by exponential uptime, downtime and speed parameters p_j, r_j, and S_j, respectively. To approximate them into an aggregate machine m_a, the following assumptions of m_a are introduced.

a) The uptime and the downtime of the aggregated machine m_a, are random variables distributed exponentially with parameters p_a and r_a, respectively.

b) When machine m_a is up, it is capable of producing with rate S_a parts per unit of time, where $S_a = \sum_{j=1}^{k} S_j$.

c) When machines $m_{i_1},\ldots,m_{i_l}$, $\{i_1,\ldots,i_l\} = I \subset \{1,\ldots,k\}$, are down, other machines m_j, $j \notin I$, $j \in \{1,\ldots,k\}$, are up during the time period T, the multi-machine system produces with rate $\sum_{j=1,j\notin I}^{k} S_j$ parts per unit of time. However, it is assumed that the aggregate machine m_a always produces with rate S_a, but with shorter uptime, i.e., m_a is only up during the interval $(\sum_{j=1,j\notin I}^{k} S_j/\sum_{n=1}^{k} S_n)T$, and m_a is down during the rest of the time (i.e., $(\sum_{j=1,j\in I}^{k} S_j/\sum_{n=1}^{k} S_n)T$) in interval T.

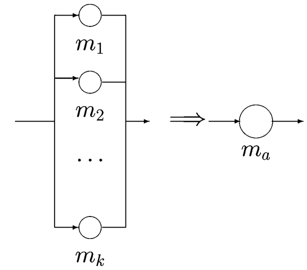

Fig. 3. Aggregation of parallel machines into one single machine.

With the aforementioned assumptions, the aggregated parameters p_a and r_a can be derived.

Lemma 1: Under assumptions a)–c), the aggregated machine m_a can be described by

$$p_a = \frac{S_a \sum_{i=1}^{k}\left(p_i r_i \prod_{j=1,j\neq i}^{k}(p_j + r_j)\right)}{k\sum_{i=1}^{k}\left[S_i r_i \prod_{j=1,j\neq i}^{k}(p_j + r_j)\right]}$$
$$r_a = \frac{S_a \sum_{i=1}^{k}\left(p_i r_i \prod_{j=1,j\neq i}^{k}(p_j + r_j)\right)}{k\sum_{i=1}^{k}\left[S_i p_i \prod_{j=1,j\neq i}^{k}(p_j + r_j)\right]} \quad (1)$$

where $S_a = \sum_{i=1}^{k} S_i$ and, moreover, the system throughput is preserved, i.e.,

$$\frac{S_a r_a}{p_a + r_a} = \sum_{i=1}^{k}\frac{S_i r_i}{p_i + r_i}. \quad (2)$$

Proof: See [8].

Using this aggregation, the recursive procedure is introduced in Section III-D.

C. Operators Φ_1 and Φ_2

Consider a serial line with machines $m_1,\ldots,m_M$, and buffers $B_1,\ldots,B_{M-1}$. Introduce operators Φ_1 and Φ_2 to calculate the probabilities that the first machine is blocked and the last machine is starved in a serial line, respectively.

- Case 1: $M = 2$-machine line

$$\Phi_i = 1 - \frac{PR(p_1, r_1, S_1, p_2, r_2, S_2, N_1)}{\frac{S_i r_i}{r_i + p_i}}, \qquad i = 1, 2, \quad (3)$$

where $PR(p_1, r_1, S_1, p_2, r_2, S_2, N_1)$ is defined by the following (see [11] for details)

— Case 1.1: $S_1 < S_2$

$$PR = \frac{S_2 e_2 A e^{k_1 N_1} + S_1 e_1 B e^{k_2 N_1} + S_1 e_1 C e^{-k_2 N_1}}{A e^{k_1 N_1} + B e^{k_2 N_1} + C e^{-k_2 N_1}}, \quad (4)$$

where

$$A = r_1 D^2 + r_1 D\left[S_1(r_1 + r_2 + p_2) - S_2(r_1 + r_2 + p_1)\right]$$
$$B = r_2 p_1 S_2\left[(S_1 - S_2)(r_1 - r_2) - (S_2 p_1 + S_1 p_2) - D\right]$$

$$C = \frac{e_2(S_2 - S_1 e_1)A + S_1 e_1(1-e_2)B}{S_1 e_1(e_2 - 1)}$$
$$D = \left([S_1(r_1+r_2+p_2) - S_2(r_1+r_2+p_1)]^2 + 4S_1S_2p_1p_2\right)^{\frac{1}{2}}$$
$$k_1 = \frac{1}{2S_1S_2(r_1+r_2)(S_1-S_2)} \times \left[r_1S_1^2(r_1+r_2+p_2) - S_1S_2 \times \left[(r_1+r_2)^2 + (r_1+r_2)(p_1+p_2) + (r_1p_2 + r_2p_1)\right] + r_2S_2^2(r_1+p_1+r_2)\right]$$
$$k_2 = \frac{(S_1r_1 + S_2r_2)D}{2S_1S_2(r_1+r_2)(S_2-S_1)}$$
$$e_i = \frac{r_i}{p_i+r_i}, \qquad i = 1, 2. \tag{5}$$

— Case 1.2: $S_1 > S_2$ (by reversibility)

$$PR = PR(p_2, r_2, S_2, p_1, r_1, S_1, N_1). \tag{6}$$

— Case 1.3: $S_1 = S_2 = S$; see Case 2.

- Case 2: $M \geq 2$-machine line ($S_i = S, \forall i$)

$$\Phi_1 = 1 - \frac{\frac{r_1^b}{p_1^b+r_1^b}}{\frac{r_1}{p_1+r_1}}$$
$$\Phi_2 = 1 - \frac{\frac{r_M^f}{p_M^f+r_M^f}}{\frac{r_M}{p_M+r_M}} \tag{7}$$

where p_1^b, r_1^b, p_M^f and r_M^f are defined through the following procedure (see [9] for details).

Procedure 1:

$$r_i^b(l+1) = r_i - r_iQ\left(p_{i+1}^b(l+1), r_{i+1}^b(l+1), p_i^f(l), r_i^f(l), N_i\right)$$
$$p_i^b(l+1) = p_i + r_i - r_i^b(l+1), \qquad i = 1, \ldots, M-1$$
$$r_i^f(l+1) = r_i - r_iQ\left(p_{i-1}^f(l+1), r_{i-1}^f(l+1) p_i^b(l+1), r_i^b(l+1), N_{i-1}\right)$$
$$p_i^f(l+1) = p_i + r_i - r_i^f(l+1), \qquad i = 2, \ldots, M \tag{8}$$

with boundary conditions

$$p_1^f(l) = p_1 \quad r_1^f(l) = r_1$$
$$p_M^b(l) = p_M \quad r_M^b(l) = r_M, \qquad l = 0, 1, 2, \ldots$$

and initial conditions

$$p_i^f(0) = p_i \quad r_i^f(0) = r_i, \qquad i = 2, \ldots, M-1.$$

Here, function $Q(\cdot)$ (see [12]) is defined by (9), as shown at the bottom of page.

Procedure 1 is similar to the procedure developed in [12], but with different aggregation equations (here, we emphasize on modification of machine downtimes and the detailed modification follows the similar idea introduced in Section III-A). It consists of backward and forward aggregations (denoted by superscripts b and f, respectively), which are similar to [11]–[13]. In the backward aggregation, the last two machines are aggregated in a single machine m_{M-1}^b. Then, m_{M-1}^b is aggregated with m_{M-2} to result in m_{M-2}^b, and so on until all machines are aggregated in m_1^b. In the forward aggregation, m_1 is aggregated with m_2^b to produce m_2^f. Then, m_2^f is aggregated with m_3^b to result in m_3^f, and so on until all machines are aggregated in m_M^f. Then, the process is repeated again. It is shown in [9] that procedure 1 is convergent and has a unique solution. Therefore, steady-state estimates exist, i.e.,

$$\lim_{l\to\infty} p_i^f(l) = p_i^f \quad \lim_{l\to\infty} p_i^b(l) = p_i^b$$
$$\lim_{l\to\infty} r_i^f(l) = r_i^f \quad \lim_{l\to\infty} r_i^b(l) = r_i^b, \qquad i = 1, 2, \ldots, M. \tag{10}$$

The system production rate (of serial line with machines $m_1, \ldots, m_M$) is estimated as

$$PR = PR(p_1, r_1, S, \ldots, p_M, r_M, S, N_1, \ldots, N_{M-1}) = \frac{Sr_1^b}{p_1^b + r_1^b} = \frac{Sr_M^f}{p_M^f + r_M^f}. \tag{11}$$

Using operators Φ_1 and Φ_2, the recursive procedure is introduced next.

D. Recursive Procedure

Let $b_{i,1}, s_{i,1}$ and b_{i,M_i}, s_{i,M_i} denote the probabilities that machines $m_{i,1}$ and m_{i,M_i} are blocked and starved, respectively. Following Steps

$$Q(p_1, r_1, p_2, r_2, N_1) = \begin{cases} \frac{(1-e_1)(1-\phi)}{1-\phi e^{-\beta N_1}} & \text{if } \frac{p_1}{r_1} \neq \frac{p_2}{r_2} \\ \frac{p_1(p_1+p_2)(r_1+r_2)}{(p_1+r_1)[(p_1+p_2)(r_1+r_2)+p_2r_1(p_1+p_2+r_1+r_2)N_1]} & \text{if } \frac{p_1}{r_1} = \frac{p_2}{r_2} \end{cases}$$
$$\beta = \frac{(r_1+r_2+p_1+p_2)(p_1r_2 - p_2r_1)}{(r_1+r_2)(p_1+p_2)}$$
$$\phi = \frac{e_1(1-e_2)}{e_2(1-e_1)}$$
$$e_i = \frac{r_i}{p_i+r_i}, \qquad i = 1, 2. \tag{9}$$

1)–4) in Section III-A, the recursive procedure to analyze parallel system i)–vii) is as follows.

Procedure 2: Parallel line $i, i = 1, \ldots, k$

$$\begin{aligned} r'_{i,1}(n+1) &= r_{i,1}\left[1-s_{i,1}(n)\right] \\ p'_{i,1}(n+1) &= p_{i,1}+r_{i,1}-r'_{i,1}(n+1) \\ r'_{i,M_i}(n+1) &= r_{i,M_i}\left[1-b_{i,M_i}(n)\right] \\ p'_{i,M_i}(n+1) &= p_{i,M_i}+r_{i,M_i}-r'_{i,M_i}(n+1) \\ b_{i,1}(n+1) &= \Phi_1\left(p'_{i,1}(n+1), r'_{i,1}(n+1),\right. \\ &\quad S_{i1}, \ldots, p'_{i,M_i}(n+1), r'_{i,M_i}(n+1), \\ &\quad \left. S_{i,M_i}, N_{i,1}, \ldots, N_{i,M_i-1}\right) \\ s_{i,M_i}(n+1) &= \Phi_2\left(p'_{i,1}(n+1), r'_{i,1}(n+1),\right. \\ &\quad S_{i,1}, \ldots, p'_{i,M_i}(n+1), r'_{i,M_i}(n+1), \\ &\quad \left. S_{i,M_i}, N_{i,1}, \ldots, N_{i,M_i-1}\right). \end{aligned} \tag{12}$$

Split (13), as shown at the bottom of the page.

Merge (14), as shown at the bottom of the page, where

$$S_s = S_m = \sum_{i=1}^{k} S_{i,1} = \sum_{i=1}^{k} S_{i,M_i}$$

with initial conditions

$$\begin{aligned} s_{i,1}(0) &= \Phi_1\left(p_1, r_1, S_1, p_s(0), r_s(0), S_s, N_1\right) \\ b_{i,M_i}(0) &= 0, \qquad i = 1, \ldots, k \end{aligned}$$

and

$$\begin{aligned} p_s(0) &= \frac{S_s \sum_{i=1}^{k}\left[p_{i,1} r_{i,1} \prod_{j=1, j\neq i}^{k} (p_{j,1}+r_{j,1})\right]}{k\sum_{i=1}^{k}\left[S_{i,1} r_{i,1} \prod_{j=1, j\neq i}^{k} (p_{j,1}+r_{j,1})\right]} \\ r_s(0) &= \frac{S_s \sum_{i=1}^{k}\left[p_{i,1} r_{i,1} \prod_{j=1, j\neq i}^{k} (p_{j,1}+r_{j,1})\right]}{k\sum_{i=1}^{k}\left[S_{i,1} p_{i,1} \prod_{j=1, j\neq i}^{k} (p_{j,1}+r_{j,1})\right]}. \end{aligned}$$

E. Convergence and Accuracy

Theorem 1: Under assumptions i)–vii), recursive procedure 2 is convergent and has a unique solution, i.e.,

$$\begin{aligned} \lim_{n\to\infty} p_s(n) &= p_s \qquad \lim_{n\to\infty} r_s(n) = r_s \\ \lim_{n\to\infty} p_m(n) &= p_m \qquad \lim_{n\to\infty} r_m(n) = r_m. \end{aligned} \tag{15}$$

Proof: See [8].

Using the limits in (15), the system production rate can be evaluated as

$$\begin{aligned} \widehat{PR} &= PR(p_m, r_m, S_m, p_2, r_2, S_2, N_2) \\ &= PR(p_1, r_1, S_1, p_s, r_s, S_s, N_1). \end{aligned} \tag{16}$$

The accuracy of estimate (16) is investigated numerically. Dozens of systems defined by assumptions i)–vii) with various machine and buffer parameter settings are simulated. In all experiments, 20 replications are carried out, each with 5000 time units of warm up time and 50 000 time units of simulation time. The 95% confidence intervals are consistently around ±0.001. Most differences between estimates (16) and simulation results are less than 2% with the largest one over 3%. In addition, all computations using procedure 2 and expression (16) are completed within 1 s on a 500-MHZ PC, while simulation takes at least 7 min. Due to page limitations, only four numerical examples are shown in Table I, where PR denotes the production rate obtained through simulation and $\widehat{PR}$ is the production rate calculated using (16). More examples can be found in [8].

Remark 3: In stead of using procedure 1, other serial line analysis methods can also be applied (to obtain operators Φ_1 and Φ_2). However, the convergence of the algorithm and the uniqueness of the solution depend on the convergence of the serial line analysis method and the uniqueness of its solution. Moreover, the current method can also be extended to study the parallel lines with unequal machine speeds. In this case, a procedure to evaluate the PR of nonhomogeneous serial

$$\begin{aligned} r''_{i,1}(n+1) &= r_{i,1}\left[1-b_{i,1}(n+1)\right], \qquad i = 1, \ldots, k \\ p''_{i,1}(n+1) &= p_{i,1}+r_{i,1}-r''_{i,1}(n+1), \qquad i = 1, \ldots, k \\ p_s(n+1) &= \frac{S_s \sum_{i=1}^{k}\left[p''_{i,1}(n+1) r''_{i,1}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,1}(n+1)+r''_{j,1}(n+1)\right)\right]}{k\sum_{i=1}^{k}\left[S_{i,1} r''_{i,1}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,1}(n+1)+r''_{j,1}(n+1)\right)\right]} \\ r_s(n+1) &= \frac{S_s \sum_{i=1}^{k}\left[p''_{i,1}(n+1) r''_{i,1}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,1}(n+1)+r''_{j,1}(n+1)\right)\right]}{k\sum_{i=1}^{k}\left[S_{i,1} p''_{i1}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,1}(n+1)+r''_{j,1}(n+1)\right)\right]} \\ s_{i,1}(n+1) &= \Phi_2\left(p_1, r_1, S_1, p_s(n+1), r_s(n+1), S_s, N_1\right), \qquad i = 1, \ldots, k. \end{aligned} \tag{13}$$

$$\begin{aligned} r''_{i,M_i}(n+1) &= r_{i,M_i}\left[1-s_{i,M_i}(n+1)\right], \qquad i = 1, \ldots, k \\ p''_{i,M_i}(n+1) &= p_{i,M_i}+r_{i,M_i}-r''_{i,M_i}(n+1), \qquad i = 1, \ldots, k \\ p_m(n+1) &= \frac{S_m \sum_{i=1}^{k}\left[p''_{i,M_i}(n+1) r''_{i,M_i}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,M_j}(n+1)+r''_{j,M_j}(n+1)\right)\right]}{k\sum_{i=1}^{k}\left[S_{i,M_i} r''_{i,M_i}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,M_j}(n+1)+r''_{j,M_j}(n+1)\right)\right]} \\ r_m(n+1) &= \frac{S_m \sum_{i=1}^{k}\left[p''_{i,M_i}(n+1) r''_{i,M_i}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,M_j}(n+1)+r''_{j,M_j}(n+1)\right)\right]}{k\sum_{i=1}^{k}\left[S_{i,M_i} p''_{i,M_i}(n+1) \prod_{j=1, j\neq i}^{k} \left(p''_{j,M_j}(n+1)+r''_{j,M_j}(n+1)\right)\right]} \\ b_{i,M_i}(n+1) &= \Phi_1\left(p_m(n+1), r_m(n+1), S_m, p_2, r_2, S_2, N_2\right), \qquad i = 1, \ldots, k \end{aligned} \tag{14}$$

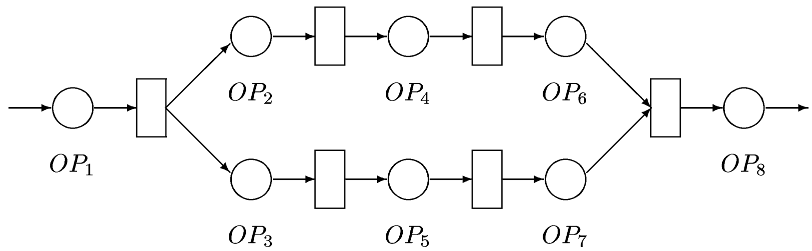

Fig. 4. Simplified layout of a parallel subsystem at an automotive assembly plant.

TABLE I
ACCURACY OF PR ESTIMATION IN PARALLEL SYSTEMS
$(\Delta = ((|PR - \widehat{PR}|)/PR) \cdot 100\%)$

		p	r	S	N
Ex.1	m_1	0.10	0.60	1.00	3
	m_2	0.10	0.60	1.00	3
	$m_{i,j}$	0.10 0.10 0.10	0.60 0.60 0.60	0.50	2 2
		0.10 0.10 0.10	0.60 0.60 0.60	0.50	2 2
		PR: 0.7459	$\widehat{PR}$: 0.7385	Δ: 0.99%	
Ex.2	m_1	0.15	0.69	1.00	5
	m_2	0.08	0.56	1.20	4
	$m_{i,j}$	0.15 0.08 0.13	0.64 0.76 0.66	0.52	3 2
		0.14 0.12 0.17	0.66 0.63 0.69	0.55	2 4
		PR: 0.7823	$\widehat{PR}$: 0.7729	Δ: 1.20%	
Ex.3	m_1	0.09	0.57	1.10	5
	m_2	0.07	0.63	0.80	3
	$m_{i,j}$	0.13 0.06 0.11	0.72 0.54 0.61	0.40	4 2
		0.05 0.25	0.43 0.94	0.65	3
		0.16	0.80	0.15	
		PR: 0.7197	$\widehat{PR}$: 0.7046	Δ: 2.10%	
Ex.4	m_1	0.08	0.64	1.10	5
	m_2	0.05	0.73	1.20	5
	$m_{i,j}$	0.04 0.06 0.07	0.77 0.57 0.69	0.15	3 3
		0.05 0.20	0.51 0.90	0.20	3
		0.15	0.68	0.20	
		0.35 0.40	0.45 0.50	0.50	4
		0.17 0.09 0.12	0.68 0.44 0.55	0.30	3 2
		PR: 0.9100	$\widehat{PR}$: 0.8769	Δ: 3.64%	

TABLE II
IDENTIFIED PARAMETERS

	OP_1	OP_2	OP_3	OP_4	OP_5	OP_6	OP_7	OP_8
p_i	0.13	0.09	0.02	0.07	0.08	0.08	0.09	0.03
r_i	0.88	0.37	0.30	0.30	0.31	0.34	0.34	0.45
S_i	1.26	0.52	0.52	0.52	0.56	0.56	0.56	1.04
N_i	13	4	4	5	6	14		

lines should be used (to replace procedure 1). Again, the convergence of procedure 2 will be dependent on the convergence of PR evaluation method for nonhomogeneous serial lines.

IV. APPLICATION: A CASE STUDY

The aforementioned method has been applied at an automotive assembly plant where a parallel subsystem is analyzed. A simplified system layout is shown in Fig. 4. Based on past system operations, the machine and buffer parameters are identified and presented in Table II. (Note that due to confidentiality reasons, the data illustrated in Table II have been modified appropriately. They are used for demonstration only.)

Using procedure 2, the system production rate is calculated and $\widehat{PR} = 0.81$ part per unit of time, while the actual system operates at $PR = 0.82$ part per unit of time. The difference is only 1.66%. Therefore, the model has acceptable accuracy in this case study.

V. CONCLUSION

This note presents an aggregation procedure using overlapping decomposition method to analyze the performance of parallel systems. The convergence of the procedure and the uniqueness of the solution are proved analytically and the accuracy is justified numerically by comparing with both simulation and actual operations. It is shown that the method results in high accuracy. This method provides a foundation for future work, such as optimal production line design and continuous improvement of existing operations.

ACKNOWLEDGMENT

The author would like to thank Dr. J. M. Alden and Dr. D. E. Blumenfeld of General Motors, as well as the anonymous reviewers, for their helpful comments and suggestions.

REFERENCES

[1] E. Ignall and A. Silver, "The output of a two-stage system with unreliable machines and limited storage," *AIIE Trans.*, vol. 9, pp. 183–188, 1977.
[2] T. Iyama and S. Ito, "The maximum production rate for an unbalanced multi-server flow line system with finite buffer storage," *Int. J. Prod. Res.*, vol. 25, pp. 1157–1179, 1987.
[3] D. Mitra, "Stochastic theory of a fluid model of producers and consumers coupled by a buffer," *Adv. Appl. Probab.*, vol. 20, pp. 646–676, 1988.
[4] R. Alves, "Performance evaluation of series-parallel systems," *J. Manufact. Oper. Manage.*, vol. 3, pp. 224–250, 1990.
[5] M. H. Burman, "New results in flow line analysis," Ph.D. dissertation, MIT, Cambridge, MA, 1995.
[6] A. Patchong and D. Willaeys, "Modeling and analysis of an unreliable flow line composed of parallel-machine stages," *IIE Trans.*, vol. 33, pp. 559–568, 2001.
[7] K.-C. Jeong and Y.-D. Kim, "An approximation method for performance analysis of assembly/disassembly systems with parallel-machine stations," *IIE Trans.*, vol. 31, pp. 391–394, 1999.
[8] J. Li, "Modeling and analysis of production systems with parallel lines," Genral Motors R&D Center, Warren, MI, R&D Rep. 9384, 2002.
[9] ——, "Performance analysis of production systems with rework loops," *IIE Trans.*, vol. 36, pp. 755–765, 2004.
[10] ——, "Throughput analysis in automotive paint shops: A case study," *IEEE Trans. Automat. Sci. Eng.*, vol. 1, pp. 90–98, Jan. 2004.
[11] S.-Y. Chiang, C.-T. Kuo, and S. M. Meerkov, "c-bottlenecks in serial production lines: Identification and application," *Math. Prob. Eng.*, vol. 7, pp. 543–578, 2001.
[12] ——, "DT-bottlenecks in serial production lines: theory and application," *IEEE Trans. Robot. Automat.*, vol. 16, pp. 567–580, Apr. 2000.
[13] D. A. Jacobs and S. M. Meerkov, "A system-theoretic property of serial production lines: Improvability," *Int. J. Syst. Sci.*, vol. 26, pp. 95–137, 1995.