Modeling and analysis of multi-stage transfer lines with unreliable machines and finite buffers

Sheng Yang [a], Cheng Wu [b] and S. Jack Hu [a]

[a] *Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109, USA*

[b] *CIMS-ERC, Department of Automation, Tsinghua University, Beijing 100084, PR China*

This paper models and analyzes multi-stage transfer lines with unreliable machines and finite buffers. The machines have exponential operation, failure, and repair processes. First, a mixed vector–scalar Markov process model is presented based on some notations of mixed vector–scalar operations. Then, several steady-state system properties are deduced from this model. These include the reversibility and duality of transfer lines, conservation of flow, and the flow rate–idle time relationship. Finally, a four-stage transfer line case is used to compare and evaluate the accuracy of some approximation methods presented in the literature with the exact numerical solutions this model can provide. The properties and their proofs in this paper lay the theoretic foundation for some widely held assumptions in decomposition techniques of long transfer lines in the area of manufacturing systems engineering.

1. Introduction

The *transfer line* is one of the major forms of a production system widely used in high volume industries such as automobile manufacturing and consumer electronics production. It consists of a series of machines separated by buffers. Figure 1 depicts a generic multi-stage transfer line *TL*, where there are k machines $(M_1, M_2, \ldots, M_k)$ separated by $k-1$ buffers $(B_1, B_2, \ldots, B_{k-1})$. The squares represent machines and the circles represent buffers. Parts flow from outside to machine M_1, then to buffer B_1, then to machine M_2, and so forth until they reach machine M_k, after which they exit the system. Each part has to be processed on each machine with some positive processing time. Other terms for a transfer line include *flow line, tandem queueing system* and *production line*.

The performance of a transfer line is often impaired by randomness occurring in the system such as random failure/repair events and processing time fluctuations. Hence, buffers are used between machines to mitigate the effects of these variations. By holding in-process parts temporarily, a buffer may enable work to continue elsewhere while some machines in the system are under repair or taking an unusually long time to process a part, and thus increase the system *production rate* (*throughput*). One of the disadvantages of buffers is that they increase the work-in-process (WIP) level in the system and may result in a longer processing cycle time. Therefore a tradeoff

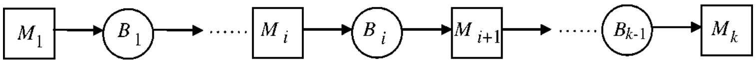

Figure 1. k-stage transfer line *TL* with k machines and $k-1$ buffers.

should be made between an appropriate throughput and a suitable WIP level when transfer lines are designed.

A great deal of literature has been devoted to the modeling and analysis of transfer and production lines using analytical methods since the early 1950's because of their economic importance as well as academic interest. A comprehensive survey by Dallery and Gershwin [9] provides extensive and elaborate reviews up to that time in this area. Current textbooks covering topics in this field include Altiok [1], Buzacott and Shanthikumar [5], Papadapoulos et al. [19], Helber [16], as well as Gershwin [9], which gives a detailed introduction on how to model and analyze transfer lines.

Two types of failures have been considered in the literature: *operation dependent failures* (ODF) and *time dependent failures* (TDF) (for example, see Dallery and Gershwin [9] and Gershwin [12]). An ODF can occur only when the machine is working. On the other hand, a TDF can occur even when the machine is idle (either starved or blocked). As noticed in Buzacott and Hanifin [6], "most failures of transfer lines are ODFs rather than TDFs. Thus an ODF model is more appropriate than a TDF model in the modeling of automated manufacturing systems." The usual assumptions of ODF models are that both the time between failures and the time until repairs are exponentially distributed. This paper assumes ODF models in transfer lines.

As a simplification of reality, three major classes of Markov models have been considered in the literature for the analysis of transfer lines to reflect the variety and characteristics of different kinds of systems. A *discrete synchronous model* is one type of discrete-state-discrete-time (DSDT) Markov process. In this model, parts are processed individually and their operation times on all machines are deterministic and equal. Machines change states (operation, failure, repair and idleness) at some discretized time instant. Buzacott and Hanifin [6], Gershwin and Schick [15], Yeralan and Muth [22], and Gershwin [11] deal with this model. A *discrete asynchronous model* is a discrete-state-continuous-time (DSCT) Markov process. It assumes discrete parts but independent random operation times such as exponential or phase-type probability distribution. Gershwin and Berman [13], Berman [3], Choong and Gershwin [7], and Jeong and Kim [17] take this type of model. A *continuous model* is a continuous time, discrete-and-continuous mixed state (MSCT) Markov model. In this model, the processing speed of machines may be independent and deterministic, but the material is treated as continuous flow rather than discrete parts. The quantity of material in a buffer is a real number ranging from zero to the capacity of the buffer. Dallery et al. [8], Burman [4], and Helber [16] study this model.

For all of these Markov models, the exact analytical solutions of system performance such as the system production rate and average WIP levels are only available in the case of two-machine transfer lines. For example, see Buzacott and Hanifin [6] for DSDT solutions, Gershwin and Berman [13] for DSCT solutions, and Gershwin and Schick [14] for MSCT solutions. Gershwin and Schick [15] attempt to extend their analytic solution to three machines. Gershwin [12] gives a thorough introduction to these solutions. However, it is very difficult (if not hopeless) to obtain exact analytical solutions of transfer lines with more than three machines. The major reason is that the system states increase exponentially with the increase of machines. The curse of dimensionality makes such problems intractable even when more powerful computers are available. As a result, two main approximate techniques have been proposed: *decomposition* methods and *aggregation* methods. The idea of the decomposition technique is to decompose the analysis of a multi-stage line into the analysis of a set of two-machine lines, which are much easier to analyze. The set of two-machine lines is assumed to have equivalent or similar behaviors to the original system. An approximate decomposition method is presented by Gershwin [11] for DSDT models. Choong and Gershwin [7] give the decomposition method for DSCT models. Dallery et al. [8] propose a DDX decomposition method for continuous models. Burman [4] then improves this DDX method. On the other hand, the basic idea of the aggregation technique is to reduce the system dimension by replacing a two-machine-one-buffer sub-line by one single equivalent machine in the system. De Koster [10] and Terracol and David [20] use aggregation techniques. Numerical and simulation experience indicates that the DDX decomposition method is usually fast and reliable with satisfactory accuracy, while the errors of aggregation methods may sometimes be large.

While there has been a great deal of literature on transfer line analysis, there is still much work to be done. For example, although the exact analytical solutions of generic multi-stage transfer lines are not available, their exact numerical solutions do exist. Up to now few people provide comparisons between the exact numerical solutions and the approximation solutions. Most of the comparisons are done between approximation methods and simulation. Therefore designing some model to provide these exact numerical solutions is meaningful in that they may be used to compare the accuracy of different approximate methods in small-size transfer line case studies. This paper presents a mixed vector–scalar Markov process model for multi-stage transfer lines with exponential operation, failure, and repair processes. This model is a natural extension of the scalar DSCT Markov model of two-machine-one-buffer transfer lines proposed by Gershwin and Berman [13]. Several system properties are proven based on this model, including the reversibility and duality of transfer lines, conservation of flow, and the flow rate–idle time relationship. A four-stage transfer line is used to compare and evaluate the accuracy of two decomposition algorithms with the exact numerical solutions this model can provide.

The major contribution of this paper is that for the first time the exact numerical solutions for small-size multi-stage transfer lines can be obtained and can be used to compare and evaluate the accuracy of some other approximation methods presented in the literature. The proofs in this paper also help verify the validity of Markov process models on analyzing manufacturing systems.

2. Notations for mixed vector–scalar operations

In this paper vectors are denoted by bold-faced letters and scalars by normal letters. For example, $\mathbf{0}$ is a null vector and 0 is integer zero. $\boldsymbol{I}_k = (1, 1, \ldots, 1)$ denotes a k-dimension vector with all its elements equal to 1.

Given a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, a scalar a, and an integer i, we have the following mixed vector–scalar operation notations:

1. *Addition* of vector $\boldsymbol{x}$ and scalar a with index i:

$$\boldsymbol{x} + a_i \text{ is defined as } \begin{cases} (x_1, x_2, \ldots, x_{i-1}, x_i + a, x_{i+1}, \ldots, x_n), & \text{if } 1 \leqslant i \leqslant n, \\ \boldsymbol{x}, & \text{else.} \end{cases}$$

2. *Replacement* of vector $\boldsymbol{x}$'s ith element by scalar a:

$$(\boldsymbol{x}|x_i = a) \text{ is defined as } \begin{cases} (x_1, x_2, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_n), & \text{if } 1 \leqslant i \leqslant n, \\ \boldsymbol{x}, & \text{else.} \end{cases}$$

3. *Reduction* of vector $\boldsymbol{x}$'s ith element:

$$\boldsymbol{x}/x_i \text{ is defined as } \begin{cases} (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n), & \text{if } 1 \leqslant i \leqslant n, \\ \boldsymbol{x}, & \text{else.} \end{cases}$$

The above mixed vector–scalar operations can be extended to one-vector–multiple-scalar operations by recursive definitions. For example, given scalar a and b, integer i and j, we define $\boldsymbol{x} + a_i + b_j$ as $(\boldsymbol{x} + a_i) + b_j$, define $\boldsymbol{x}|(x_i = a, x_j = b)$ as $((\boldsymbol{x}|x_i = a)|x_j = b)$, and define $\boldsymbol{x}/(x_i, x_j)$ as $((\boldsymbol{x}/x)/x_j)$.

In addition, the following vector operation notations are used:

4. The *join* of an n-dimension vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and an m-dimension vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ is an $(n+m)$-dimension vector $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$, denoted by $(\boldsymbol{x}, \boldsymbol{y})$.

5. Given vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, its *reverse vector* $\boldsymbol{x}^{\mathrm{r}} = (x_n, x_{n-1}, \ldots, x_1)$.

6. Given vectors $\boldsymbol{J} = (j_1, j_2, \ldots, j_n)$, $\boldsymbol{A} = (a_1, a_2, \ldots, a_n)$, and $\boldsymbol{B} = (b_1, b_2, \ldots, b_n)$, symbol $\sum_{\boldsymbol{J}=\boldsymbol{A}}^{\boldsymbol{B}}$ is used to denote a series of sums $\sum_{j_1=a_1}^{b_1} \sum_{j_2=a_2}^{b_2} \cdots \sum_{j_n=a_n}^{b_n}$, and is called the *vector sum of vector* $\boldsymbol{J}$ from $\boldsymbol{A}$ to $\boldsymbol{B}$. Obviously for two vector sums of vector $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$, we have $\sum_{\boldsymbol{J}_1=\boldsymbol{A}_1}^{\boldsymbol{B}_1} \sum_{\boldsymbol{J}_2=\boldsymbol{A}_2}^{\boldsymbol{B}_2} = \sum_{(\boldsymbol{J}_1,\boldsymbol{J}_2)=(\boldsymbol{A}_1,\boldsymbol{B}_2)}^{(\boldsymbol{B}_1,\boldsymbol{B}_2)}$.

3. Model assumptions and system description

For the convenience of modeling and analyzing the k-stage transfer line depicted in figure 1, we assume the following: all the random variables (processing times, uptimes, and downtimes) are independent. The transfer of parts through the buffers takes no time. Machine failures are ODFs. When a failure occurs, the part stays on the machine; it can be reworked like a new part when the machine is up again (i.e., no scrapping of parts). Even if a machine M_i in the system is up, it cannot process parts if no parts are available in the upstream buffer B_{i-1} (B_{i-1} is empty) or if no room is left in the downstream buffer B_i for the to-be-processed part (B_i is full). In the former condition, the machine is said to be *starved*; in the latter is *blocked* (Dallery and Gershwin [9]) refers to this type of blocking as blocking-before-service (BBS)). A machine either starved or blocked is *idle* and cannot fail. Whenever a machine is up and neither blocked nor starved, it is used for processing. The first machine M_1 is never starved and the last machine M_k is never blocked. The system properties and performance measures studied in this paper are all concerned with steady-state (average long-term) behaviors of the system.

Machine M_i can be in two possible states in the system: *up* (*operational*) or *down* (*under repair*). The binary random variable α_i is defined to be 1 when M_i is up and 0 when M_i is down. When M_i fails, α_i goes from state 1 to state 0. When M_i is fixed, the transition from $\alpha_i = 0$ to $\alpha_i = 1$ occurs; $i = 1, \ldots, k$. Let vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ denote the *machine state vector*.

Processing, failure and repair times for machine M_i are assumed to be exponential random variables with parameters μ_i, p_i and r_i; $i = 1, \ldots, k$, respectively. These quantities are the *processing rate*, *failure rate* and *repair rate* of M_i, respectively. The *isolated efficiency* (*availability*) e_i of machine M_i is the average fraction of the time that M_i would be operational if it were operated in isolation, that is, never starved nor blocked. We have $e_i = r_i/(r_i + p_i)$ (see Dallery and Gershwin [9] and Gershwin [12] for explanation). The *isolated production rate* of machine M_i, ρ_i, is given by $\rho_i = \mu_i e_i$. This is the rate at which machine M_i would process parts in isolation.

The total space for parts in buffer B_i is finite and it is convenient to define the *capacity* N_i of buffer B_i to be the total number of parts that can be stored in B_i plus the space for one part on machine M_{i+1}; $i = 1, \ldots, k-1$. Let vector $\boldsymbol{N} = (N_1, N_2, \ldots, N_{k-1})$ denote the *buffer capacity vector*. We also define the buffer level n_i to be the random variable that indicates the number of parts in buffer B_i at any time, including the part on machine M_{i+1}, if any. It satisfies $0 \leqslant n_i \leqslant N_i$; $i = 1, \ldots, k-1$. Let vector $\boldsymbol{n} = (n_1, n_2, \ldots, n_{k-1})$ denote the *buffer level vector*, we have $\mathbf{0} \leqslant \boldsymbol{n} \leqslant \boldsymbol{N}$.

The condition that machine M_i is neither starved nor blocked is that $n_{i-1} \geqslant 1$ and $n_i \leqslant N_i - 1$. Hence we define a binary function of n_{i-1} and n_i, $L_i(n_{i-1}, n_i)$, to be 1 when $n_{i-1} \geqslant 1$ and $n_i \leqslant N_i - 1$ (the condition that M_i is neither starved nor blocked), otherwise $L_i(n_{i-1}, n_i)$ equals 0 (M_i is either starved or blocked), $i = 1, \ldots, k$. When $i = 1$ or $i = k$, we assume that $n_0 = +\infty$ and $n_k = -\infty$ to reflect the fact that the first machine is never starved and the last machine is never blocked. We have

$$L_i(n_{i-1}, n_i) = \begin{cases} 1, & \text{if } n_{i-1} \geqslant 1 \text{ and } n_i \leqslant N_i - 1, \\ 0, & \text{else,} \end{cases} \qquad i = 1, 2, \ldots, k.$$

The condition that machine M_i is operating on some part is that M_i is up and M_i is neither starved nor blocked, or equivalently, $\alpha_i L_i(n_{i-1}, n_i) = 1$.

The state of the system can be denoted by $\boldsymbol{s} = (n_1, n_2, \ldots, n_{k-1}, \alpha_1, \alpha_2, \ldots, \alpha_k) = (\boldsymbol{n}, \boldsymbol{\alpha})$. System state $\boldsymbol{s}$ is the join of the buffer level vector $\boldsymbol{n}$ and the machine state vector $\boldsymbol{\alpha}$, $\mathbf{0} \leqslant \boldsymbol{s} \leqslant (\boldsymbol{N}, \boldsymbol{I}_k)$. The probability that the system is in this state is written $p(n_1, n_2, \ldots, n_{k-1}, \alpha_1, \alpha_2, \ldots, \alpha_k)$ or $p(\boldsymbol{n}, \boldsymbol{\alpha})$.

4. Balance equations and performance measures

Given a steady state $\boldsymbol{s} = (\boldsymbol{n}, \boldsymbol{\alpha})$, the rate of the system leaving it should equal the rate of entering it, otherwise it is not a steady state. Now consider a small time interval $[t, t+\Delta t)$ with $\Delta t \to 0$. At the beginning the system state is $(\boldsymbol{n}, \boldsymbol{\alpha})$. There are three kinds of events which may enable the system to leave the current state $(\boldsymbol{n}, \boldsymbol{\alpha})$ during this time interval:

1. if machine M_i is down at the beginning ($\alpha_i = 0$), it may be repaired with the rate of r_i during $[t, t+\Delta t)$;
2. if machine M_i is operating on some part ($\alpha_i L_i(n_{i-1}, n_i) = 1$), it may complete the operation with the rate of μ_i during $[t, t+\Delta t)$, or
3. it may fails with the rate of p_i during $[t, t+\Delta t)$.

Note that during such a small time interval $[t, t+\Delta t)$ the probability that two or more events occur is o(Δt) and can be neglected. Therefore the rate of the system leaving $(\boldsymbol{n}, \boldsymbol{\alpha})$ during $[t, t+\Delta t)$ has the following sum form:

$$p(\boldsymbol{n}, \boldsymbol{\alpha}) \sum_{i=1}^{k} \left[r_i(1-\alpha_i) + (\mu_i + p_i)\alpha_i L_i(n_{i-1}, n_i) \right].$$

Similarly, the events which may result in the system entering state $(\boldsymbol{n}, \boldsymbol{\alpha})$ during $[t, t+\Delta t)$ include:

1. at the beginning the system is in state $(\boldsymbol{n} + 1_{i-1} + (-1)_i, \alpha)$ and machine M_i is operating on some part ($\alpha_i L_i(n_{i-1}+1, n_i - 1) = 1$). Then M_i may complete the operation with the rate of μ_i during $[t, t+\Delta t)$ so that the system enters $(\boldsymbol{n}, \boldsymbol{\alpha})$;
2. at the beginning the system is in state $(\boldsymbol{n}, \boldsymbol{\alpha} | \alpha_i = 1 - \alpha_i)$ and machine M_i is down. Then M_i may be repaired ($\alpha_i = 1$) with the rate of r_i during $[t, t+\Delta t)$ so that the system enters $(\boldsymbol{n}, \boldsymbol{\alpha})$;
3. at the beginning the system is in state $(\boldsymbol{n}, \boldsymbol{\alpha} | \alpha_i = 1 - \alpha_i)$ and machine M_i is operating on some part ($(1-\alpha_i)L_i(n_{i-1}, n_i) = 1$). Then M_i may fails ($\alpha_i = 0$) with the rate of p_i during $[t, t+\Delta t)$ so that the system enters $(\boldsymbol{n}, \boldsymbol{\alpha})$.

Again, since the probability that two or more events occur during $[t, t+\Delta t)$ is o(Δt), the probability that the system may enter $(\boldsymbol{n}, \boldsymbol{\alpha})$ from any other situations can be neglected. Hence the rate of the system entering $(\boldsymbol{n}, \boldsymbol{\alpha})$ during $[t, t+\Delta t)$ is

$$\sum_{i=1}^{k} \Big[p\big(\boldsymbol{n} + 1_{i-1} + (-1)_i, \boldsymbol{\alpha}\big) \mu_i \alpha_i L_i(n_{i-1}+1, n_i - 1) \\ + p(\boldsymbol{n}, \boldsymbol{\alpha} | \alpha_i = 1 - \alpha_i)\big(r_i \alpha_i + p_i(1-\alpha_i) L_i(n_{i-1}, n_i)\big) \Big].$$

According to the analysis above, we obtain the system steady-state balance equation:

$$\begin{aligned} p(\boldsymbol{n},\boldsymbol{\alpha}) \sum_{i=1}^{k} \big[r_i(1-\alpha_i) + (\mu_i + p_i)\alpha_i L_i(n_{i-1}, n_i)\big] \\ = \sum_{i=1}^{k} \big[p\big(\boldsymbol{n} + 1_{i-1} + (-1)_i, \boldsymbol{\alpha}\big)\mu_i \alpha_i L_i(n_{i-1}+1, n_i - 1) \\ + p(\boldsymbol{n}, \boldsymbol{\alpha} | \alpha_i = 1-\alpha_i)\big(r_i\alpha_i + p_i(1-\alpha_i)L_i(n_{i-1}, n_i)\big)\big], \end{aligned} \quad (1)$$

where $\mathbf{0} \leqslant$ state $(\boldsymbol{n},\boldsymbol{\alpha}) \leqslant (\boldsymbol{N},\boldsymbol{I}_k)$.

It should be noted that if some state $(\boldsymbol{n} + 1_{i-1} + (-1)_i, \boldsymbol{\alpha})$ on the right hand side of equation (1) is out of the range $\mathbf{0}$ to $(\boldsymbol{N},\boldsymbol{I}_k)$, its corresponding function $L_i(n_{i-1}+1, n_i - 1) = 0$ by definition, and hence that state can be omitted on the right hand side. Since $0 \leqslant n_i \leqslant N_i$; $i = 1,\dots,k-1$, and $\alpha_i = \{0,1\}$; $i = 1,\dots,k$, there are a total of $T = \prod_{i=1}^{k-1}(N_i + 1)2^k$ steady-states and balance equations. The system normalization equation is

$$\sum_{\boldsymbol{n}=0}^{\boldsymbol{N}} \sum_{\boldsymbol{\alpha}=0}^{\boldsymbol{I}_k} p(\boldsymbol{n},\boldsymbol{\alpha}) = 1. \quad (2)$$

Balance equation (1) plus normalization equation (2) provide $T+1$ linear equations with T unknown, so one balance equation is redundant. Theoretically speaking, the exact numerical solutions of steady-state probability $p(\boldsymbol{n},\boldsymbol{\alpha})$ are unique and can be derived from solving these $T+1$ linear equations, even though the number of equations expand exponentially with the increase of machines in the system.

System performance measures can be calculated based on these steady-state probabilities. The *efficiency* of machine M_i in the system, E_i, is defined as the probability of M_i working in the system ($\alpha_i L_i(n_{i-1}, n_i) = 1$). It corresponds to the proportion of time during which M_i is neither idle (starved or blocked) nor down. E_i can be calculated by the following sum:

$$E_i = \sum_{\boldsymbol{n}=0}^{\boldsymbol{N}} \sum_{\boldsymbol{\alpha}=0}^{\boldsymbol{I}_k} p(\boldsymbol{n},\boldsymbol{\alpha})\alpha_i L_i(n_{i-1}, n_i), \quad i = 1,2,\dots,k. \quad (3)$$

The difference between the isolated efficiency e_i and the efficiency E_i is that e_i is a characteristic of machine M_i itself while E_i is affected by other machines and buffers in the system. The *production rate* of machine M_i in the system, P_i, is the average number of parts on which M_i finishes operations in the system per unit of time. The relationship between P_i and E_i is $P_i = \mu_i E_i$. In the next section we demonstrate that all the machines have identical production rates in the system, which equal the *system production rate* P.

Another important performance measure is the average *work-in-process (WIP) level* (expected inventory) of buffer B_i. It can be calculated by the following equation:

$$\bar{n}_i = \sum_{\boldsymbol{n}=0}^{\boldsymbol{N}} \sum_{\boldsymbol{\alpha}=0}^{\boldsymbol{I}_k} n_i p(\boldsymbol{n}, \boldsymbol{\alpha}), \quad i = 1, 2, \ldots, k-1. \tag{4}$$

The *total WIP level* in the system is given by $\bar{n} = \bar{n}_1 + \bar{n}_2 + \cdots + \bar{n}_{k-1}$. Using Little's law, the average flow time of a part, W, can be obtained as: $W = \bar{n}/P$ (see Dallery and Gershwin [9]).

The probability that machine M_i is down in the system, denoted by $p(\alpha_i = 0)$, can be written as

$$p(\alpha_i = 0) = \sum_{\boldsymbol{n}}^{\boldsymbol{N}} \sum_{\boldsymbol{\alpha}/\alpha_i=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}, \boldsymbol{\alpha} | \alpha_i = 0). \tag{5}$$

5. Analysis of system properties

5.1. *Reversibility and duality of transfer lines*

Consider a flow line, *TL**, which is obtained by reversing the flow of parts in transfer line *TL* in figure 1. The first machine M_1^* of *TL** is the same as the last machine M_k of *TL*, the last machine M_k^* of *TL** is the same as the first machine M_1 of *TL*. More generally, machine M_i^* of *TL** is the same as machine M_{k-i+1} of *TL*. Also, buffer B_i^* of TL^* is the same as buffer B_{k-i} of *TL*. All quantities of *TL** are labeled by letters with $*$. Therefore $(\mu_i^*, p_i^*, r_i^*) = (\mu_{k-i+1}, p_{k-i+1}, r_{k-i+1})$, $N_i^* = N_{k-i}$. All other assumptions of *TL** are the same as *TL*. The *reversibility* means that the production rate of the reversed line *TL** is the same as that of the original line *TL*. The *duality* is that the average WIP level of buffer B_i^* in *TL** plus the average WIP level of the corresponding buffer B_{k-i} in *TL* sums up to the capacity N_i^* of buffer B_i^*. Muth [18], Ammar and Gershwin [2], and others establish the reversibility and/or duality of transfer lines based on the comparison of the sample paths of the two systems using evolution equations. Alternatively, this paper uses balance equation (1) to prove the reversibility and duality.

Theorem 1. Given transfer line *TL* and its reversed line *TL**, the production rate P_i^* of machine M_i^* in *TL** equals the production rate P_{k-i+1} of machine M_{k-i+1} in *TL*. The average WIP level of buffer B_i^* in *TL** plus the average WIP level of the corresponding buffer B_{k-i} in *TL* equals the capacity N_i^* of buffer B_i^*, i.e., $P_i^* = P_{k-i+1}$, $\bar{n}_i^* + \bar{n}_{k-i} = N_i^*$.

Proof. Theorem 1 follows if it can be proven that the steady-state probability $p(\boldsymbol{n}, \boldsymbol{\alpha})$ of line *TL* equals the steady-state probability $p^*(\boldsymbol{N}^* - \boldsymbol{n}^{\mathrm{r}}, \boldsymbol{\alpha}^{\mathrm{r}})$ of line *TL**, where $\boldsymbol{N}^*$

is the buffer capacity vector of TL^*, $\boldsymbol{n}^{\rm r}$ and $\boldsymbol{\alpha}^{\rm r}$ are the reverse vectors of $\boldsymbol{n}$ and $\boldsymbol{\alpha}$, respectively.

Assume that $(\boldsymbol{n}^*, \boldsymbol{\alpha}^*) = (\boldsymbol{N}^* - \boldsymbol{n}^{\rm r}, \boldsymbol{\alpha}^{\rm r})$ for line TL^*, then it is observed that the following statements hold:

1. $n_i = N_i - n^*_{k-i},\ N_i = N^*_{k-i};\quad i = 1, 2, \ldots, k-1.$
2. $\mu_i = \mu^*_{k-i+1},\ p_i = p^*_{k-i+1},\ r_i = r^*_{k-i+1};\quad i = 1, 2, \ldots, k.$
3. Function $L_i(n_{i-1}, n_i)$ of line $TL = L^*_{k-i+1}(n^*_{k-i}, n^*_{k-i+1})$ of line TL^* since condition $n_{i-1} \geqslant 1$ equals $n^*_{k-i+1} \leqslant N^*_{k-i+1} - 1$, and condition $n_i \leqslant N_i - 1$ equals $n^*_{k-i} \geqslant 1$.
4. Function $L_i(n_{i-1}+1, n_i-1)$ of line $TL = L^*_{k-i+1}(n^*_{k-i}+1, n^*_{k-i+1}-1)$ of line TL^* since condition $n_{i-1}+1 \geqslant 1$ equals $n^*_{k-i+1} - 1 \leqslant N^*_{k-i+1} - 1$, and condition $n_i - 1 \leqslant N_i - 1$ equals $n^*_{k-i} - 1 \geqslant 1$.

Substituting the four statements above into balance equation (1), we find that the balance equation (1) of line TL concerning state $(\boldsymbol{n}, \boldsymbol{\alpha})$ is exactly the same as that of line TL^* concerning state $(\boldsymbol{N}^* - \boldsymbol{n}^{\rm r}, \boldsymbol{\alpha}^{\rm r})$. This implies that the T balance equations of line TL are the same as those of line TL^* except that the sequences of equations in two lines are different. Thus $p(\boldsymbol{n}, \boldsymbol{\alpha}) = p^*(\boldsymbol{N}^* - \boldsymbol{n}^{\rm r}, \boldsymbol{\alpha}^{\rm r})$, $\boldsymbol{0} \leqslant (\boldsymbol{n}, \boldsymbol{\alpha}) \leqslant (\boldsymbol{N}, \boldsymbol{I}_k)$. Combining it with definition (3), (4) and $P_i = \mu_i E_i$, we know $P^*_i = P_{k-i+1}$, $\bar{n}^*_i + \bar{n}_{k-i} = N^*_i$. $\square$

By theorem 1, we deduce that if a transfer line has symmetric parameters, then its average WIP level $\bar{n}_i + \bar{n}_{k-i} = N_i$ and the average WIP level of the middle buffer is about 50%.

5.2. *Transient states*

The probability that buffer level $n_i = x$ and machine state $\alpha_j = y$ in the system is denoted by $p(n_i = x, \alpha_j = y)$, and can be calculated by

$$p(n_i = x, \alpha_j = y) = \sum_{\boldsymbol{n}/n_i=0}^{\boldsymbol{N}/N_i} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_i = x, \boldsymbol{\alpha}|\alpha_j = y).$$

For example, $p(n_{j-1} = 0, \alpha_j = 0)$ is the probability that machine M_j is down ($\alpha_j = 0$) and its upstream buffer B_{j-1} is empty ($n_{j-1} = 0$); $p(n_j = N_j, \alpha_j = 0)$ is the probability that machine M_j is down ($\alpha_j = 0$) and its downstream buffer B_j is full ($n_j = N_j$). According to the assumptions in section 3, machine M_j cannot operate, and hence cannot fail, if upstream buffer B_{j-1} is empty or downstream buffer B_j is full. This yields the following theorem:

Theorem 2. $p(n_{j-1} = 0, \alpha_j = 0) = 0$; $2 \leqslant j \leqslant k$, and $p(n_j = N_j, \alpha_j = 0) = 0$; $1 \leqslant j \leqslant k-1$.

Proof. Only the first equation is proven. The proof of the second equation is similar.

Consider balance equation (1) of state $(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0)$ in line *TL* and break down the sum of i on both sides into three terms: $i = j-1$, $i = j$, and $(i \neq j-1$ and $i \neq j)$. Note that two of the terms on the right hand side can be omitted since $L_{j-1}(n_{j-2}+1, -1) = 0$ and $L_j(0, n_j) = 0$. Now operating on both sides of this equation with vector sums $\sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}}$ and $\sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}}$, we obtain the following equation form:

$$(1) + (2) + (3) + (4) + (5) = (6) + (7) + (9),$$

where

$$(1) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0) r_{j-1}(1-\alpha_{j-1}),$$

$$(2) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0) \mu_{j-1}\alpha_{j-1}L_{j-1}(n_{j-2}, n_{j-1}),$$

$$(3) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0) p_{j-1}\alpha_{j-1}L_{j-1}(n_{j-2}, n_{j-1}),$$

$$(4) = r_j \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0),$$

$$(5) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|\alpha_j = 0) \times \sum_{i=1,\ i\neq j-1,\ i\neq j}^{k} \bigl(r_i(1-\alpha_i) + (\mu_i + p_i)\alpha_i L_i(n_{i-1}, n_i)\bigr),$$

$$(6) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p\bigl(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|(\alpha_j = 0, \alpha_{j-1} = 1-\alpha_{j-1})\bigr) r_{j-1}\alpha_{j-1},$$

$$(7) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} p\bigl(\boldsymbol{n}|n_{j-1} = 0, \boldsymbol{\alpha}|(\alpha_j = 0, \alpha_{j-1} = 1-\alpha_{j-1})\bigr) \times p_{j-1}(1-\alpha_{j-1})L_{j-1}(n_{j-2}, n_{j-1}),$$

$$(8) = \sum_{\boldsymbol{n}/n_{j-1}=0}^{\boldsymbol{N}/N_{j-1}} \sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}} \sum_{i=1,\ i\neq j-1,\ i\neq j}^{k} \bigl\{p\bigl(\boldsymbol{n}|n_{j-1} = 0 + 1_{i-1} + (-1)_i, \boldsymbol{\alpha}|\alpha_j = 0\bigr) \times \mu_i\alpha_i L_i(n_{i-1}+1, n_i-1)$$

$$+\,p\big(\boldsymbol{n}|n_{j-1}=0,\boldsymbol{\alpha}|(\alpha_j=0,\alpha_i=1-\alpha_i)\big)\big(r_i\alpha_i+p_i(1-\alpha_i)L_i(n_{i-1},n_i)\big)\big\}.$$

It can be proven (see Yang [21] for a proof) that: (1) = (6), (2) = (7), and (5) = (8). Canceling these terms from the two sides of the equation above yields a new reduced equation: (3) + (4) = 0. Since both (3) and (4) are weighted sums of non-negative probabilities, we know (3) = (4) = 0. (4) = 0 implies the first equation of theorem 2. □

5.3. *Balance between repair states and operation states*

From section 4 we know that E_j is the probability of machine M_j working in the system, and $p(\alpha_j = 0)$ is the probability that machine M_j is down in the system. The following theorem asserts that the rate of transitions from the set of states in which machine M_j is down to the set of states in which M_j is operational ($r_j p(\alpha_j = 0)$) is equal to the rate of transitions in the opposite direction ($p_j E_j$).

Theorem 3. $r_j p(\alpha_j = 0) = p_j E_j$, $1 \leqslant j \leqslant k$.

The proof of theorem 3 is similar to that of theorem 2. First consider balance equation (1) of state $(\boldsymbol{n}, \boldsymbol{\alpha}|\alpha_j = 0)$ in line *TL* and break down the sum of i on both sides into two terms: $i = j$ and $i \neq j$. After operating on both sides of that equation with vector sums $\sum_{\boldsymbol{n}=0}^{\boldsymbol{N}}$ and $\sum_{\boldsymbol{\alpha}/\alpha_j=0}^{\boldsymbol{I}_{k-1}}$, most of the terms on two sides are canceled and only two terms left: $r_j p(\alpha_j = 0) = p_j E_j$. See Yang [21] for a detailed proof.

5.4. *Balance between two sets of operation states*

Denoting the probability that machine M_i is working and the buffer level n_j of buffer B_j is x ($0 \leqslant x \leqslant N_j$) by $w_i(n_j = x)$, then $w_i(n_j = x)$ can be calculated by

$$w_i(n_j=x)=\sum_{\boldsymbol{n}/n_j=0}^{\boldsymbol{N}/N_j}\sum_{\boldsymbol{\alpha}}^{\boldsymbol{I}_k}p(\boldsymbol{n}|n_j=x,\boldsymbol{\alpha})\alpha_i L_i(n_{i-1},n_i),\quad 0\leqslant x\leqslant N_j.$$

For example, $w_j(n_j = x)$ is the probability that machine M_j is working with x parts in its downstream buffer B_j, and $w_{j+1}(n_j = x + 1)$ is the probability that machine M_{j+1} is working with $(x + 1)$ parts in its upstream buffer B_j. Theorem 4 asserts that the rate of transitions from the set of states with x parts in buffer B_j to the set of states with $x + 1$ parts in buffer B_j, $\mu_j w_j(n_j = x)$, is equal to the rate of transitions in the opposite direction, $\mu_{j+1} w_{j+1}(n_j = x + 1)$.

Theorem 4.

$$\mu_j w_j(n_j=x)=\mu_{j+1}w_{j+1}(n_j=x+1),\quad 1\leqslant j\leqslant k-1,\ 0\leqslant x\leqslant N_j-1. \tag{6}$$

Proof. This can be proven by induction. The following is the outline of proof. See Yang [21] for a detailed proof.

(1) For $x = 0$, consider balance equation (1) of state $(\boldsymbol{n}|n_j = 0, \boldsymbol{\alpha})$ in line *TL* and break down the sum of i on both sides into three terms: $i = j$, $i = j+1$, and ($i \neq j$ and $i \neq j+1$). One term on the left hand side can be omitted since $L_{j+1}(0, n_{j+1}) = 0$, and two terms on the right hand side can be omitted since $L_j(n_{j-1}+1, -1) = 0$ and $L_{j+1}(0, n_{j+1}) = 0$. After operating on both sides of the equation with vector sums $\sum_{\boldsymbol{n}/n_j=0}^{\boldsymbol{N}/N_j}$ and $\sum_{\boldsymbol{\alpha}=0}^{\boldsymbol{I}_k}$, most of the terms on two sides are canceled and only two terms are left: $\mu_j w_j(n_j = 0) = \mu_{j+1} w_{j+1}(n_j = 1)$.

(2). Assume that theorem 4 holds for $x \leqslant m-1$. For $x = m$, consider balance equation (1) of state $(\boldsymbol{n}|n_j = m, \boldsymbol{\alpha})$ in line *TL* and break down the sum of i on both sides into three terms: $i = j$, $i = j+1$, and ($i \neq j$ and $i \neq j+1$). Operating on both sides of that equation with vector sums $\sum_{\boldsymbol{n}/n_j=0}^{\boldsymbol{N}/N_j}$ and $\sum_{\boldsymbol{\alpha}=0}^{\boldsymbol{I}_k}$, we note that most of the terms on two sides are canceled and only four terms left in the equation:

$$\mu_j w_j(n_j = m) + \mu_{j+1} w_{j+1}(n_j = m) = \mu_j w_j(n_j = m-1) + \mu_{j+1} w_{j+1}(n_j = m+1).$$

By induction the second term on the left equals the first term on the right. This yields the equation of theorem 4 for $x = m$: $\mu_j w_j(n_j = m) = \mu_{j+1} w_{j+1}(n_j = m+1)$. Hence theorem 4 holds. □

5.5. *Conservation of flow*

The following theorem demonstrates the conservation of flow in the system. It shows that all machines in a transfer line have the same production rate. Therefore the system production rate $P = P_i = P_j$, $1 \leqslant i, j \leqslant k$.

Theorem 5. $\mu_i E_i = \mu_j E_j$, $1 \leqslant i, j \leqslant k$.

Proof. Summing both sides of equation (6) in theorem 5 for $x = 0, 1, \ldots, N_j - 1$, we obtain

$$\begin{aligned}
\text{left hand side} &= \mu_j \sum_{x=0}^{N_j-1} w_j(n_j = x) \\
&= \mu_j \sum_{x=0}^{N_j-1} \sum_{\boldsymbol{n}/n_j=0}^{\boldsymbol{N}/N_j} \sum_{\boldsymbol{\alpha}}^{\boldsymbol{I}_k} p(\boldsymbol{n}|n_j = x, \boldsymbol{\alpha}) \alpha_j L_j(n_{j-1}, n_j) = \mu_j E_j, \\
\text{right hand side} &= \mu_{j+1} \sum_{x=0}^{N_j-1} w_{j+1}(n_j = x+1) \\
&= \mu_{j+1} \sum_{x=0}^{N_j-1} \sum_{\boldsymbol{n}/n_j=0}^{\boldsymbol{N}/N_j} \sum_{\boldsymbol{\alpha}}^{\boldsymbol{I}_k} p(\boldsymbol{n}|n_j = x+1, \boldsymbol{\alpha}) \alpha_{j+1} L_{j+1}(n_j, n_{j+1}) \\
&= \mu_{j+1} E_{j+1}.
\end{aligned}$$

Therefore $\mu_i E_i = \mu_j E_j$, $1 \leqslant i, j \leqslant k$. □

5.6. Flow rate–idle time relationship

The next theorem, theorem 6, establishes the flow rate–idle time relation. It shows that a machine is operational exactly as often as it would be in isolation when it is neither blocked nor starved (not idle) in the system.

Theorem 6. $e_i = p(\alpha_i = 1 | n_{i-1} \neq 0 \text{ and } n_i \neq N_i)$ and $E_i = e_i p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i)$, $1 \leqslant i \leqslant k$.

Proof. By definition of conditional probability,

$$p(\alpha_i = 1 | n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = \frac{p(\alpha_i = 1, n_{i-1} \neq 0 \text{ and } n_i \neq N_i)}{p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i)}. \tag{7}$$

Note that $p(\alpha_i = 1, n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = E_i$ by definition of E_i, and

$$\begin{aligned} & p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) \\ & \quad = p(\alpha_i = 1, n_{i-1} \neq 0 \text{ and } n_i \neq N_i) + p(\alpha_i = 0, n_{i-1} \neq 0 \text{ and } n_i \neq N_i) \\ & \quad = E_i + p(\alpha_i = 0, n_{i-1} \neq 0 \text{ and } n_i \neq N_i). \end{aligned}$$

By theorem 2 we know that $\{\alpha_i = 0\}$ and $\{n_{i-1} = 0 \text{ or } n_i = N_i\}$ are mutually exclusive events. Thus

$$p(\alpha_i = 0, n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = p(\alpha_i = 0).$$

Theorem 3 shows $p(\alpha_i = 0) = p_i E_i / r_i$. Substituting all these relationships into equation (7) yields

$$p(\alpha_i = 1 | n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = \frac{E_i}{E_i + \frac{p_i E_i}{r_i}} = \frac{r_i}{r_i + p_i} = e_i.$$

On the other hand, by equation (7) we have

$$\begin{aligned} E_i &= p(\alpha_i = 1 | n_{i-1} \neq 0 \text{ and } n_i \neq N_i) p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) \\ &= e_i p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i). \end{aligned}$$ □

The production rate of machine M_i in the system can be written

$$\begin{aligned} P_i &= \mu_i E_i = \mu_i e_i p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) \\ &= \mu_i e_i \big(1 - p(n_{i-1} = 0) - p(n_i = N_i) + p(n_{i-1} = 0 \text{ and } n_i = N_i)\big) \\ &\approx \mu_i e_i \big(1 - p(n_{i-1} = 0) - p(n_i = N_i)\big). \end{aligned}$$

The last approximation comes from the observation that the probability of one machine in the system being starved and blocked simultaneously is very small in reality. This approximate relationship between flow rate and idle time is widely used in decomposition techniques of long transfer lines.

5.7. *Impact of bottleneck machines*

The next theorem asserts that if the isolated production rate of some machine in a transfer line is much smaller than those of other machines, the system production rate will be mainly dominated by that bottleneck machine, and the efficiencies of other machines will approach zero.

Theorem 7. If the isolated production rate of some machine M_j is made much smaller than those of other machines in the system, i.e., $\rho_j \to \mathrm{o}(\rho_i)$, $i = 1, \ldots, k$ and $i \neq j$, then the system production rate $P \to \rho_j$, machine M_j's efficiency $E_j \to e_j$, while other machines' efficiency $E_i \to 0$, $i \neq j$.

Proof. By theorem 6, $P = \rho_i p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = \rho_j p(n_{j-1} \neq 0 \text{ and } n_j \neq N_j)$,

$$\rho_j \to \mathrm{o}(\rho_i), \text{ yields } p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) = \frac{\rho_j}{\rho_i} p(n_{j-1} \neq 0 \text{ and } n_j \neq N_j) \to 0.$$

By theorem 6, $E_i = e_i p(n_{i-1} \neq 0 \text{ and } n_i \neq N_i) \to 0$.

In addition, $p(n_{i-1} \neq 0 \text{ or } n_i \neq N_i) \to 0$ implies that $p(n_{i-1} = 0 \text{ or } N_i) \to 1$, $i = 1, \ldots, k$ and $i \neq j$.

This yields $p(n_{j-1} = N_{j-1}) \to 1$ and $p(n_j = 0) \to 1$. Thus $p(n_{j-1} \neq 0 \text{ and } n_j \neq N_j) \to 1$.

Consequently, $E_j = e_j p(n_{j-1} \neq 0 \text{ and } n_j \neq N_j) \to e_j$ and $P \to \rho_j$. □

5.8. *Numerical experiments*

Although the system states increase exponentially with increasing numbers of machines, it is possible to obtain the exact numerical solutions of system performance for small size transfer lines by solving $T + 1$ linear equations. These exact solutions can then be used to evaluate the accuracy of different approximate methods. For example, Gershwin [12] presents the DDX decomposition method for discrete synchronous transfer lines. Yang [21] points out that the deduction of equations describing the interruptions of flow is not accurate and proposes the addition of two new terms, $-p_i p(i-1; 011)/E_u(i)$ and $-p_{i+1} p(i+1; N11)/E_d(i)$ to equation (4.31) and equation (4.32) in Gershwin [12], respectively. (Gershwin has admitted this inaccuracy on his web site, see http://web.mit.edu/manuf-sys/www/) Next, a four-stage transfer line is used to show that these modifications can lead to more accurate results.

Consider a four-stage discrete asynchronous transfer line. The parameters of the system are $p_i = 0.1$, $r_i = 0.9$; $i = 1, 2, 3, 4$. $\mu_1 = \mu_3 = \mu_4 = 10$, the buffer capacity vector $\boldsymbol{N} = (3, 3, 3)$. Now vary the processing rate μ_2 of machine M_2 from 1 to 20 and observe the system production rate as a function of processing rate μ_2. The solutions of the original DDX method by Gershwin [12], the modified solutions by Yang [21], and the exact numerical solutions by solving $T = \prod_{i=1}^{k-1}(N_i + 1)2^k = 1024$ linear equations using Matlab each time are depicted in figure 2. Figure 3 shows the

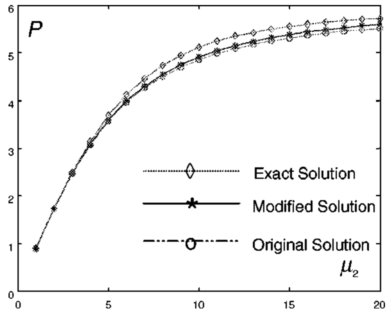

Figure 2. Production rate vs. μ_2 curve.

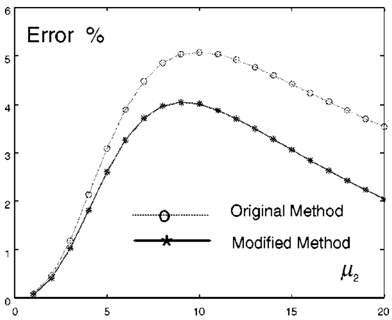

Figure 3. Errors vs. μ_2 curve.

errors of these two approximate methods compared with the exact numerical solutions. Note that

(1) The errors of the modified method are always smaller than those of the original method. The maximum error of the original method is about 5.08%, while the maximum error of the modified method is only 4.04%.

(2) The maximum errors of these two approximate methods are both reached when machine M_2's processing rate μ_2 is close to other machines' processing rates, in other words, when the line is nearly homogeneous. This phenomenon has been noted by us in other numerical examples.

6. Conclusions

This paper studies discrete asynchronous transfer lines subject to exponential operation, failure, and repair processes. A mixed vector–scalar Markov process model is presented to describe the operation, failure and repair behaviors of multi-stage transfer lines with k unreliable machines and $k-1$ buffers. Some important steady-state system properties, such as the reversibility and duality of transfer lines, conservation of flow, and the flow rate–idle time relationship, are deduced from this model. These properties are widely assumed as prerequisites to decomposition techniques of long transfer lines in other literature.

Theoretically speaking, the method in this paper may be extended to multi-stage discrete synchronous transfer lines and continuous transfer lines, and similar steady-state transition equations may be deduced based on those mixed vector–scalar operations. However, the transition equations of discrete synchronous transfer lines and continuous transfer lines will be much more complex than the balance equations of discrete asynchronous lines presented in this paper. This is because in a discrete synchronous transfer line the probability that two or more events occur within one unit of time cannot be neglected, and in a continuous transfer the interior and boundary equations should be considered separately. Similar system properties can be proven for discrete synchronous transfer lines and continuous transfer lines because most of the terms in the transition equations can be canceled from both sides of the transition equations after some vector sum or integration operations are taken. Therefore we are confident that these properties still hold with more general models pertaining to a wide range of transfer lines.

For a k-stage transfer line, there are a total of $T = \prod_{i=1}^{k-1}(N_i + 1)2^k$ steady states and the same number of balance equations. Theoretically speaking, steady-state probability $p(\boldsymbol{n}, \alpha)$ can be derived from solving these T linear equations plus the normalization equation. Then system performance measures can be calculated using the steady-state distribution $p(\boldsymbol{n}, \alpha)$. However, since the system state space expands exponentially with an increase in the number of machines, this method is only applicable to transfer lines of very small size, and this paper has more theoretical value than practical interest. For the analysis of middle and large size production lines, approximate techniques are the only feasible solutions. From the viewpoint of computational complexity, approximate techniques convert the problem from solving exponentially increasing linear equations to solving polynomial increasing non-linear equations. For example, the DDX decomposition algorithm presented in Dallery et al. [8] and Gershwin [12] turns the problem from solving $T = \prod_{i=1}^{k-1}(N_i + 1)2^k$ linear equations to solving $6k$ non-linear equations. Since the computational complexity is reduced from exponential increase to polynomial increase, the new challenge of solving non-linear equations seems inevitable in approximate methods instead of solving linear equations in the original problem.

Acknowledgement

We would like to express our special thanks to S.B. Gershwin for his continuous encouragement, great patience and constructive suggestions on this paper. We also appreciate the valuable comments from two anonymous reviewers. Support for S. Yang and S.J. Hu from the Engineering Research Center for Reconfigurable Machining Systems at the University of Michigan is also acknowledged.

References

[1] T. Altiok, *Performance Analysis of Manufacturing Systems* (Springer, New York, 1996).

[2] M.H. Ammar and S.B. Gershwin, Equivalence relations in queueing models of fork/join queueing networks with blocking, Perf. Eval. 19 (1989) 233–245.

[3] O. Berman, Efficiency and production rate of a transfer line with two machines and a finite storage buffer, Europ. J. Oper. Res. 9 (1982) 295–308.

[4] M.H. Berman, New results in flow line analysis, Ph.D. thesis, Massachusetts Institute of Technology, USA (1995).

[5] J.A. Buzacott and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems* (Prentice Hall, Englewood Cliffs, NJ, 1993).

[6] J.A. Buzacott and L.E. Hanifin, Models of automatic transfer lines with inventory banks: a review and comparison, AIIE Trans. 10 (1978) 197–207.

[7] Y.F. Choong and S.B. Gershwin, A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times, IIE Trans. 19 (1987) 150–159.

[8] Y. Dallery, R. David and X. Xie, Approximate analysis of transfer lines with unreliable machines and finite buffers, IEEE Trans. A.C. 34 (1989) 943–953.

[9] Y. Dallery and S.B. Gershwin, Manufacturing flow line systems: a review of models and analytical results, Queueing Systems 12 (1992) 3–94.

[10] M.B.M. De Koster, Estimation of line efficiency by aggregation, Int. J. Prod. Res. 25 (1987) 615–626.

[11] S.B. Gershwin, An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite sotrage space and blocking, Oper. Res. 35 (1987) 291–305.

[12] S.B. Gershwin, *Manufacturing Systems Engineering* (Prentice-Hall, Englewood Cliffs, NJ, 1994).

[13] S.B. Gershwin and O. Berman, Analysis of transfer lines consisting of two unreliable machines with random processing times and finite sorage buffers, AIIE Trans. 13 (1981) 2–11.

[14] S.B. Gershwin and I.C. Schick, Continuous model of an unreliable two-stage material flow system with a finite interstage buffer, Technical Report LIDS-R-1039, MIT Laboratory for Information and Decision Systems (1980).

[15] S.B. Gershwin and I.C. Schick, Modeling and analysis of three-stage transfer lines with unreliable machine and finite buffers, Oper. Res. 31 (1983) 354–380.

[16] S. Helber, *Performance Analysis of Flow Lines with Non-Linear Flow of Material* (Springer, 1998).

[17] K.-C. Jeong and Y.-D. Kim, Performance analysis of assembly/disassembly systems with unreliable machines and random processing times, IIE Trans. 30 (1998) 41–53.

[18] E.J. Muth, The reversibility property of production lines, Manag. Sci. 25 (1979) 152–158.

[19] H. Papadapoulos, C. Heavey and J. Browne, *Queuing Theory in Manufacturing Systems Analysis and Design* (Chapman & Hall, London, 1993).

[20] C. Terracol and R. David, An aggregation method for performance valuation of transfer lines with unreliable machines and finite buffers, in: *IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC (1987).

[21] S. Yang, Analysis and control of unreliable and complex processing production lines, Ph.D. thesis, Tsinghua Univ., Beijing, China (1996).

[22] S. Yeralan and E.J. Muth, A general model of a production line with intermediate buffer and station breakdown, IIE Trans. 19 (1987) 130–139.