A Decomposition Method for Analyzing Inhomogeneous Assembly/Disassembly Systems

Stanley B. Gershwin [a,*] and Mitchell H. Burman [b]

[a] *Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139-4307*
E-mail: gershwin@mit.edu

[b] *Analytics, Inc., 101 Rogers Street, Cambridge, Massachusetts 02142*
E-mail: mburman@analytics-consulting.com

This paper presents a decomposition method for analyzing assembly/disassembly manufacturing systems with continuous material, exponential repair and failure time distributions, and processing times that differ from machine to machine. Decomposition equations are derived, an algorithm is developed, special cases are explored, and numerical results are discussed.

1. Introduction

1.1. Motivation

Nearly all manufacturing involves assembly, and much modern manufacturing is high-volume. For example, Burman, Gershwin, and Suyematsu (1998) describe an automated printer mechanism production system at Hewlett Packard that made hundreds of thousand of units every month. The process involved making separate subassemblies, and then assembling them into a single product. In this system, machines failed at random times and required random lengths of time to repair. When they were operational, machines took a constant amount of time to do each operation, and that time differed from machine to machine. There were finite in-process storage areas at various points in the production system. The effects of the buffers on the system performance were studied and shown to be of great importance.

These characteristics are common in many industries, and fast, accurate

* Corresponding author.

analytic methods to predict factory performance (production rate and average inventories), and thus to help design factories, are needed. This paper contributes such a method.

1.2. Approach

We describe a single-part-type assembly/disassembly production system as a network with continuous material, unreliable machines, and finite buffers. To evaluate production rate and expected in-process inventory for a limited class of systems — acyclic networks — we propose a decomposition of the system into two-machine, one-buffer lines. The flow of material in the buffer of each small line closely approximates the behavior of the flow of material in the corresponding buffer of the original system. We develop equations that relate the parameters of the machines of the small lines to one another and to the original system's machines and buffers. There are enough such equations to determine the parameters of the machines in the small lines. We develop an algorithm that solves these equations efficiently and produces estimates of production rate and average inventory levels.

1.3. Literature

Surveys of the flow line literature appear in Buxey *et al.* (1973) and Buzacott and Hanifin (1978b). Smunt and Perkins (1985) focused on asynchronous flow lines, including a great deal of the simulation literature; Perros (1986) reviewed general results of queuing networks with blocking; and Awate and Sastry (1987) summarized the solution methods of the important papers.

Dallery and Gershwin (1992) is the most recent survey of the literature that relates to this paper. Recent books include Buzacott and Shanthikumar (1993), Askin and Standridge (1993), Papadopoulos, Heavey, and Brown (1993), Gershwin (1994), and Altiok (1997). See also Le Bihan (1998).

Early analytic work focused on various two-machine models (Buzacott, 1967b; Buzacott, 1972; Gershwin and Berman, 1981; Gershwin and Schick, 1980). Gershwin (1987) developed a decomposition method to analyze a system with more than two machines whose machines had equal, deterministic processing times. Dallery, David, and Xie (1988) provided the DDX algorithm, which was a fast and reliable technique for solving the decomposition equations.

This was not directly applicable to systems in which machines had different processing times. A decomposition for systems with machines with different processing times was presented in Choong and Gershwin (1987). It used the Gershwin and Berman (1981) two-machine, discrete material, exponential processing time model as its building block. Gershwin (1989) solved the Choong and Gershwin equations with a DDX-type procedure. However, the exponential processing time assumption limited the usefulness of this method for systems with small buffers and with deterministic processing times that differed from machine to machine.

Earlier work on the decomposition of long lines with continuous material and different processing times was done by Semery (1987), Hong and Seong (1989), and Glassey and Hong (1993).

Burman (1995) and Burman and Gershwin (1997) extend earlier decomposition work to tandem flow lines with continuous material, differing operation rates, and unreliable machines. The present paper is a generalization of this work. It is also a generalization of the assembly/disassembly network decomposition analysis of Gershwin (1991) and Di Mascolo, David, and Dallery (1991) for *homogeneous* systems, systems in which machines could only operate at the same speed.

Le Bihan (1998) describes several homogenization methods for tandem lines. These methods can be extended to A/D systems, so that the original nonhomgeneous tree-structured system is transformed into a homogeneous one, which is then analyzed by the methods of Gershwin (1991) or Di Mascolo, David, and Dallery (1991).

An alternative assembly system analysis approach was provided by Liu and Buzacott (1990). In this method, machines and buffers are replaced by equivalent systems without buffers. This allows the assembly/disassembly system to be replaced by a line, which is then analyzed.

1.4. Outline

We describe the system to be analyzed in Section 2; then in Section 3 we describe an approximate decomposition. The algorithm involves an *evaluation sequence* — the order in which the decomposed building blocks are evaluated. An example of such a sequence appears in Section 4. We describe an algorithm for solving the decomposition equations in Section 5. This algorithm follows the DDX algorithm (Dallery et al., 1988) and its extension, the ADDX algorithm

(Burman, 1995; Burman and Gershwin, 1997) very closely. An improvement, which reduces computation time, is described in Section 6. We present numerical results in Section 7 and conclude in Section 8.

2. Model Assumptions and Notation

The model described here is an extension of the homogeneous, deterministic processing time assembly/disassembly model of Gershwin (1991, 1994) (for discrete material) and Di Mascolo, David, and Dallery (1991) (for continuous material, but where all machines operated at the same speed), the continuous material two-machine line of Gershwin and Schick (1980) and Gershwin (1994), and the continuous material long-line analysis of Burman (1995) and Gershwin and Burman (1997). In this section, we describe the assumptions, notation, and conventions of this model.

The notation is consistent with that of Gershwin (1994), Burman (1995), and Burman and Gershwin (1997). The network structure and decomposition notation are shown in Figures 1 and 2.

1. Network Structure and Notation

   1a The factory produces one part type. Assembly and disassembly take place, but each machine and each buffer deals with only a single stage of the product or a single component or subassembly. There is a single path for each component and subassembly, there is no rework or rejection, and there is no reentrant material flow.

   1b The network can be represented as a connected graph in which the machines are nodes and the buffers are arcs. (We develop an algorithm which is specific to tree-structured, or acyclic, networks.) There are k_M machines and k_B buffers. Each buffer is connected to exactly two machines, and is named b or $b(j,i)$, or (j,i) if the machine upstream is j and the machine downstream is M_i. The machine upstream of Buffer b is called $M_{u(b)}$, and the machine immediately downstream of b is $M_{d(b)}$. (If $b=(j,i)$, then $j=u(b)$ and $i=d(b)$.)

   1c The set of buffers upstream of Machine M_i is called $U(i)$ and the set of buffers downstream of M_i is $D(i)$ (Figure 1). One of these sets may be empty. If $U(i)$ is empty, M_i is an *input machine.* If $D(i)$ is empty, M_i is

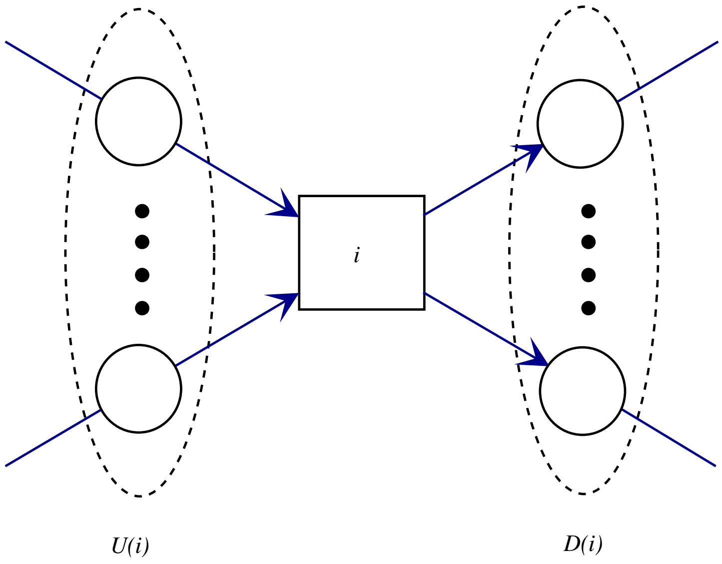

Figure 1. Network Notation

an *output machine*. Input machines are assumed to be never starved and output machines are never blocked. Define u_i to be the cardinality of $U(i)$, the number of buffers immediately upstream of Machine M_i, and let d_i be the cardinality of $D(i)$.

2. Individual Machines and Buffers

2a Buffer b has capacity N_b. That is, the amount of material x_b in buffer b can be between 0 and N_b. The material is continuous, so x_b and N_b are real numbers.

2b Machine M_i can be operational (up) or under repair (down). When it is operational and not starved or blocked, it can process material at rate μ_i. That is, M_i reduces the amounts of material in each of the buffers in $U(i)$ and increases the amounts of material in each of the buffers in $D(i)$ at rate μ_i. It does this until the first of the upstream buffers becomes empty or the first of the downstream buffers becomes full, whichever happens first. If the machine is starved or blocked, it may be stopped or it may process material at a slower rate than μ_i, as described in Points 3a and 3b.

2c Machine M_i in isolation (*i.e.,* never impeded by other machines or buffers) is completely defined by parameters r_i, p_i, and μ_i. When it is operational, it processes material at rate μ_i (as defined in Point 2b). Its failure and repair processes are exponentially distributed with parameters r_i and p_i (defined precisely in points 4b and 4c. It is convenient to define the *isolated efficiency*

$$e_i = \frac{r_i}{r_i + p_i}$$

and *isolated production rate*

$$\rho_i = \mu_i e_i.$$

3. Blockage and Starvation

3a Machine M_i is *fully blocked* if one (or more) of the buffers in $D(i)$ is full and the machine downstream of that buffer is failed or fully blocked or fully starved. It is *fully starved* if one (or more) of the buffers in $U(i)$ is empty and the machine upstream of that buffer is failed or fully blocked or fully starved. When a machine is fully blocked or starved, it cannot operate and it cannot fail.

3b Machine M_i is *partially blocked* if it is not fully blocked and if one (or more) of the buffers in $D(i)$ is full and the machine downstream of that buffer is operating at a rate which less than μ_i. The *speed* of a machine is the instantaneous rate at which it is processing material. Machine M_i operates at a speed which is the minimum of the speeds of all such partially blocking machines.
Machine M_i is *partially starved* if it is not fully starved and if one (or more) of the buffers in $D(i)$ is empty and the machine upstream of that buffer is operating at a rate which less than μ_i. Machine M_i operates at a speed which is the minimum of the speeds of all such partially starving machines.
Define $\mu_i(t)$ as the speed of M_i at time t. It satisfies $0 \leq \mu_i(t) \leq \mu_i$.

3c Dallery, David, and Xie (1989) show that in a *homogeneous* line (one in which all μ_i are the same), a machine cannot be simultaneously fully starved and fully blocked. Here, Machine M_i cannot be fully starved or fully blocked by more than one buffer. At most one adjacent buffer can be empty or full and (if empty) preceded by a machine which is down or fully starved or (if full) followed by a machine which is down or fully blocked. This is because in order for the machine to be fully blocked and starved at the same time, two adjacent machines must be down, and the second failure must occur at a precise time instant after the first. Because failures are independent, this is an event of probability 0.

3d The previous point does not carry over to partial starvations or blockages. It would appear to be impossible for most systems, and unlikely in others. One situation in which simultaneous partial starvation and blockage can happen is the following three-machine line. Assume Machines 1 and 3 have the same speed, which is less than that of Machine 2 (*i.e.*, $\mu_1 = \mu_3 < \mu_2$). Assume that the buffers are the same size ($N_1 = N_2 = N$). Consider the scenario in which, first, Machines 1 and 3 are up and Machine 2 is down. If this persists long enough, the first buffer becomes full and the second becomes empty ($x_1 = N$ and $x_2 = 0$). In the second part of the scenario, Machine 2 is repaired and all three machines stay up for a long time. Because Machine 2 is faster than Machine 1 and Machine 2 is not blocked (for some time), x_1 decreases. Similarly, Machine 2 is faster than Machine 3 and is not starved, so x_2 increases. This continues until, simultaneously, $x_1 = 0$ and $x_2 = N$. (The simultaneity is ensured by the equal speeds of the first and last machines and the equal buffer sizes.) This situation cannot occur in a system with all machines operating at the same speed because the buffer levels do not change when all machines are operational.

We approximate the probability of simultaneous partial starvation and blockage of the same machine as 0. We *assume* that this does not cause serious difficulty. Evidence for this is (a) the preliminary simulation experimentation by Choong and Gershwin (1987) on a related class of systems: discrete material transfer lines with exponentially distributed operation, repair, and failure times (reported in Gershwin, 1994) in which M_2 of a three-machine line had to be much faster than the other two in order for there to be a noticeable discrepancy between decomposition and simulation results; and (b) the good performance of the flow line analysis algorithms of Burman (1995), which are based on the same assumptions as those made here.

4. State Space and Dynamics

4a α_i indicates the repair state of Machine M_i and x_b is the amount of material (the level) of Buffer b. The capacity of Buffer b is N_b, and $0 \leq x_b \leq N_b$. The state of the system is

$$s = (x_1, \cdots, x_{k_B}, \alpha_1, \cdots, \alpha_{k_M}).$$

4b If Machine M_i is under repair, the time until the repair is completed is distributed exponentially with parameter r_i. That is,

$$\text{prob}\left[\alpha_i(t+\delta t) = 1 \middle| \alpha_i(t) = 0\right] = r_i \delta t + o(\delta t).$$

4c While Machine M_i is operational and neither starved nor blocked, the time until the next failure is distributed exponentially with parameter p_i. That is,

$$\begin{aligned} &\text{prob}\left[\alpha_i(t+\delta t) = 0 \middle| \alpha_i(t) = 1, \right. \\ &\left. (x_b(t) > 0 \; \forall \; b \in U(i)), (x_b(t) < N_b \; \forall \; b \in D(i))\right] = p_i \delta t + o(\delta t). \quad (1) \end{aligned}$$

Failures are *operation-dependent* (Buzacott and Hanifin, 1978): machines cannot fail when they are fully starved or fully blocked. They fail at reduced rates (defined in Point 4d) when they are partially starved or blocked.

4d The amount of material in Buffer b at time t, $x_b(t)$, $0 \leq x_b(t) \leq N_b$, varies in a way that depends on the states of Buffer b and those of the adjacent machines and buffers.

Let $i = u(b)$ or $d(b)$. That is, M_i is either the machine upstream of Buffer b or the machine downstream. There are three cases to consider. During time interval $(t, t+\delta t)$:

i. When $0 < x_b < N_b$

A. the change in x_b is $(\alpha_{u(b)}(t)\mu_{u(b)} - \alpha_{d(b)}(t)\mu_{d(b)})\delta t$

B. the probability of repair of Machine M_i, that is, the probability that $\alpha_i(t+\delta t) = 1$ given that $\alpha_i(t) = 0$, is $r_i \delta t$.

C. the probability of failure of Machine M_i, that is, the probability that $\alpha_i(t+\delta t) = 0$ given that $\alpha_i(t) = 1$, is $p_i \delta t$.

ii. When $x_b = 0$ and buffer b is the only buffer immediately upstream of Machine $M_{d(b)}$ that is empty and no buffer immediately downstream of $M_{d(b)}$ is full,

A. the change in x is $(\alpha_{u(b)}(t)\mu_{u(b)}(t) - \alpha_{d(b)}(t)\mu_{d(b)})^{+}\delta t$
This says (1) when $x_b = 0$, x_b can only increase; and (2) the rate of increase depends on the *current* speed of Machine $M_{u(b)}$.

B. the probability of repair of Machine M_i, is $r_i \delta t$.

C. if Machine $M_{u(b)}$ is down or fully blocked, Machine $d(b)$ is fully starved and it cannot fail. If Machine $M_{u(b)}$ is up, Machine $M_{d(b)}$ may be partially starved. The probability of failure of Machine $M_{d(b)}$ is

$$\frac{p_{d(b)}\mu_{d(b)}(t)}{\mu_{d(b)}}$$

where

$$\mu_{d(b)}(t) = \min(\mu_{u(b)}(t), \mu_{d(b)}).$$

This is because, when the buffer is empty, the downstream machine can operate no faster than the upstream. If the upstream machine is faster than the downstream, the downstream machine operates as it does normally, and the buffer immediately becomes non-empty. If the upstream machine is slower than the downstream, it limits the speed of the downstream when the buffer is empty. We assume here that the failure rate of a machine is proportional to its operating speed (with a proportionality constant that can differ from machine to machine).

iii. When $x_b = N_b$ and buffer b is the only buffer immediately downstream of Machine $M_{d(b)}$ that is full and no buffer immediately upstream of $M_{d(b)}$ is empty,

A. the change in x is $(\alpha_{u(b)}(t)\mu_{u(b)} - \alpha_{d(b)}(t)\mu_{d(b)}(t))^{-}\delta t$

B. the probability of repair of Machine M_i is $r_i \delta t$.

C. if Machine $M_{d(b)}$ is down, Machine $M_{u(b)}$ is fully blocked and cannot fail. If Machine $M_{d(b)}$ is up, the probability of failure of Machine $M_{u(b)}$ is

$$\frac{p_{u(b)}\mu_{u(b)}(t)}{\mu_{u(b)}}$$

where

$$\mu_{u(b)}(t) = \min(\mu_{u(b)}(t), \mu_{u(b)}).$$

The upstream machine can fail only if the downstream machine is operational.
If more than one buffer adjacent to Machine $M_{d(b)}$ is at an extreme that slows down that machine (*i.e.,* an upstream buffer empty or a downstream buffer full), they must all be operating at the same speed. If they were not, at least one of the empty or full buffers would become non-empty or non-full.

5. Steady-State Behavior

5a A steady-state distribution exists because the state space is bounded. However, the steady state may be a function of the initial condition, if the network is not tree-structured. (For example, if the network is a simple loop, the number of parts in that loop is constant — the same as the number when the system was initialized.) Therefore the system might be non-ergodic. Consequently, we can define

$$\mathbf{p}(s|s(0)) = \lim_{t\to\infty} \text{ prob } \{ \text{ state of the system at time } t = s| \\ \text{state of the system at time } 0 = s(0)\}.$$

5b *Performance Measures* The *production rate* of Machine M_i, in parts per time unit, is

$$P_i(s(0)) = E(\mu_i(t)|s(0))$$

The *average level* of Buffer b is

$$E(x_b(t)|s(0)) = \bar{x}_b(s(0)) = \int_s x_b \text{ prob } (s|s(0)). \tag{2}$$

The steady-state performance measures depend on the initial state of the system. For example, if a system has a loop and if the initial number of parts in a loop is 0, the number of parts in that loop will remain 0 forever. Consequently, the production rate in that loop and the average buffer levels there will be 0. The goal of Section 3 is to describe an efficient method for calculating P_i and $\bar{x}_b$ for tree-structured systems, in which the initial state does not influence the steady-state behavior.

Conservation of Flow Because there is no mechanism for the creation or destruction of material, flow is conserved in steady state, or

$$P(s(0)) = P_1(s(0)) = P_2(s(0)) = \cdots = P_{k_M}(s(0)). \tag{3}$$

6. Tree-Structured Networks
   In the following, we only consider tree-structured networks. Consequently, we need not be concerned with the initial state $s(0)$. Because of the tree structure, the number of buffers is 1 less than the number of machines. Let $k_M = k$; then $k_B = k - 1$. Even in this case, however, it is difficult to write a simple expression for P. Consequently, we evaluate P by calculating the production rate of the approximate two-machine lines defined in the decomposition below.

3. Assembly/Disassembly Decomposition Equations

In this section, we derive the equations of the decomposition. The structure of the decomposition is displayed in Figure 2; to each buffer there corresponds a two-machine line. The buffer in that line is the same size as the original buffer. The parameters of the two pseudo-machines are chosen so that the flow of material (both its average rate and its interruption behavior) through the buffer in the two-machine line closely approximates the flow through the original. The upstream pseudo-machine machine represents all upstream causes of flow variability, and the downstream pseudo-machine represents all downstream causes of flow variability.

The two-machine lines must be analyzed to determine throughput, probabilities of blockage and starvation, and other quantities. In this algorithm, the analysis of these lines is performed according to the continuous model of Gershwin and Schick (1980) and Gershwin (1994).

There are six parameters to be determined for each buffer, or $6(k-1)$ unknowns. We must therefore derive that number of equations. In order to do this, we define the meaning of a pseudo-machine being down. This leads to the meanings of failure and repair of such a machine, which in turn helps to determine

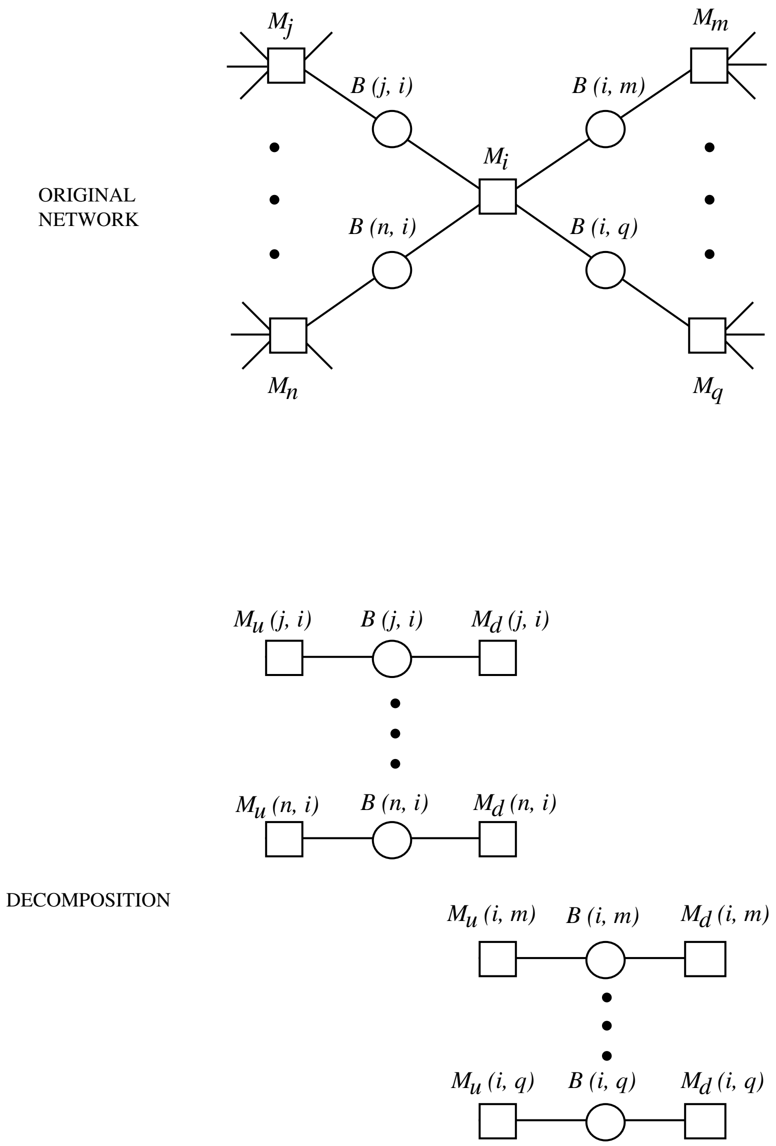

Figure 2. Network Decomposition Notation

the repair and failure rates. We must also determine the production rate of the pseudo-machine.

3.1. Pseudo-Machine States

Here, we define what it means for pseudo-machine $M_u(i,m)$ to be down. The goal is to have the pseudo-machine down for periods whose average frequency ($p_u(i,m)$) and duration ($1/r_u(i,m)$) are similar to those of the periods during which no material enters buffer (i,m) in the original system. In addition, the production rate when the pseudo-machine is up and not starved or blocked — $\mu_u(i,m)$ — should be close to the average rate that material enters Buffer (i,m) when it is not blocked.

The event {Pseudo-machine $M_u(i,m)$ down} is

$$M_u(i,m) \text{ down } = M_i \text{ down or } \{M_u(\beta) \text{ down and } x_\beta = 0 \text{ for some } \beta \in U(i)\}$$

$$\text{or } \{M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ for some } \beta \in D(i), \beta \neq (i,m)\} \tag{4}$$

That is, flow into buffer (i,m) is stopped, due to upstream causes, when Machine M_i is down. It is also stopped when an immediately upstream buffer β is empty and the pseudo-machine upstream of that buffer is down. Finally, it is stopped when one of the buffers downstream of Machine M_i *other than* (i,m) is full and the pseudo-machine immediately downstream of it is down.

From this, we can define the event {Pseudo-machine $M_u(i,m)$ up}:

$$M_u(i,m) \text{ up } = \text{ NOT } \{M_u(i,m) \text{ down }\} =$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \tag{5}$$

There are similar expressions for $\{M_d(i,m) \text{ down}\}$ and $\{M_d(i,m) \text{ up}\}$.

We use these relationships and others to derive the $6(k-1)$ equations needed for the parameters. Among them are conservation of flow equations (Section 3.2), flow rate-idle time equations (Section 3.3), resumption of flow equations (Section 3.4), interruption of flow equations (Section 3.5), and boundary conditions (Section 3.6).

3.2. Conservation of Flow

Conservation of flow is the assertion that the flow through all the two-machine lines is the same:

$$P(b) = P \text{ for all } b \tag{6}$$

3.3. Flow Rate-Idle Time

The following is an approximate extension of the *Flow Rate-Idle Time* equation.

$$P = \mu_i e_i \left[1 - \sum_{\beta \in U(i)} \text{prob } (x_\beta = 0) - \sum_{\beta \in D(i)} \text{prob } (x_\beta = N_\beta) \right] \tag{7}$$

Versions of this equation appear in all of the earlier decompositions (Gershwin, 1994; Burman, 1995). It says that the rate that Machine M_i operates is its isolated production rate ($\mu_i e_i$) times the probability that it is not starved or blocked. The two-machine approximations for the probabilities include the effects of partial starvation and partial blockage.

Consider a two-machine line whose buffer is called β, whose upstream machine is called $M_u(\beta)$ and whose downstream machine is called $M_d(\beta)$. Define

$$e_u(\beta) = \frac{r_u(\beta)}{r_u(\beta) + p_u(\beta)}; \quad e_d(\beta) = \frac{r_d(\beta)}{r_d(\beta) + p_d(\beta)}. \tag{8}$$

Consider the special case of (7) for the upstream machine:

$$P(\beta) = \mu_u(\beta) e_u(\beta)(1 - \text{prob } (x_\beta = N_\beta)),$$

and the special case for the downstream machine:

$$P(\beta) = \mu_d(\beta) e_d(\beta)(1 - \text{prob } (x_\beta = 0)).$$

These equations can be written

$$\text{prob } (x_\beta = N_\beta) = 1 - \frac{P(\beta)}{\mu_u(\beta) e_u(\beta)}$$

and

$$\text{prob } (x_\beta = 0) = 1 - \frac{P(\beta)}{\mu_d(\beta) e_d(\beta)}.$$

Now consider (7) for Machine M_i in a general tree-structured network. The last two expressions can be used to relate the behavior of the approximate two-machine lines with that of the full network. Using them and (6) in (7) leads to

$$P = \mu_i e_i \left[1 - \sum_{\beta \in U(i)} \left\{ 1 - \frac{P}{\mu_d(\beta) e_d(\beta)} \right\} - \sum_{\beta \in D(i)} \left\{ 1 - \frac{P}{\mu_u(\beta) e_u(\beta)} \right\} \right]. \tag{9}$$

Using the definitions of u_i and d_i, (9) can be written

$$P = \mu_i e_i \left[1 - u_i + P \sum_{\beta \in U(i)} \frac{1}{\mu_d(\beta) e_d(\beta)} - d_i + P \sum_{\beta \in D(i)} \frac{1}{\mu_u(\beta) e_u(\beta)} \right]$$

or, after more manipulation,

$$u_i + d_i - 1 = P \left(-\frac{1}{\mu_i e_i} + \sum_{\beta \in U(i)} \frac{1}{\mu_d(\beta) e_d(\beta)} + \sum_{\beta \in D(i)} \frac{1}{\mu_u(\beta) e_u(\beta)} \right). \tag{10}$$

To write this in a form that is useful for the algorithm, we must solve it for $\mu_u(i,m)$, where $(i,m) \in D(i)$, and for $\mu_d(j,i)$, where $(j,i) \in U(i)$:

$$\mu_u(i,m) =$$

$$\frac{1}{e_u(i,m)} \left(\frac{u_i + d_i - 1}{P} + \frac{1}{\mu_i e_i} - \sum_{\beta \in U(i)} \frac{1}{\mu_d(\beta) e_d(\beta)} - \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{1}{\mu_u(\beta) e_u(\beta)} \right)^{-1} \tag{11}$$

and

$$\mu_d(j,i) =$$

$$\frac{1}{e_d(j,i)} \left(\frac{u_i + d_i - 1}{P} + \frac{1}{\mu_i e_i} - \sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} \frac{1}{\mu_d(\beta) e_d(\beta)} - \sum_{\beta \in D(i)} \frac{1}{\mu_u(\beta) e_u(\beta)} \right)^{-1} \tag{12}$$

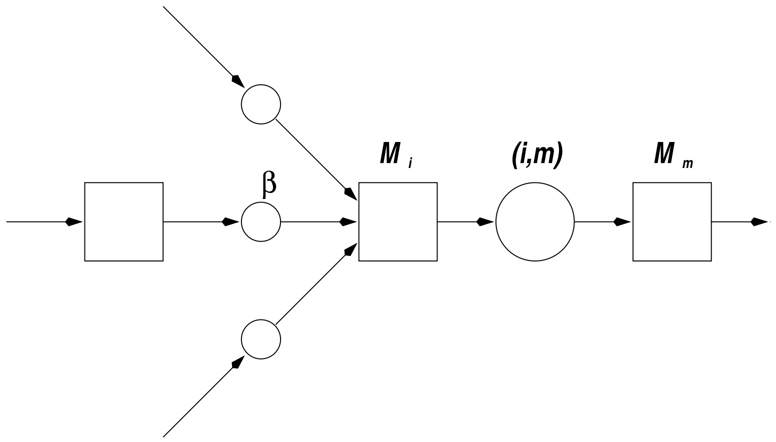

Figure 3. Original network in which $\beta \in U(i)$

3.4. Resumption of Flow

The resumption of flow equations determine the repair rates ($r_u(i,m)$ and $r_d(j,i)$) of the pseudo-machines in the two-machine lines. The upstream repair rate is defined by

$$r_u(i,m)\delta t = \text{prob}\ \Big[M_u(i,m) \text{ up at } t+\delta t \Big|$$

$$M_u(i,m) \text{ down and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]. \tag{13}$$

From the definitions of $M_u(i,m)$ being down and up ((4) and (5)),

$$r_u(i,m)\delta t = \text{prob}\ \Big[M_i \text{ up} \ \text{ and } \ \{M_u(\beta) \text{ up} \ \text{ or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \ \text{ and}$$

$$\{M_d(\beta) \text{ up} \ \text{ or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \ \text{ at } t+\delta t \Big|$$

$$M_i \text{ down} \ \text{ or } \ \{M_u(\beta) \text{ down} \ \text{ and } x_\beta = 0 \text{ for some } \beta \in U(i)\} \ \text{ or}$$

$$\{M_d(\beta) \text{ down} \ \text{ and } x_\beta = N_\beta \text{ for some } \beta \in D(i), \beta \neq (i,m)\} \ \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big].$$

The case in which $\beta \in U(i)$ is illustrated in Figures 3 and 4. The case in which $\beta \in D(i)$ and $\beta \neq (i,m)$ is illustrated in Figures 5 and 6.

We can expand this as

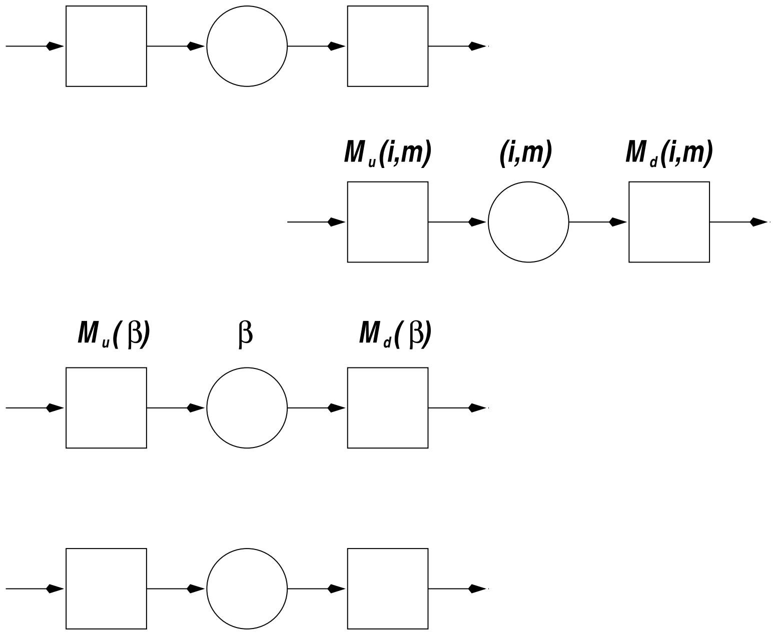

Figure 4. Decomposition of network in which $\beta \in U(i)$

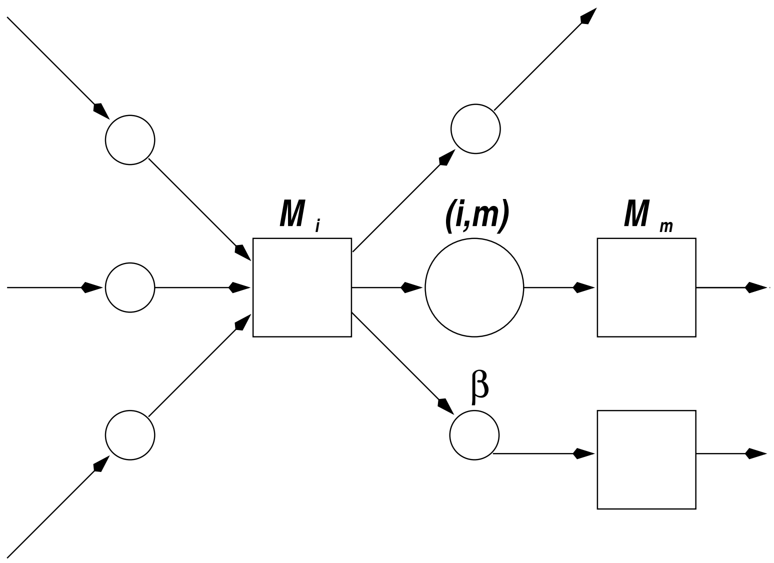

Figure 5. Original network in which $\beta \in D(i)$, $\beta \neq (i, m)$

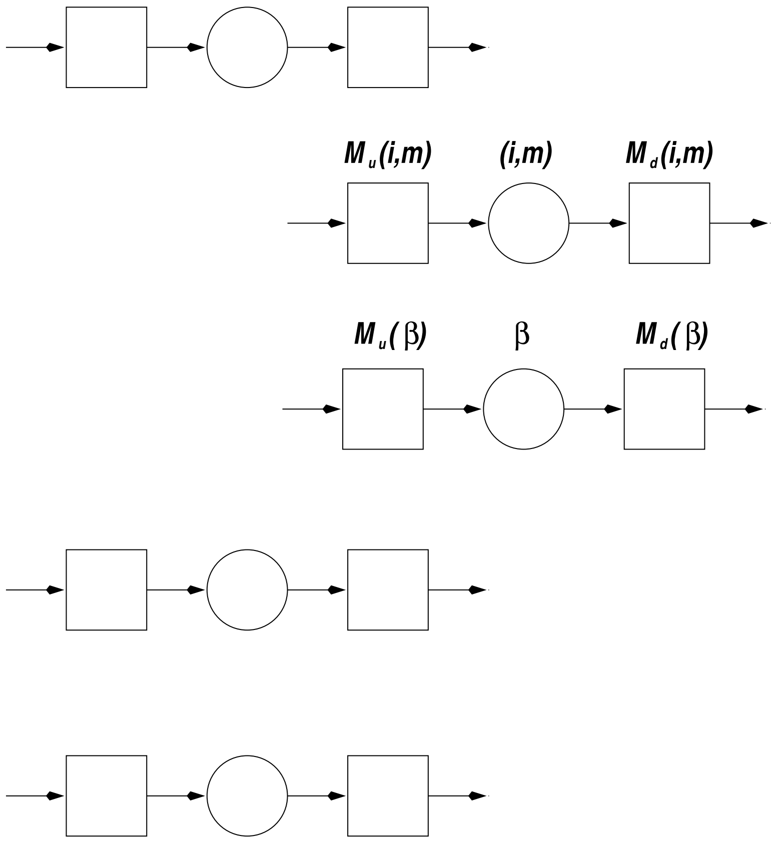

Figure 6. Decomposition of network in which $\beta \in D(i)$, $\beta \neq (i,m)$

$$r_u(i,m)\delta t = \sum_{\beta \in U(i)} A(\beta)X(\beta) + \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} A(\beta)X(\beta)$$

$$+B(i,m)\left(1 - \sum_{\substack{\beta \in D(i) \bigcup U(i) \\ \beta \neq (i,m)}} X(\beta)\right)$$

where

- if $\beta \in U(i)$,

$$A(\beta) = \text{prob } \Big[M_u(i,m) \text{ up at } t + \delta t \Big|$$

$$M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big] \tag{14}$$

$$X(\beta) = \text{prob } \Big[M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } x_{(i,m)} < N_{(i,m)} \Big|$$

$$M_u(i,m) \text{ down and } x_{(i,m)} < N_{(i,m)} \Big]$$

Here and in the rest of the paper, we do not write the time argument when the conditional and conditioning events occur at the same time.

- if $\beta \in D(i), \beta \neq (i,m)$,

$$A(\beta) = \text{prob } \Big[M_u(i,m) \text{ up at } t + \delta t \Big|$$

$$M_u(\beta) \text{ down and } x_\beta = N_\beta \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

$$X(\beta) = \text{prob } \Big[M_u(\beta) \text{ down and } x_\beta = N_\beta \text{ and } x_{(i,m)} < N_{(i,m)} \Big|$$

$$M_u(i,m) \text{ down and } x_{(i,m)} < N_{(i,m)} \Big]$$

and

$$B(i,m) = \text{prob } \Big[M_u(i,m) \text{ up at } t + \delta t \Big|$$

$$M_i \text{ down and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]. \tag{15}$$

To determine $r_u(i,m)$, we must find expressions for $A(\beta)$, $X(\beta)$, and $B(i,m)$. From (15), $B(i,m)$ is the probability of M_i changing state in $(t, t+\delta t)$. This is because in order for $M_u(i,m)$ to be up at $t+\delta t$, M_i must be up at $t+\delta t$. To order δt, the probability of more than one event (repair or failure) occurring in an interval is 0. Therefore $M_u(i,m)$ is up at $t+\delta t$ if and only if M_i gets repaired in $(t, t+\delta t)$, so

$$B(i,m) = r_i \delta t.$$

From (14), if $\beta \in U(i)$,

$$A(\beta) = \text{prob } \Big[M_u(i,m) \text{ up at } t+\delta t \Big|$$

$$M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } M_i \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

In order for $M_u(i,m)$ to be up at $t+\delta t$, M_i and $M_u(\beta)$ must be up. Therefore,

$$A(\beta) = \text{prob } \Big[M_i \text{ up and } M_u(\beta) \text{ up at } t+\delta t \Big|$$

$$M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } M_i \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

The only difference between the state of the system at t and at $t+\delta t$ is the change of state of $M_u(\beta)$, so

$$A(\beta) = \text{prob } \Big[M_u(\beta) \text{ up at } t+\delta t \Big| M_u(\beta) \text{ down at } t \Big] = r_u(\beta)\delta t.$$

By similar reasoning, if $\beta \in D(i)$,

$$A(\beta) = r_u(\beta)\delta t.$$

To determine $X(\beta)$, for $\beta \in U(i)$,

$$X(\beta) = \text{prob } \Big[M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } x_{(i,m)} < N_{(i,m)} \Big|$$

$$M_i \text{ down or } \{M_u(\gamma) \text{ down and } x_\gamma = 0 \text{ for some } \gamma \in U(i)\} \text{ or}$$

$$\{M_d(\gamma) \text{ down and } x_\gamma = N_\gamma \text{ for some } \gamma \in D(i), \gamma \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \Big]$$

Since the first event is a subset of the second,

$$X(\beta) = \frac{\text{prob } (M_u(\beta) \text{ down and } x_\beta = 0 \text{ and } x_{(i,m)} < N_{(i,m)})}{\text{prob } (M_u(\beta) \text{ down and } x_{(i,m)} < N_{(i,m)})}$$

In this fraction, the numerator is $\mathbf{p}_\beta(0,0,1)$, the steady-state probability that the buffer is empty, the first machine is down, and the second machine is up, in two-machine line β. This quantity, all other two-machine line probabilities needed here, and all two-machine performance measures, can be evaluated using the methods of Gershwin (1994), Section 3.10.

The denominator satisfies (Gershwin, 1994)

$$\text{prob } (M_u(\beta) \text{ down and } x_{(i,m)} < N_{(i,m)}) = \frac{p_u(i,m)E_u(i,m)}{r_u(i,m)}$$

in which $E_u(i,m)$ is the probability that $M_u(i,m)$ is up. Therefore,

$$X(\beta) = \frac{\mathbf{p}_\beta(0,0,1)r_u(i,m)}{p_u(i,m)E_u(i,m)}$$

Similarly, if $\beta \in D(i)$,

$$X(\beta) = \frac{\mathbf{p}_\beta(N_\beta,0,1)r_u(i,m)}{p_u(i,m)E_u(i,m)}$$

We can now put all these expressions together, and we have

$$r_u(i,m) = r_i \left(1 - \sum_{\beta \in U(i)} \frac{\mathbf{p}_\beta(0,0,1)r_u(i,m)}{p_u(i,m)E_u(i,m)} - \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{\mathbf{p}_\beta(N_\beta,1,0)r_u(i,m)}{p_u(i,m)E_u(i,m)} \right)$$

$$+ \sum_{\beta \in U(i)} \frac{r_u(i,m)\mathbf{p}_\beta(0,0,1)r_u(\beta)}{p_u(i,m)E_u(i,m)} + \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{r_u(i,m)\mathbf{p}_\beta(N_\beta,1,0)r_d(\beta)}{p_u(i,m)E_u(i,m)} \tag{16}$$

and, by similar reasoning,

$$r_d(j,i) = r_i \left(1 - \sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} \frac{\mathbf{p}_\beta(0,0,1)r_d(j,i)}{p_d(j,i)E_d(j,i)} - \sum_{\beta \in D(i)} \frac{\mathbf{p}_\beta(N_\beta,1,0)r_d(j,i)}{p_d(j,i)E_d(j,i)} \right)$$

$$+ \sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} \frac{r_d(j,i)\mathbf{p}_\beta(0,0,1)r_u(\beta)}{p_d(j,i)E_d(j,i)} + \sum_{\beta \in D(i)} \frac{r_d(j,i)\mathbf{p}_\beta(N_\beta,1,0)r_d(\beta)}{p_d(j,i)E_d(j,i)} \tag{17}$$

3.5. Interruption of Flow

The interruption of flow equations determine $p_u(i,m)$ and $p_d(i,m)$. Similarly to (13),

$$p_u(i,m)\delta t = \text{prob } \Big[M_u(i,m) \text{ down at } t+\delta t \Big| \quad M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big] \tag{18}$$

or,

$$p_u(i,m)\delta t = \text{prob } \Big[M_i \text{ down or } \{M_u(\beta) \text{ down and } x_\beta = 0 \text{ for some } \beta \in U(i)\} \text{ or}$$

$$\{M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ for some } \beta \in D(i), \beta \neq (i,m)\} \text{ at } t+\delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big].$$

From the definition of $\{M_u(i,m) \text{ down}\}$ (4), if M_i is down, then $M_d(\beta)$ is down, for all $\beta \in U(i)$. But if $M_u(\beta)$ is down and $x_\beta = 0$ for some $\beta \in U(i)$, then $M_d(\beta)$ must be up (because, in a two-machine line of this type, the state (0,0,0) is transient). Consequently, $\{M_i \text{ down}\}$ and $\{M_u(\beta) \text{ down and } x_\beta = 0 \text{ for any } \beta \in U(i)\}$ are mutually exclusive.

Similarly, $\{M_i \text{ down}\}$ and $\{M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ for some } \beta \in D(i)\}$ are mutually exclusive. Therefore, the probabilities of the events at $t+\delta t$ are additive:

$$p_u(i,m)\delta t = \text{prob } \Big[M_i \text{ down at } t+\delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

$$+ \sum_{\beta \in U(i)} \text{prob } \Big[M_u(\beta) \text{ down and } x_\beta = 0 \text{ at } t+\delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\left.\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t\right]$$

$$+ \sum_{\beta \in D(i), \beta \neq (i,m)} \text{prob} \left[M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ at } t + \delta t \right|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and}$$

$$\left. x_{(i,m)} < N_{(i,m)} \text{ at } t \right]. \tag{19}$$

3.5.1. The First Term of (19)

To analyze the first term of (19), define the mutually exclusive set of events $\{\mathcal{E}_\beta^\alpha\}$. In this notation, α represents a state of Machine M_i (up or down); and β represents any buffer adjacent to Machine M_i except (i,m), or we may have $\beta = *$, which means "no buffer". See Figures 3 – 6. β is a buffer, other than (i,m), which is adjacent to M_i and impeding the flow through it. It is empty if upstream or full if downstream. It would cause M_i to be fully starved or blocked if its other machine were down. Recall that we have assumed that no machine can be simultaneously starved or blocked by more than one buffer (Assumption 3c).

- If $\beta \in U(i)$,

$$\mathcal{E}_\beta^\alpha = \{M_i \text{ in state } \alpha \text{ at } t + \delta t \text{ and}$$

$$\left.\left[x_\beta = 0 \text{ and } x_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up } \right] \text{ at } t\right\}.$$

- If $\beta \in D(i)$, $\beta \neq (i,m)$,

$$\mathcal{E}_\beta^\alpha = \{M_i \text{ in state } \alpha \text{ at } t + \delta t \text{ and}$$

$$\left.\left[x_\beta = N_\beta \text{ and } x_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up } \right] \text{ at } t\right\}.$$

- If $\beta = *$,

$$\mathcal{E}^\alpha_* = \{M_i \text{ in state } \alpha \text{ at } t+\delta t \text{ and}$$

$$[\text{NO upstream buffers are empty and NO downstream buffers are full}$$

$$\left.\text{and } x_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up}\,\right] \text{ at } t\Big\}.$$

Therefore,

$$\left\{\left[x_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up}\,\right] \text{ at } t+\delta t\right\} = \bigcup_{\substack{\alpha = \text{ up, down;} \\ \text{all } \beta}} \mathcal{E}^\alpha_\beta$$

and,

$$\{M_i \text{ down at } t+\delta t\} = \bigcup_{\text{all } \beta} \mathcal{E}^{\text{down}}_\beta$$

The union over β includes $\beta \in U(i)$, $\beta \in D(i)$ (but $\beta \neq (i,m)$), and $\beta = *$.

The term we are analyzing, the first term of (19), is

$$\text{prob}\left[M_i \text{ down at } t+\delta t \right|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\left.\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t\right]$$

$$= \text{prob}\left[M_i \text{ down at } t+\delta t \middle| M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t\right]$$

$$= \text{prob}\left[\bigcup_{\text{all } \beta} \mathcal{E}^{\text{down}}_\beta \middle| \bigcup_{\substack{\alpha = \text{ up, down;} \\ \text{all } \gamma}} \mathcal{E}^\alpha_\gamma\right]$$

which is equal to

$$\sum_{\text{all } \beta} \text{prob}\left[\mathcal{E}^{\text{down}}_\beta \middle| \bigcup_{\text{all } \alpha,\gamma} \mathcal{E}^\alpha_\gamma\right]$$

because the sets $\{\mathcal{E}^\alpha_\beta\}$ are mutually exclusive. As a consequence, the first term of (19) is equal to

$$\sum_{\text{all } \beta} \text{prob } \left[\mathcal{E}_\beta^{\text{down}} \middle| \mathcal{E}_\beta^{\text{up}} \cup \mathcal{E}_\beta^{\text{down}}\right] \text{prob } \left[\mathcal{E}_\beta^{\text{up}} \cup \mathcal{E}_\beta^{\text{down}} \middle| \bigcup_{\text{all } \alpha,\gamma} \mathcal{E}^\alpha_\gamma\right] \tag{20}$$

The first factor of (20) is

$$\text{prob } \left[\mathcal{E}_\beta^{\text{down}} \middle| \mathcal{E}_\beta^{\text{up}} \cup \mathcal{E}_\beta^{\text{down}}\right] =$$

- prob $\Big\{\Big\{M_i$ down at $t+\delta t$ and $\Big[x_\beta = 0$ and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}\Big|$

  $\Big[x_\beta = 0$ and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}$

  $$= p_i \delta t \frac{\mu_u(\beta)}{\mu_i}$$

  if $\beta \in U(i)$.

- prob $\Big\{\Big\{M_i$ down at $t+\delta t$ and $\Big[x_\beta = N_\beta$ and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}\Big|$

  $\Big[x_\beta = N_\beta$ and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}$

  $$= p_i \delta t \frac{\mu_d(\beta)}{\mu_i}$$

  if $\beta \in D(i)$, $\beta \neq (i,m)$.

- prob $\{\{M_i$ down at $t+\delta t$ and [NO upstream buffers are empty and
  NO downstream buffers are full and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}\Big|$
  [NO upstream buffers are empty and

  NO downstream buffers are full and $x_{(i,m)} < N_{(i,m)}$ and $M_u(i,m)$ up $\Big]$ at $t\Big\}$

  $$= p_i \delta t$$

  if $\beta = *$.

We have analyzed the first factor of (20). The second factor is

$$\text{prob}\left[\mathcal{E}_\beta^{\text{up}} \cup \mathcal{E}_\beta^{\text{down}} \Big| \bigcup_{\text{all } \alpha,\gamma} \mathcal{E}_\gamma^\alpha\right]$$

Again, there are three cases. If $\beta \in U(i)$, this is

$$\text{prob}\left[n_\beta = 0 \text{ and } n_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up } \Big| n_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up }\right].$$

Since the numerator is a subset of the denominator, this is

$$\frac{\text{prob}\left[n_\beta = 0 \text{ and } n_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up }\right]}{\text{prob}\left[n_{(i,m)} < N_{(i,m)} \text{ and } M_u(i,m) \text{ up }\right]}. \tag{21}$$

We now show that the event in the numerator is approximately equivalent to the event $\{n_\beta = 0, M_u(\beta) \text{ up}, M_d(\beta) \text{ up }\}$. To analyze the numerator, we again make use of the approximation that M_i can be starved or blocked by at most one machine.

- If $M_u(i,m)$ is up, then M_i must be up. Since $n_\beta = 0$, $M_u(\beta)$ must be up. In addition, M_i up and $n_{(i,m)} < N_{(i,m)}$ implies that $M_d(\beta)$ must be up. Therefore, the event in the numerator implies the event $\{n_\beta = 0, M_u(\beta) \text{ up}, M_d(\beta) \text{ up }\}$.
- If $\{n_\beta = 0, M_u(\beta) \text{ up}, M_d(\beta) \text{ up }\}$, then
  - $n_\beta = 0$.
  - $n_{(i,m)} < N_{(i,m)}$. This is because $n_\beta = 0$ and only one buffer adjacent to M_i can be empty or full.
  - $M_u(i,m)$ is up. This is because $M_d(\beta)$ up implies that M_i is up; and M_i up, $n_\beta = 0$, and $M_u(\beta)$ up imply that $M_u(i,m)$ is up.

The numerator is prob $\{n_\beta = 0, M_u(\beta) \text{ up}, M_d(\beta) \text{ up }\}$ or $\mathbf{p}_\beta(0,1,1)$. It can be evaluated from the two-machine line formulas of Gershwin (1994).

The denominator of (21) is

$$E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)} \tag{22}$$

where $E_u(i,m)$ is the *efficiency* of pseudo-machine $M_u(i,m)$. This the fraction of its maximum production rate that this machine is actually producing. That is, $E_u(i,m) = P(i,m)/\mu_u(i,m)$, in which $P(i,m)$ is the production rate of two-machine line (i,m). Expression (22) follows from Gershwin (1994), Equation (3.224), one of the equations for the production rate of a two-machine line. Thus, (21) is

$$\frac{\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

The second case is where $\beta \in D(i), \beta \neq (i,m)$. The analysis is very similar, and the corresponding term is

$$\frac{\mathbf{p}_\beta(N_\beta,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

The third case is where $\beta = *$, *i.e.*, where Machine M_i is neither starved nor blocked. In that case, the second factor of (20) is

$$1 - \sum_{\beta \in U(i)} \frac{\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

$$- \sum_{\beta \in D(i), \beta \neq (i,m)} \frac{\mathbf{p}_\beta(N_\beta,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

Finally, the first term of (19) is

$$\sum_{\beta \in U(i)} p_i \delta t \frac{\mu_u(\beta)}{\mu_i} \frac{\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

$$+ \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} p_i \delta t \frac{\mu_d(\beta)}{\mu_i} \frac{\mathbf{p}_\beta(N_\beta,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}}$$

$$+ p_i \delta t \left(1 - \sum_{\beta \in U(i)} \frac{\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \right.$$
$$\left. - \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{\mathbf{p}_\beta(N_\beta,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \right)$$

3.5.2. The Rest of (19)

The remainder of (19) is

$$\sum_{\beta \in U(i)} \text{prob} \Big[M_u(\beta) \text{ down and } x_\beta = 0 \text{ at } t + \delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

$$+ \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \text{prob} \Big[M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ at } t + \delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and}$$

$$x_{(i,m)} < N_{(i,m)} \text{ at } t \Big].$$

The events $\{M_u(\beta)$ down and $M_d(\beta)$ down and $x_\beta = 0\}$ and $\{M_u(\beta)$ down and $M_d(\beta)$ down and $x_\beta = N_\beta\}$ are transient in any two-machine line. Therefore, this expression can be written

$$\sum_{\beta \in U(i)} \text{prob} \Big[M_u(\beta) \text{ down and } M_d(\beta) \text{ up and } x_\beta = 0 \text{ at } t + \delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

$$+ \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \text{prob} \Big[M_u(\beta) \text{ up and } M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ at } t + \delta t \Big|$$

$$M_i \text{ up and } \{M_u(\beta) \text{ up or } x_\beta > 0 \text{ for all } \beta \in U(i)\} \text{ and}$$

$$\{M_d(\beta) \text{ up or } x_\beta < N_\beta \text{ for all } \beta \in D(i), \beta \neq (i,m)\} \text{ and}$$

$$x_{(i,m)} < N_{(i,m)} \text{ at } t\Big] .$$

Recall that the conditioning event is $\{M_u(i,m)$ up and $x_{(i,m)} < N_{(i,m)}$ at $t\}$. Thus, we can write this as

$$\sum_{\beta \in U(i)} \text{prob}\, \Big[M_u(\beta) \text{ down and } M_d(\beta) \text{ up and } x_\beta = 0 \text{ at } t+\delta t \Big|$$

$$M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big]$$

$$+ \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \text{prob}\, \Big[M_u(\beta) \text{ up and } M_d(\beta) \text{ down and } x_\beta = N_\beta \text{ at } t+\delta t \Big|$$

$$M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \text{ at } t \Big] . \tag{23}$$

To analyze the upstream ($\beta \in U(i)$) sum, we observe that for the state to be $\{M_u(\beta)$ down and $M_d(\beta)$ up and $x_\beta = 0\}$ at $t + \delta t$, we need only consider, at time t, the following states:

- $\{M_u(\beta)$ down and $M_d(\beta)$ up and $x_\beta = 0\}$. The transition probability from this state is

  $$1 - r_u(\beta)\delta t$$

  because the second machine cannot fail.
- $\{M_u(\beta)$ down and $M_d(\beta)$ up and $0 < x_\beta < \mu_d(\beta)\delta t\}$. The transition probability from this set of states is

  $$(1 - r_u(\beta)\delta t)(1 - p_d(\beta)\delta t)$$

  Here, the second machine can fail.
- $\{M_u(\beta)$ up and $M_d(\beta)$ up and $x_\beta = 0\}$. The transition probability from this state is

  $$p_u(\beta)\delta t \left(1 - p_d(\beta)\delta t \frac{\mu_u(\beta)}{\mu_d(\beta)}\right)$$

The first two cases do not involve either machine changing state, and the last requires only one to change state. Another possible state at time t would be to have both machines down at time t, but $\{M_u(\beta)$ down and $M_d(\beta)$ down and $x_\beta = 0\}$ is transient. All other candidates have probabilities of transition to $\{M_u(\beta)$ down and $M_d(\beta)$ up and $x_\beta = 0\}$ on the order of δt^2.

In the first two cases, since the first machine is up and the second is down, the buffer level could only decrease during $(t, t+\delta t)$. We distinguish between the cases because the expressions for the probabilities are different.

The upstream summation in (23) is therefore

$$\sum_{\beta \in U(i)} \left\{ (1 - r_u(\beta)\delta t) \text{prob } \left[M_u(\beta) \text{ down and } M_d(\beta) \text{ up and } x_\beta = 0 \right| \right.$$

$$\left. M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \right]$$

$$+ (1 - r_u(\beta)\delta t)(1 - p_d(\beta)\delta t) \text{prob } \left[M_u(\beta) \text{ down and } M_d(\beta) \text{ up and } 0 < x_\beta < \mu_d(\beta)\delta t \right|$$

$$\left. M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \right]$$

$$+ p_u(\beta)\delta t \left(1 - p_d(\beta)\delta t \frac{\mu_u(\beta)}{\mu_d(\beta)} \right) \text{prob } \left[M_u(\beta) \text{ up and } M_d(\beta) \text{ up and } x_\beta = 0 \right|$$

$$\left. \left. M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \right] \right\}$$

The first term is 0 because $\{M_u(\beta)$ down and $x_\beta = 0\}$ implies that $M_u(i,m)$ is down.

In both the second and third terms, the conditional event is a subset of the conditioning event. This is because $\{M_d(\beta)$ up$\}$ implies $\{M_i$ up and $x_{(i,m)} < N_{(i,m)}\}$. $\{M_i$ up and $x_\beta > 0\}$ implies $\{M_u(i,m)$ up$\}$ (for the second term). $\{M_u(\beta)$ up and $x_\beta = 0$ and M_i up $\}$ implies $\{M_u(i,m)$ up$\}$ (for the third term). Consequently, this sum can be written (after simplifying the coefficients)

$$\sum_{\beta \in U(i)} \left\{ (1 - r_u(\beta)\delta t - p_d(\beta)\delta t) \frac{\text{prob } [M_u(\beta) \text{ down and } M_d(\beta) \text{ up and } 0 < x_\beta < \mu_d(\beta)\delta t]}{\text{prob } \left[M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \right]} \right.$$

$$\left. + p_u(\beta)\delta t \frac{\text{prob } [M_u(\beta) \text{ up and } M_d(\beta) \text{ up and } x_\beta = 0]}{\text{prob } \left[M_u(i,m) \text{ up and } x_{(i,m)} < N_{(i,m)} \right]} \right\}$$

The denominator of both terms has already been shown to be (22). The numerator of the first term is $f_\beta(0,0,1)\mu_d(\beta)\delta t$ and the numerator of the second term is $\mathbf{p}_\beta(0,1,1)$. The expression is therefore

$$\sum_{\beta\in U(i)} \left\{ (1 - r_u(\beta)\delta t - p_d(\beta)\delta t) \frac{f_\beta(0,0,1)\mu_d(\beta)\delta t}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \right.$$

$$\left. + p_u(\beta)\delta t \frac{\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \right\}$$

which can be further simplified to

$$\sum_{\beta\in U(i)} \frac{f_\beta(0,0,1)\mu_d(\beta) + p_u(\beta)\mathbf{p}_\beta(0,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \delta t.$$

Finally, using equation (3.205) of Gershwin (1994), this becomes

$$\frac{\delta t}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \sum_{\beta\in U(i)} r_u(\beta)\mathbf{p}_\beta(0,0,1).$$

A similar derivation produces the following expression for the downstream sum:

$$\frac{\delta t}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} r_d(\beta)\mathbf{p}_\beta(N_\beta,1,0).$$

3.5.3. Summary: The Complete Interruption of Flow Equations

The upstream equation is

$$p_u(i,m) = \frac{p_i}{\mu_i \left[E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}\right]} \sum_{\beta\in U(i)} \mu_u(\beta)\mathbf{p}_\beta(0,1,1)$$

$$+ \frac{p_i}{\mu_i \left[E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}\right]} \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \mu_d(\beta)\mathbf{p}_\beta(N_\beta,1,1)$$

$$+p_i \left(1 - \frac{\sum_{\beta \in U(i)} \mathbf{p}_\beta(0,1,1) + \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \mathbf{p}_\beta(N_\beta,1,1)}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \right)$$

$$+\frac{1}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \sum_{\beta \in U(i)} r_u(\beta)\mathbf{p}_\beta(0,0,1)$$

$$+\frac{1}{E_u(i,m) - \mathbf{p}_{(i,m)}(N_{(i,m)},1,1)\frac{\mu_d(i,m)}{\mu_u(i,m)}} \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} r_d(\beta)\mathbf{p}_\beta(N_\beta,1,0). \tag{24}$$

The downstream equation is

$$p_d(j,i) = \frac{p_i}{\mu_i \left[E_d(j,i) - \mathbf{p}_{(j,i)}(0,1,1)\frac{\mu_u(j,i)}{\mu_d(j,i)} \right]} \sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} \mu_u(\beta)\mathbf{p}_\beta(0,1,1)$$

$$+\frac{p_i}{\mu_i \left[E_d(j,i) - \mathbf{p}_{(j,i)}(0,1,1)\frac{\mu_u(j,i)}{\mu_d(j,i)} \right]} \sum_{\beta \in D(i)} \mu_d(\beta)\mathbf{p}_\beta(N_\beta,1,1)$$

$$+p_i \left(1 - \frac{\sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} \mathbf{p}_\beta(0,1,1) + \sum_{\beta \in D(i)} \mathbf{p}_\beta(N_\beta,1,1)}{E_d(j,i) - \mathbf{p}_{(j,i)}(0,1,1)\frac{\mu_u(j,i)}{\mu_d(j,i)}} \right)$$

$$+\frac{1}{E_d(j,i) - \mathbf{p}_{(j,i)}(0,1,1)\frac{\mu_u(j,i)}{\mu_d(j,i)}} \sum_{\substack{\beta \in U(i) \\ \beta \neq (j,i)}} r_u(\beta)\mathbf{p}_\beta(0,0,1)$$

$$+\frac{1}{E_d(j,i) - \mathbf{p}_{(j,i)}(0,1,1)\frac{\mu_u(j,i)}{\mu_d(j,i)}} \sum_{\beta \in D(i)} r_d(\beta)\mathbf{p}_\beta(N_\beta,1,0) \tag{25}$$

3.6. Boundary Conditions

For all buffers β such that the machine upstream of β in the original network, $M_{u(\beta)}$, is an input machine, the parameters of $M_u(\beta)$ are those of $M_{u(\beta)}$. That is,

$$\begin{aligned} r_u(\beta) &= r_{u(\beta)} \\ p_u(\beta) &= p_{u(\beta)} \\ \mu_u(\beta) &= \mu_{u(\beta)} \end{aligned}$$

Similarly, for all buffers β such that the machine downstream of β in the original network, $M_{d(\beta)}$, is an output machine, the parameters of $M_d(\beta)$ are those of $M_{d(\beta)}$. That is,

$$\begin{aligned} r_d(\beta) &= r_{d(\beta)} \\ p_d(\beta) &= p_{d(\beta)} \\ \mu_d(\beta) &= \mu_{d(\beta)} \end{aligned}$$

4. Evaluation sequence for simple assembly line

The algorithm described below sweeps back and forth across the decomposed system. The *evaluation sequence* determines the order in which the algorithm visits the two-machine lines. For a flow line, the sequence visits the buffers in the order that parts do, and then reverses this. The sequence is less self-evident for an assembly system. In this section, we describe a sequence for a simple class of systems. Evaluation sequences for more general systems are described in Gershwin (1994).

Simple assembly lines form a special class of A/D systems in which there is no disassembly, and assembly stations can assemble no more than two items. That is, d_i is always 1 and u_i may be 1 or 2. (In fact, there is a main line, with assembly input branches, but this does not affect the equations or algorithm.) An example is in Figure 7.

4.1. General simple assembly lines

Set $b(1)$ = the first buffer in the main line (a in Figure 7). Then number the buffers sequentially until the first assembly station is reached (**B** in Figure 7). The next number is assigned to the first buffer of the new sequence entering the

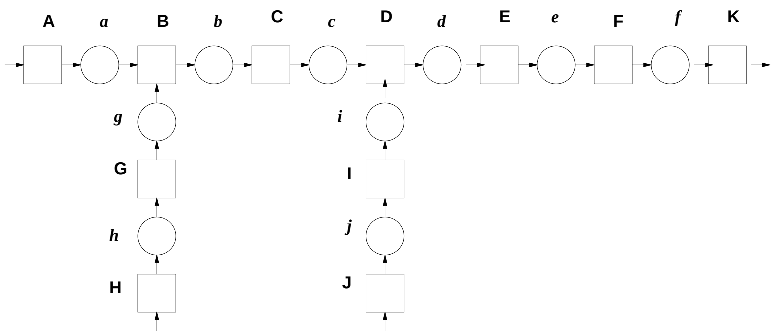

Figure 7. Large Simple Assembly Network

s	1	2	3	4	5	6	7	8	9	10
buffer $b(s)$	a	h	g	b	c	j	i	d	e	f
Δ sequence	$M_u(a)$	$M_u(h)$	$M_u(g)$	$M_u(b)$	$M_u(c)$	$M_u(j)$	$M_u(i)$	$M_u(d)$	$M_u(e)$	$M_u(f)$
Υ sequence	$M_d(f)$	$M_d(e)$	$M_d(d)$	$M_d(i)$	$M_d(j)$	$M_d(c)$	$M_d(b)$	$M_d(g)$	$M_d(h)$	$M_d(a)$

Table 1
Sequences for Figure 7

assembly machine (h). Continue in the new sequence until the assembly machine last visited (**B**) is reached again.

Assign the next number to the first buffer downstream of the assembly machine (b). Proceed downstream and assign numbers sequentially to the buffers as they are encountered. Continue this way until the next assembly machine (**D**) is reached, and continue the process in this way until all buffers are numbered.

The first machines of each two-machine line encountered in this way form the Δ sequence. In the case of a simple assembly network, all the machines in the Δ sequence are upstream machines. The Υ sequence is set of machines encountered first when traversing the set of two-machine lines in the reverse of the Δ sequence. These machines are all downstream machines.

4.2. Example

In Figure 7, there are 11 machines, labeled A, ..., K and 10 buffers, a, ..., b. Machines B and D are assembly machines. The main assembly line is A, a, B, b, C, c, D, d, E, F, f, K. The assembly branches are H, h, G, g, B and J, j, I, i, D. The evaluation sequences are shown in Table 1.

5. A/D-DDX algorithm

The evaluation algorithm is a combination of the A/D algorithm in Section 5.4 of Gershwin (1994) and the ADDX algorithm of Burman (1995) Section 3.2.

1. Initialization
   Set $(\mu_u(i,m), r_u(i,m), p_u(i,m)) = (\mu_d(j,i), r_d(j,i), p_d(j,i)) = (\mu_i, r_i, p_i)$ for all buffers. Determine the evaluation sequence $b(s)$.
2. Δ phase
   Let $s = 2, \cdots k-1$. Evaluate two-machine line $L(s-1)$ using buffer size $N_{b(s-1)}$ and the most recent values of the parameters of machines $M_u(b(s-1))$ and $M_d(b(s-1))$.
   - Determine $p_u(b(s))$ from equation (24).
   - Determine $r_u(b(s))$ from equation (16).
   - Determine $e_u(b(s))$ from equation (8).
   - Determine $\mu_u(b(s))$ from equation (11). Replace P in (11) by $P(b(s-1))$.
3. Υ phase
   Let $s = k-2, \cdots 1$. Evaluate two-machine line $L(s+1)$ using buffer size $N_{b(s+1)}$ and the most recent values of the parameters of machines $M_u(b(s+1))$ and $M_d(b(s+1))$.
   - Determine $p_d(b(s))$ from equation (25).
   - Determine $r_d(b(s))$ from equation (17).
   - Determine $e_d(b(s))$ from equation (8).
   - Determine $\mu_d(b(s))$ from equation (12). Replace P in (12) by $P(b(s+1))$.
4. Termination Condition
   Various possibilities exist, such as, stop when the largest $|P(b(s) - P(b(1))|$ is less than a given value; or the maximum difference between the value of a parameter at the $n+1$st iteration and its value at the nth is less than a threshold.

Selecting the initial values of the parameters automatically satisfies the boundary conditions (Section 3.6). This is because the algorithm never comes back to those machines.

There are other possible ways of replacing P in (11) and (12). See, for example, Gershwin (1994), Section 5.4. Different choices will have different consequences for convergence speed and reliability. What is important is that different expressions be used, so that (11) and (12) together imply conservation of flow. See Dallery, David, and Xie (1988).

6. Extended A/D-DDX algorithm

In the algorithm above, the quantities $p_u(b(s))$, $r_u(b(s))$, $\mu_u(b(s))$, $p_d(b(s))$, $r_d(b(s))$, and $\mu_d(b(s))$ appear on both sides of the equations. The previous values are used on the right sides, and the new values are determined. Burman (1995) and Burman and Gershwin (1997) sped up the flow-line version of the algorithm by solving the equations for the quantities to be determined. They no longer appear on the right side, and consequently, fewer iterations are needed for convergence. We did that here as well in obtaining the numerical results below.

For example, (16) can be written

$$r_u(i,m) = \frac{r_i}{1 + \left\{ \begin{array}{l} r_i \left(1 - \displaystyle\sum_{\beta \in U(i)} \frac{\mathbf{p}_\beta(0,0,1)}{p_u(i,m)E_u(i,m)} - \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{\mathbf{p}_\beta(N_\beta,1,0)}{p_u(i,m)E_u(i,m)} \right) \\ \quad + \displaystyle\sum_{\beta \in U(i)} \frac{\mathbf{p}_\beta(0,0,1) r_u(\beta)}{p_u(i,m)E_u(i,m)} + \sum_{\substack{\beta \in D(i) \\ \beta \neq (i,m)}} \frac{\mathbf{p}_\beta(N_\beta,1,0) r_d(\beta)}{p_u(i,m)E_u(i,m)} \end{array} \right\}}$$

In this form of the equation, $r_u(i,m)$ appears only on the left side. Similar transformations can be done for the other quantities.

7. Numerical Results

Figure 8 shows an assembly system with eleven machines and ten buffers. The parameters of this system are listed in Table 2. Each line of that table shows the parameters of a machine and the capacity of the immediately downstream buffer. We also show the average buffer levels for one case. We vary the size of Buffer $(M2, M3)$ which is 15 for the case in the table. (All other buffer sizes are 20.)

Machine	r	p	μ	N	$\bar{n}$
M1	0.10	0.01	0.90	20.00	14.855695
M2	0.10	0.01	0.90	15.00	8.790485
M2C	0.10	0.01	1.00	20.00	16.774604
M2B	0.10	0.01	1.00	20.00	16.648199
M2A	0.10	0.01	1.00	20.00	17.045951
M3	0.10	0.01	0.90	20.00	8.467701
M3A	0.10	0.01	1.00	20.00	15.212316
M4	0.10	0.01	0.90	20.00	5.144305
M4B	0.10	0.01	1.00	20.00	15.484635
M4A	0.10	0.01	1.00	20.00	13.306918
M5	0.10	0.01	0.90		

Table 2

System Parameters and Mean Buffer Levels

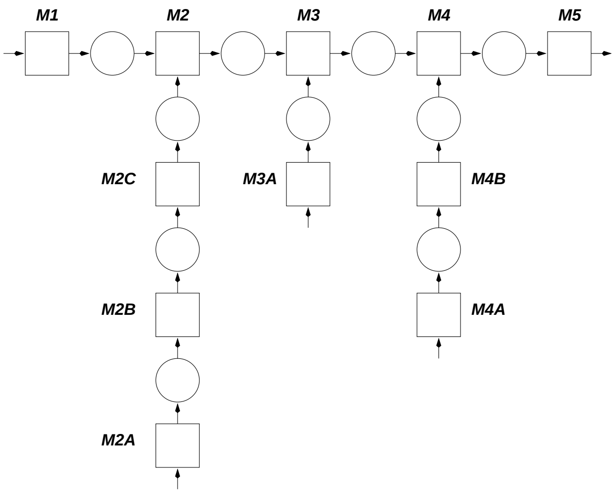

Figure 8. Example Network

Figures 9 and 10 show how the production rate P and the average amounts of material in the buffers on the main line vary. Also indicated in these graphs are the results of simulations. Averages and 95% confidence intervals are shown.

The production rate appears to increase monotonically, and approaches a limit. As the size of Buffer $(M2, M3)$ increases, its average level increases. This suggests that the part of the system upstream of this buffer is faster than the part of the line downstream of it. In fact, if we analyze the portion of the system upstream of Buffer $(M2, M3)$ separately from the portion downstream, we find that the production rate of the upstream portion of the line is 0.773 and that of

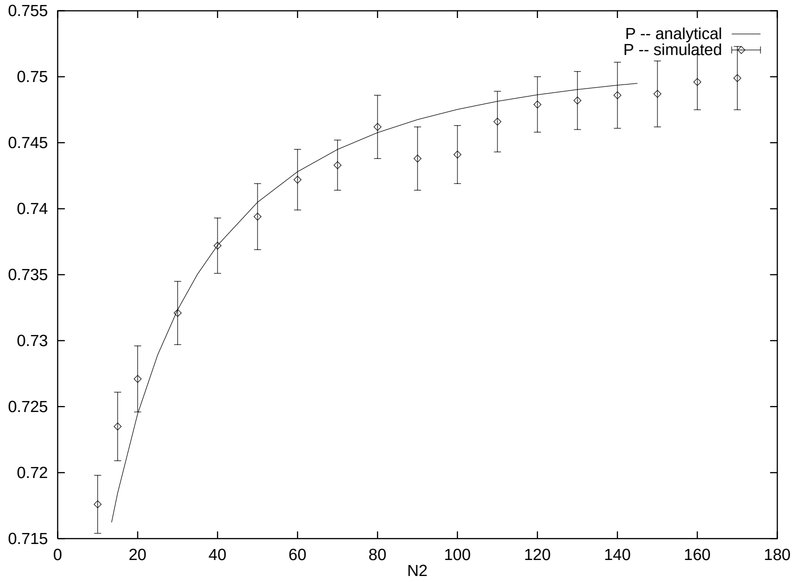

Figure 9. P vs. N_2

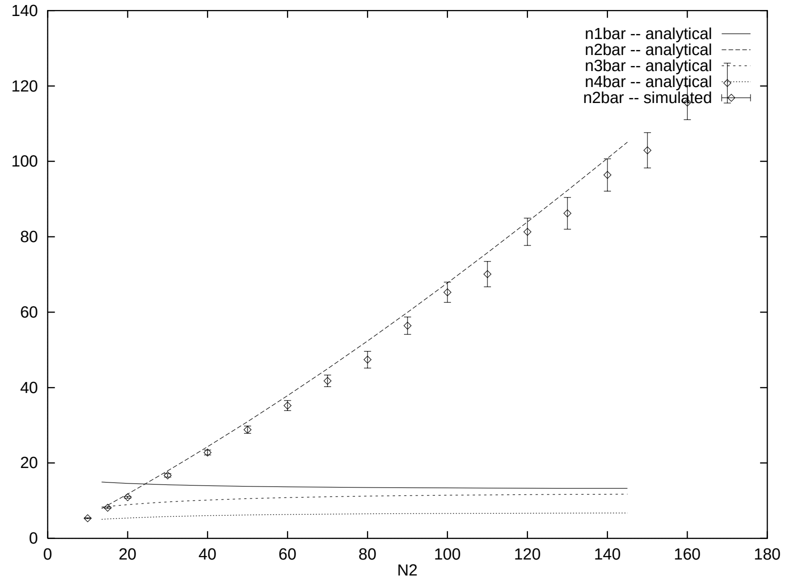

Figure 10. $\bar{n}_i$ vs. N_2, $i = 1, 2, 3, 4$

the downstream portion is 0.758.

The level of Buffer $(M1, M2)$ decreases as the size of Buffer $(M2, M3)$ increases because increasing Buffer $(M2, M3)$ makes the portion of the line downstream of Buffer $(M1, M2)$ faster. The levels of the other buffers examined increase as Buffer $(M2, M3)$ increases because the portion of the line upstream of them is becoming more productive.

Other numerical results Burman (1995) describes extensive and systematic testing of the special case of this method that was restricted to tandem production lines. He found that the method was accurate, that it converged reliably, and that it was fast. We believe that there is no reason for these properties not to persist in the more general case described here.

8. Conclusion

We have described a decomposition method for tree-structured assembly/disassembly manufacturing systems in which machines operate at different speeds. This method is accurate and fast enough to be included in an optimization technique. Future research should be devoted to optimization, to extensions to networks with loops, and to networks with multiple part types.

Acknowledgments

We thank Augusto Giancola for programming the decomposition, and Andreas Klamm for writing the simulation.

References

[1] T. Altiok (1997), *Performance Analysis of Manufacturing Systems*, Springer Verlag, New York, 1997.

[2] R. G. Askin and C. R. Standridge (1993), *Modeling and Analysis of Manufacturing Systems*, John Wiley & Sons, 1993.

[3] P. Awate and B. Sastry (1987), "Analysis and decomposition of transfer and flow lines," *Operations Research*, 24:175–196, 1987.

[4] M. H. Burman (1995) "New Results in Flow Line Analysis," MIT OR Center Ph. D. Thesis, June, 1995.

[5] M. H. Burman, S. B. Gershwin, and C. Suyematsu (1998), "Hewlett-Packard Uses Operations Research to Improve the Design of a Printer Production Line," *Interfaces*, Volume 28, Number 1, January-February, 1998, pp. 24-36.

[6] M. H. Burman and S. B. Gershwin (1997), "Analysis of Continuous Material Models of Unreliable Flow Lines," MIT, 1997.

[7] G. Buxey, N. Slack, and R. Wild (1973), "Production flow line system design — A review," *AIIE Transactions*, 5:37–48, 1973.

[8] J. Buzacott (1967b) *Markov Chain Analysis of Automatic Transfer Line with Buffer Stock*, PhD thesis, University of Birmingham, 1967.

[9] J. Buzacott (1972), "The effect of station breakdowns and random processing times on the capacity of flow lines," *AIIE Transactions*, 4:308–312, 1972.

[10] J. Buzacott and L. Hanifin (1978), "Models of automatic transfer lines with inventory banks — A review and comparison," *IIE Transactions*, 10:197–207, 1978.

[11] Buzacott, J. A. and Shanthikumar, J. G. 1993, *Stochastic Models of Manufacturing Systems*, Prentice Hall, Englewood Cliffs, New Jersey.

[12] Y. F. Choong and S. B. Gershwin (1987), "A Decomposition Method for the Approximate Evaluation of Capacitated Transfer Lines with Unreliable Machines and Random Processing Times," *IIE Transactions*, Vol. 19, No. 2, pp. 150-159.

[13] Y. Dallery, R. David, and X.-L. Xie (1988), "An Efficient Algorithm for Analysis of Transfer Lines with Unreliable Machines and Finite Buffers," *IIE Transactions,* Vol. 20, No. 3, pp. 280-283, September, 1988.

[14] Y. Dallery, R. David, and X.-L. Xie (1989), "Approximate Analysis of Transfer Lines with Unreliable Machines and Finite Buffers," *IEEE Transactions on Automatic Control,* Vol. 34, No. 9, pp. 943-953, September, 1989.

[15] Y. Dallery and S. Gershwin (1992), "Manufacturing flow lines systems: A review of models and analytic results," *Queueing Systems: Theory and Applications*, 12(1-2):3–94, December 1992.

[16] M. Di Mascolo, R. David, and Y. Dallery (1991), "Modeling and Analysis of Assembly Systems with Unreliable Machines and Finite Buffers," *IIE Transactions*, Vol. 23, N0. 4, December 1991, pp. 302-314.

[17] S. Gershwin (1987), "An efficient decomposition algorithm for the approximate evaluation of tandem queues with finite storage space and blocking," *Operations Research*, 35:291–305, 1987.

[18] S. Gershwin (1989), "An efficient decomposition algorithm for unreliable tandem queueing systems with finite buffers," in H. G. Perros and T. Altiok, editors, *Queueing Networks with Blocking*, pages 127–146. North Holland, Amsterdam, 1989.

[19] S. B. Gershwin (1994), *Manufacturing Systems Engineering,* Prentice-Hall, 1994.

[20] S. Gershwin and O. Berman (1981), "Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers," *AIIE Transactions*, 13, 1981.

[21] S. B. Gershwin and I. C. Schick (1980), "Continuous model of an unreliable two-stage material flow system with a finite interstage buffer," Technical Report LIDS-R-1039, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1980.

[22] C. Glassey and Y. Hong (1993), "The analysis behavior of an unreliable n-stage automatic transfer line with $(n-1)$ inter-stage buffer storages," *International Journal of Production research*, 31(3):519–530, 1993.

[23] Y. Hong and D. Seong (1989), "The analysis of an unreliable n-machine flow-line manu-

facturing system with random processing times," Pohang Institute of Technology, Dept. of Industrial Engineering, 1989. Technical Report 89-03.

[24] H. Le Bihan (1998), "De Nouvélles Methodes Analytiques pour l'Évaluation des Performances de Lignes de Production," Thèse de doctorat de l'Université Paris 6, September, 1998.

[25] X.-G. Liu and J. A. Buzacott (1990), "Approximate Models of Assembly Systems with Finite Inventory Banks," *European Journal of Operational Research*, Vol. 45, pp 143-154, 1990.

[26] H. T. Papadopoulos, C. Heavey, and J. Brown (1993), *Queueing Theory in Manufacturing Systems Analysis and Design*, Chapman & Hall, 1993.

[27] H. G. Perros (1986), "Queueing networks with blocking: A bibliography," *Performance Evaluations, Rev. 12*, 8-14, 1986.

[28] A. Semery (1987), "Modelisation des lif: Integration du modele de dubois et forestier a l'heuristique de gershwin, pour les lignes a temps de cycle differents," *Rapport de recherche AS/619-87*, 1987.

[29] T. Smunt and W. Perkins (1985), "Stochastic unpaced line design: Review and further experimental results," *Journal of Operations Management*, 5:351–373, 1985.