Multiple Part Type Decomposition Method in Manufacturing Processing Line

By

Young Jae Jang

B.S., Aerospace Engineering
Boston University, 1997

Submitted to the Department of Mechanical Engineering
and to the Sloan School of Management
in Partial Fulfillment of the Requirements for the Degrees of

Master of Science in Mechanical Engineering
and
Masters of Science in Operations Research
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2001

Signature of Author ....
Department of Mechanical Engineering
May 15th, 2001

Certified by ....
Stanley B. Gershwin
Senior Research Scientist, Department of Mechanical Engineering
Thesis Supervisor

Accepted by ....
Ain Sonin
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Accepted by ....
Cynthia Barnhart
Co-director, Operations Research Center

Multiple Part Type Decomposition Method in Manufacturing Processing Line

by

Young Jae Jang

Submitted to the Department of Mechanical Engineering
and
the Sloan School of Management
on May 15th, 2001, in partial fulfillment of the
requirements for the degrees of
Master of Science in Mechanical Engineering
and
Master of Science in Operations Research

Abstract

This dissertation presents an analysis of a synchronous transfer line (one in which all production times are identical) that produces multiple different part types on unreliable machines. The rates of failure and repair of the machines need not be identical. Inventory is stored between machines in finite buffers, or in work areas in the machines themselves, and different buffers may have different sizes. The machines operate according to a *static priority rule*, operating on the highest priority part whenever possible, and only operating on lower priority parts when unable to produce those with higher priorities due to either blockage or starvation. In order to analyze this system, we introduces a new method of decomposition - party type decomposition; decomposing two part type line into distinct two single part type line. An iterative algorithm is employed to solve the resulting equations of the decomposed line. Estimates for performance measures, such as average buffer levels, throughput, and proportion of time due to blockage and starvation are presented, and compared to extensive discrete event simulation.

Thesis Supervisor: Stanley B. Gershwin
Title: Senior Research Scientist, Department of Mechanical Engineering

Acknowledgments

First of all, I want to thank to my thesis advisor Stanley B. Gershwin for his encouragement during the last two years. I would not have survived my research without his support. His kindness and consideration for me as well as his graduate students shatters the stereotyped image of the distant MIT professor.

I would like to thank my fellow group members at my office: Nicola Maggio, Meow Shean Young, and Loren Warner. They always encouraged me when I faced difficulties.

Thanks are also due to my friends at the Korean Catholic Church at Boston. Without their friendship, I could not have survived my life as a graduate student.

Finally, I would like to thank my parents, grandmother, and God.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The design, operation, and analysis of manufacturing systems are of great economic importance. Most of the current work applied in industry relies on computer simulation of the stochastic processes of the production flow line. However, simulations require a considerable time to construct and run. Work done by Gershwin suggest that by using docomposition method he developed, it is possible to construct a closed formulation of production line under various operational assumptions. Current formulations of decomposition methods include those with single-class single-failure mode lines, single-class, multi-failure mode lines. These formulations usually yield solutions that approximate the solutions of the simulation without the required computational power and time. However all of the decomposition methods done assume that there is only one part type. The contribution of this work presented in this paper will be to extend the method described above to a processing line with more than one part type. Since most manufacturing systems consists of machines that do manufacturing more than one part type of product, it is essential that models be developed to reflect this. The challenge of formulating the multi-class line is not completely new. Nemec tried to formulate a deterministic single-failure multi-class transfer line. However, this formulation only worked for small two-class type lines. The work presented in this paper will propose a new way of decomposion method - part type decompostion that decomposes the multiple part type line into single party type lines. This decrease the complexity of the line and make the line easier to evaluate the behavior of the line. While the methods presented appear to be generalizable to more than two part types, we restrict our attention to only two in order to make precise all of the issue involved in the decomposition.

1.2 Literature Review

Single-Part Type Transfer Lines with Single-Failure-Modes

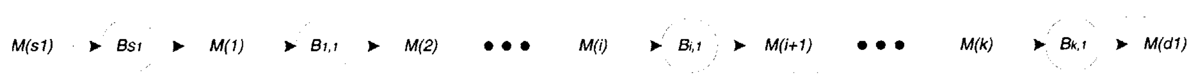

Figure 1-1: A Single Part Type Line

The manufacturing process line is a production system whose work proceeds in a linear fashion from one machine to the next. A single-part type line is one in which the manufacturing process line only builds one type of part. An example of a single-part type line is shown in Figure (1-1). A flow line has machines (M) which perform some work in a part, and such are depicted by the squares in the figure. Parts flow from machines into buffers (B), or storage centers, which are depicted by circles. The arrows that connect the machines and buffers represent the path of work-in-process, and the direction is from left to right.

One way to analyze flow lines is to break them into simpler structures, specifically,

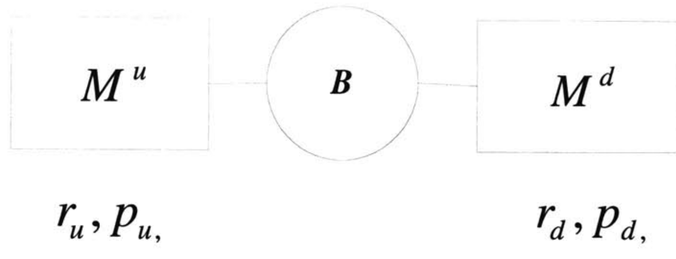

Figure 1-2: Two Machine Line

two-machine-lines. This is the technique called *decomposition*. Once a formulation and solution to the two-machine-line is found, it may be possible to find an approximate solution to the complete flow line. A two-machine-line is depicted in Figure 1-2. In order to solve the two-machine-line, it is necessary to have a behavior assumption and a representation of the machines and the production process. The representation requires the size of the buffer (N), the failure rate of the machines (p), the repair rates (r), and the processing rates (μ).

The simplest characterization of the production flow is the deterministic model. Under a deterministic assumption, the processing rates of all machines are constant. A machine processes one part, in one time unit, asynchronously from other machines. In addition, a machine cannot process a part if it is starved (there is no available material in the buffer preceding it), or it is blocked (there is no space in the buffer receiving parts from the machine). Generally, a machine is not allowed to fail unless it is working on a part. In addition, in a two-machine-line, the up stream machine is never starved (there is always raw material), and the down stream machine is never blocked (there is always space to put completed parts). The formulation and solution of the resulting deterministic two-machine-line is achieved by solving a two-dimensional Markov chain with $4(N-1)$ states [1]. The solution to such a chain

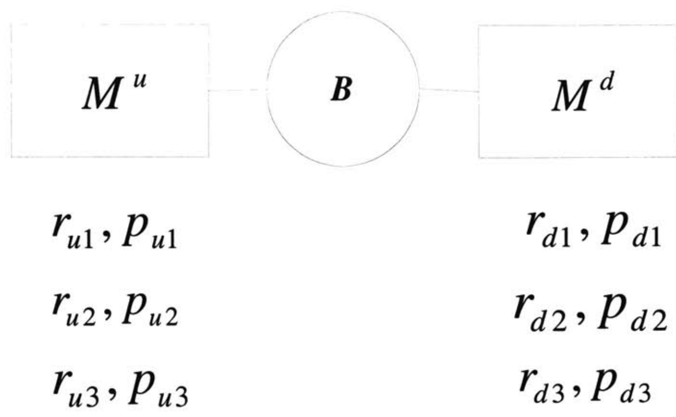

Figure 1-3: Singe-Part-Type, Multiple-Failure-Mode Two-Machine-Line

is given as the steady state probability of all states, the line's buffer levels, and the overall production rate.

Through other types of assumptions and solution techniques, other process behaviors can be captured. For example, using a continuous flow assumption, it is possible to allow for machines to have different processing rates.

Single-Part Type Transfer Lines with Multiple-Failure-Modes

The transfer line models discussed above assume that machines may fail only in one way. Current work done by Tolio [3] allows for a similar formulation of production lines with the added feature that a given machine may fail in one of several modes, and be repaired in the mode corresponding to the specific failure mode. Thus, for example, a machine may fail because a part got stuck, and take an average of 5 minutes to repair, or because the motor exploded, and take an average of 5 days to repair. A two-machine-line building block representation is depicted in Figure 1-3.

Another feature of multiple failure lines is that in the decomposition process, two-machine-lines can assign failure modes to account for the probability of starvation and blockage due to failures of machines outside of the two-machine-line. These failure modes are called *virtual failure modes* as they are not *real failure modes.* For every two-machine-line, behavior paramenters are analyzed and changed in an ordered way until convergence is achieved. By allowing a two-machine-line to account directly for new possibilities of failure, the accuracy of the solution is usually improved.

Multiple-Part Type Transfer Lines with Single-Failure-modes

Work conducted by Nemec [2] formulated and solved for deterministic behavior lines that processed more than one part type. Nemec's formulation only works for small two-class type lines. The reason why the formulation works only in a limited set of lines is that he considered every single detail of behavior of the line that his decomposition equations were too complicate to be solved analytically.

1.3 Thesis Outline

This dissertation is organized as follows. Chapter 2 introduces a Markov model of a processing line with two different part types, flexible, unreliable machines, and finite intermediate buffers. The decomposition of the long line into smaller, tractable two-machine lines is discussed. Chapter 3 presents the analysis of the Markov chain for the two-machine lines. The decomposition derivation is derived in Chapter 4 and 5. An algorithm to solve the decomposition is presented in Chapter 6, as are numerical results concerning the accuracy of the decomposition, and the qualitative behavior of the system.

Chapter 2

Multi-Part Type Processing Lines

2.1 Introduction

In this chapter we introduce a Markov chain model of a two-part type line. The purpose of developing the model is to predict the performance of the line: the production rate of each part type and average buffer levels. Here, we describe the model assumptions and notation. We also describe how we use the model to obtain the performance measures. Most of the assumptions and notations describing the system follow the convention set by Gershwin [1] and Nemec [2].

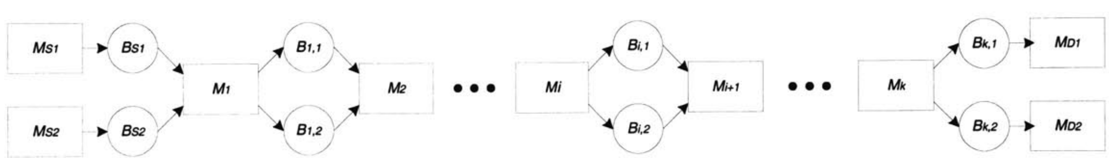

Figure 2-1: A two-part type flow line

2.2 Two Part Type Processing Line

Figure 2-1 represents the transfer line processing two different part types. The line consists of two kinds of components: processing machines (M_i), denoted by the squares; and finite-capacity storage (B_i) for work in process inventory, denoted by the circles. At the beginning and end of the line, there are supply machines: (M_{S1}), and (M_{S2}), and demand machines: (M_{D1}), and (M_{D2}).

Machines M_{S1} and M_{D1} process only Type 1 parts, while Machines M_{S2}, and M_{D2} process only Type 2 parts. Each machine between the supply and demand machines can process both part types. We assume that there is no set-up time incurred when the machine switches production from one part type to another. When machine M_i completes work on a part, it sends the part to a buffer downstream of the machine. Each part type has a distinct buffer after each machine. Therefore, a Type 1 part processed at machine M_i would be sent to the buffer B_{i1}. A Type 2 part processed at the same machine would be sent to buffer B_{i2}.

2.2.1 Processing Machine

We assume that all the machines in the line, including supply and demand machines, are unreliable. Let α denote the state of of a machine. If $\alpha = 1$, the machine is said to be *up* or *working*. If $\alpha = 0$, the machine is said to be *down* or *failed*.

We let $\alpha_{S1}(t)$ denote the state of supply machine S_1 at the end of time step t We define $\alpha_{S2}(t)$ similarly for S_2. For the demand machine, D_1, and D_2, we let the corresponding state variables be $\alpha_{D1}(t)$ and $\alpha_{D2}(t)$. For processing machine M_i, the state variable representing the state of the machine at the end of time step t is written $\alpha_i(t)$.

We make the assumption that all the machines in the line, including the supply and demand machines, have *homogeneous processing times* — that is, the lengths of time that parts spend in machines are fixed, known in advanced, and the same for all the machines. Therefore, the processing times are assumed to be scaled to unity. Furthermore, we assume that the yield of all machines is 100%. That is, we do not allow for the scrapping or rework of parts.

2.2.2 Buffers

We assume that all buffers, the including supply and demand buffers, have finite size. The size of buffer (i, j) is denoted $N_{i,j}$, where i indicates the production stage, and $j = 1$ or 2, represents the part type. We let buffers $S1$ and $S2$ denote the supply buffers for Type 1 and Type 2, respectively. Likewise, buffers $D1$ and $D2$ denote the demand buffers for Type 1 and Type 2, respectively. We denote the current level of buffer (i, j) at the end of time step t by $n_{i,j}(t)$. Therefore, $0 \leq n_{i,j}(t) \leq N_{i,j}$, for all (i, j), and for all $t \geq 0$. A machine is said to be *starved* for a given part type if the upstream buffer corresponding to that part type is empty. It is *blocked* for a given part type if the corresponding downstream buffer is full. We make the assumptions that the supply machines are never starved and the demand machines are never blocked.

2.2.3 Machine Parameters

As mentioned earlier, all machines in the line are assumed to be unreliable: either up or down. We further assume that all machines, including the machines representing supply and demand cannot fail if they are idle. This is called *operation dependent failures.* This means that the supply machines cannot fail if they are blocked. This also means that the demand machines cannot fail if they are starved. A processing machine cannot fail if it is either starved or blocked for Type 1 parts, and at the same time starved or blocked for Type 2 parts.

All machines are assumed to have geometrically distributed up and down times.

We assume that the probability that processing machine M_i fails is the same, regardless of the part type the machine is working on or the history of the system. We let r_i represent the probability that machine M_i is up in time step $t+1$, given it was down in time step t. Likewise, p_i represents the probability that machine M_i is down in time step $t+1$, given it was up and not blocked or starved in time step t. For the supply machines, we let r_{S1} and r_{S2} represent the probability that machine M_{S1} and M_{S2} are up in time step $t+1$, given they were down in time step t, respectively. Also, p_{S1} and p_{S2} represent the probability that machine S_1 and S_2 are down in time step $t+1$, given they were up *and* not blocked in time step t. For the demand machines D_1 and D_2, the corresponding parameters are written r_{D1}, p_{D1}, r_{D2}, and p_{D2}. Let us define k be the number of machines that are processing two different part types in the line, not including the supply and demand machines. For the processing machine M_i, this can be written as:

$$\begin{aligned} r_i &= Pr\left[\alpha_i(t+1)=1|\alpha_i(t)=0\right] \qquad (2.1)\\ p_i &= Pr\left[\alpha_{i,1}(t+1)=0|\{\alpha_{i,1}(t)=1 \cap n_{i-1,1}(t)>0 \cap n_{i,1}(t)<N_{i,1}\} \cup \right.\\ &\qquad \{\alpha_{i,1}(t)=1 \cap (n_{i-1,1}(t)=0 \cup n_{i,1}(t)=N_{i,1})\\ &\qquad \left.\cap\, n_{i-1,2}(t)>0 \cap n_{i,2}(t)<N_{i,2}\}\right]. \end{aligned}$$

for $i=1,\ldots,k$

Likewise, for the supply and demand machines,

$$r_{S1} = Pr\left[\alpha_{S1}(t+1)=1|\alpha_{S1}(t)=0\right] \qquad (2.2)$$

$$
\begin{aligned}
r_{S2} &= Pr\left[\alpha_{S2}(t+1) = 1 | \alpha_{S2}(t) = 0\right] \\
p_{S1} &= Pr\left[\alpha_{S1}(t+1) = 0 | \alpha_{S1}(t) = 1 \cap n_{S1}(t) < N_{S1}\right] \\
p_{S2} &= Pr\left[\alpha_{S2}(t+1) = 0 | \alpha_{S2}(t) = 1 \cap n_{S2}(t) < N_{S2}\right] \\
r_{D1} &= Pr\left[\alpha_{D1}(t+1) = 1 | \alpha_{D1}(t) = 0\right] \\
r_{D2} &= Pr\left[\alpha_{D2}(t+1) = 1 | \alpha_{D2}(t) = 0\right] \\
p_{D2} &= Pr\left[\alpha_{D1}(t+1) = 0 | \alpha_{D1}(t) = 1 \cap n_{D1}(t) > 0\right] \\
p_{D2} &= Pr\left[\alpha_{D2}(t+1) = 0 | \alpha_{D2}(t) = 1 \cap n_{D2}(t) > 0\right]
\end{aligned}
$$

2.2.4 Part Type Priority Policy

Since each machine in the production line now must choose which part to work on when it has a choice, we are required to state a policy by which that choice is made. Our assumption is that each machine will work on Type 1 whenever the machine is up, the upstream buffer for Type 1 is not empty, and the downstream buffer for Type 1 is not full. Each machine will only work on a Type 2 part if it is up, and either blocked or starved for Type 2 parts, and not starved and blocked for Type 2 parts.

2.2.5 Efficiency

Let us denote the efficiency of Type 1 part at machine i by $E(i,1)$. This is the fraction of time that M_i is working on Type 1 parts. We know that machine M_i will make a Type 1 part at the end of time $t+1$ if M_i is not starved for type one at time t, M_i is not blocked for type one at time t, and M_i is up at the end of time $t+1$.

This probability is expressed as follows:

$$E(i,1) = Pr\left[\alpha_i(t+1) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t) < N_{i,1}\right] \tag{2.3}$$

Let the quantity $E(i,2)$ denote the efficiency of Type 2 parts. This is the fraction of time that M_i is working on Type 1 parts. From our assumptions, we know that machine M_i will make a Type 2 part at time $t+1$ if M_i is either blocked or starved for type one at time t; M_i is not starved for type two at time t; M_i is not blocked for Type 2 parts at time t; and M_i is up at the end of time $t+1$. This is expressed as follows:

$$\begin{aligned} E(i,2) = \; & Pr\left[\alpha_i(t+1) = 1 \cap (n_{i-1,1}(t) = 0 \cup n_{i,1}(t) = N_{i,1}) \right. \\ & \left. \cap\, n_{i-1,2}(t) > 0 \cap n_{i,2}(t) < N_{i,2}\right] \end{aligned} \tag{2.4}$$

In steady state, because of conservation of flow [?], we require that each machine in the line makes the same numbers of type one and type two part. If we denote the throughput for the demand machine for part type j by E_{D_j}, and the supply machine for part type j by E_{S_j}, then we must have

$$E_{S_j} = E(i,1) = E(i,2) = \ldots = E(i,k) = E_{D_j},$$

for $j = 1, 2$.

2.3 Part Type Decomposition

As described in Chapter 1, we are able to analyze a single part type line. Now we want to analyze the line that processes more than one part type. To do this we decompose the system into single part type lines first, and then analyze each single part type separately. We call this procedure *part type decomposition.* This decomposition procedure is represented in Figure 2-2.

We suppose that there are two observers in each machine in the real line. One observer watches the flow of only Type 1 parts. We call him the *Type 1 observer.* On the other hand, the second observer watches the flow of only type two parts. This is the *Type 2 observer.* Since each observer watches only one part type, the observers are unable to tell that they are in the system that is processing two different part types. Since Type 1 parts have priority over Type 2 parts, each part type observer sees that this machine is processing a part whenever there is a part available in the immediate upstream buffer and there is space to put a processed part in the immediate downstream – The part flow he observes would be very similar to what he would observe if he were observing a single part type line.

However, there is a special behavior that the Type 1 observers would notice in the two part type line, that they would not experience if he were in a single part type line. Suppose that the Type 1 observer does not see any inflow of parts due to starvation: there has been an actual machine failure in the upstream part of the line, starving the line downstream of it. The observer would think that his machine is idle because he can only see Type 1 parts. However, the actual machine may not be idle; instead it may be working on the second part type. While the actual machine is working on the second part type, it might fail. From the Type 1 observer's perspective, the machine has failed while it was idle. We call this *idleness failure.*

This idleness failure is a behavior that can be observed only by Type 1 observer in a multi-part type line. Conventionally, in a single part type line, we assume that both the real machines and the pseudo-machines in the two machine lines of the decomposition have operation dependent failures. Therefore, we must relax that assumption for the two machine line in the multiple part type case. We present a discrete-time, discrete-state Markov model of precisely such a line in Chapter 3.

Now, suppose that we take the point of view of second part type observer. We similarly misinform this observer: we lead him to believe that he is watching the flow of a single part type line in his machine. The machine may stop processing the second part type for three reasons: the machine is actually down, the machine is processing part type one, or the machine is starved or blocked for the part type two. However, since the observer sees only the second part type, he believes that the machine is down when the machine is processing the first part type.

We must therefore choose the parameters of the machine in the second part type line so that judging by the behavior of the machine, the observer of the machine is unable to tell whether or not the machine is processing one or two part types. We present the equations for the parameters of the second part type line in Chapter ??.

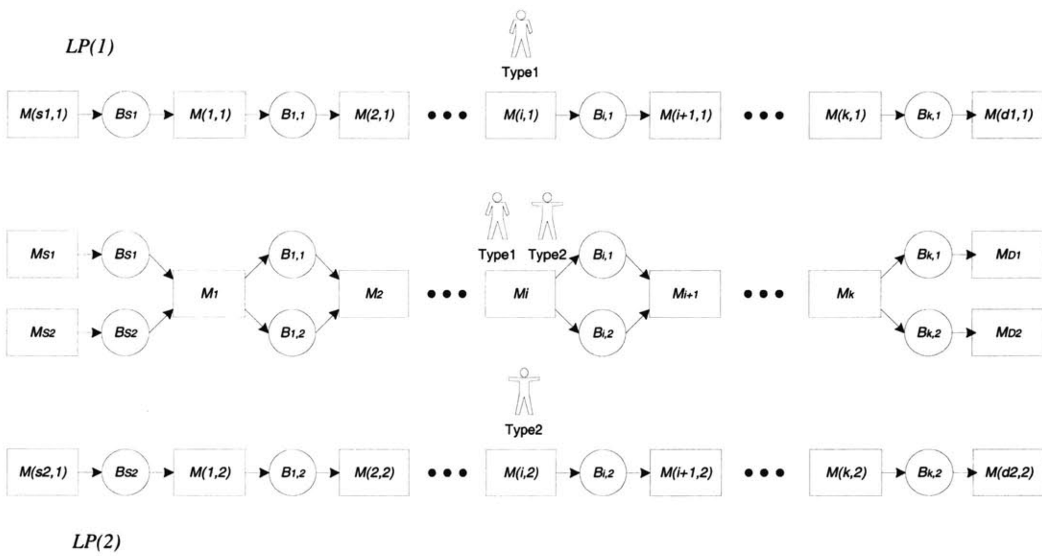

Figure 2-2: Part Type Decomposition

Chapter 3

Two-Machine Line Model with Idleness Failure

3.1 Introduction

This chapter presents the analysis of a Markov chain model of the two-machine transfer processing line. As discussed in Section ??, in order to decompose the Markov chain model of the two-part type processing line, we need a new two-machine line. The two-machine line presented here is similar to the deterministic processing time model described by Gershwin [1]. However, in our model, the machines in the two-machine line are no longer restricted to failing only if they are not blocked or starved. We eliminate the assumption that machines in the line can fail only when they are operating on a part. Since a machine in the line can fail while it is idle - staved or blocked, we call the line, *a two-machine line with idleness failure.* Nemec [2] modeled the two-machine line with idleness failure and we follow his notation and assumptions.

3.2 Model assumptions, and conventions

Figure 3-1 illustrates the two-machine line. We denote the upstream machine by M_u and the downstream machine by M_d. The size of the buffer is N, and the level of the buffer at time t is $n(t)$. Therefore, it follows that $0 \leq n(t) \leq N$, for all t.

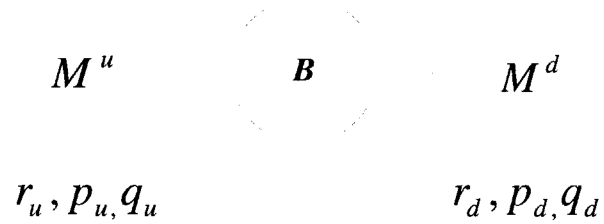

Figure 3-1: A two-machine transfer line with idleness failures.

Material flows from outside the system to the upstream machine to the buffer to the downstream machine and then out of the system. We assume that only one part type is produced in the line and the production time at each of the machine is identical and equal to 1.

We assume that the machines can fail while they are either operating on a part or while they are idle, but we do not assume that the probabilities of failure are identical. In particular, we assume that the probability that M_u fails while it is working on a part, given it is not blocked, is p_u, and the probability that it fails while it is blocked is q_u. We define the quantities p_d and q_d for M_d similarly. Finally, we denote the probabilities that M_u and M_d are repaired while they are down by r_u and r_d, respectively. We call a failure that takes place while the machine is operating on a part an *operational failure*, and a failure that occurs when the machine is idle an *idleness failure.*

Finally, we define the state of the two-machine line to be $s = (n, \alpha_u, \alpha_d)$, where n

is the level of the intermediate buffer ($0 \leq n \leq N$), α_u is 0 if M_u is down, and 1 if M_u is up, and α_d is 0 if M_d is down, and 1 if M_d is up.

3.3 Transition Equations

In this section we define the transition equations for the Markov model. Nemec originally modeled the equations and therefore, we review his model in this section.

3.3.1 Transient States

The assumptions of the model imply that certain states are transient; that is, they have zero steady state probability. Transient states cannot be reached from any state except possibly other transient states. The following states are transient:

- (0,1,0) is transient because it cannot be reached from any other state. If $\alpha_u(t+1) = 1$ and $\alpha_d(t+1) = 0$, then $n(t+1) = n(t) + 1$.

- (0,1,1) is transient because it cannot be reached from any other state. If $n(t) = 0$ and $\alpha_u(t+1) = 1$ and $\alpha_d(t+1) = 1$, then $n(t+1) = 1$ since M_d is starved and thus not able to operate. If $n(t) > 0$ and $\alpha_u(t+1) = 1$ and $\alpha_d(t+1) = 1$, then $n(t+1) = n(t)$.

- (N,0,1) is transient. If $\alpha_d(t+1) = 1$ and $\alpha_u(t+1) = 0$, and $n(t) > 0$, then M_d will produce a part in time step $t+1$ because it was not starved in times step t, and is never blocked, by assumption.

- (N,1,1) is transient. If $\alpha_u(t+1) = 1$ and $\alpha_d(t+1) = 1$, and $n(t) > 0$, then M_d will produce a part in time step $t+1$ because it was not starved at time step t and is assumed never to be blocked, but M_u will not produce a time step in time $t+1$ because it was blocked at time step t.

Some states that were previously transient without idleness failure now have positive probability with idleness failure. For example, state $(0, 0, 0)$ was transient without idleness failure. However, with idleness failure, the system can reach state $(0, 0, 0)$ from the state $(0, 0, 1)$ if the downstream machine fails while it is starved, and the upstream machine is not repaired.

3.3.2 Lower boundary equations ($n \leq 1$)

$$\mathbf{p}(0,0,0) = (1-r_u)(1-r_d)\mathbf{p}(0,0,0) + (1-r_u)q_d\mathbf{p}(0,0,1) \tag{3.1}$$

$$\mathbf{p}(1,1,0) = r_u q_d \mathbf{p}(0,0,1) + r_u(1-r_d)\mathbf{p}(0,0,0) \tag{3.2}$$

$$\begin{aligned}\mathbf{p}(0,0,1) = {} & (1-r_u)r_d\mathbf{p}(0,0,0) + p_u r_d \mathbf{p}(1,1,0) \\ & +(1-r_u)(1-q_d)\mathbf{p}(0,0,1) + (1-r_u)r_d\mathbf{p}(1,0,0) \\ & +(1-r_u)(1-p_d)\mathbf{p}(1,0,1) + p_u(1-p_d)\mathbf{p}(1,1,1)\end{aligned} \tag{3.3}$$

$$\begin{aligned}\mathbf{p}(1,0,0) = {} & p_u(1-r_d)\mathbf{p}(1,1,0) + (1-r_u)(1-r_d)\mathbf{p}(1,0,0) \\ & +(1-r_u)p_d\mathbf{p}(1,0,1) + p_u p_d \mathbf{p}(1,1,1)\end{aligned} \tag{3.4}$$

$$\begin{aligned}\mathbf{p}(1,0,1) = {} & (1-r_u)r_d\mathbf{p}(2,0,0) + (1-r_u)(1-p_d)\mathbf{p}(2,0,1) \\ & +p_u r_d \mathbf{p}(2,1,0) + p_u(1-p_d)\mathbf{p}(2,1,1)\end{aligned} \tag{3.5}$$

$$\begin{aligned}\mathbf{p}(1,1,1) = {} & r_u r_d \mathbf{p}(0,0,0) + (1-p_u)r_d\mathbf{p}(1,1,0) \\ & +r_u(1-q_d)\mathbf{p}(0,0,1) + r_u r_d \mathbf{p}(1,0,0) \\ & +r_u(1-p_d)\mathbf{p}(1,0,1) + (1-p_u)(1-p_d)\mathbf{p}(1,1,1)\end{aligned} \tag{3.6}$$

$$\begin{aligned}\mathbf{p}(2,1,0) = {} & (1-p_u)(1-r_d)\mathbf{p}(1,1,0) + r_u(1-r_d)\mathbf{p}(1,0,0) \\ & +r_u p_d \mathbf{p}(1,0,1) + (1-p_u)p_d\mathbf{p}(1,1,1)\end{aligned} \tag{3.7}$$

3.3.3 Internal equations ($2 \leq n \leq N-2$)

The internal equations are identical to the internal equations in Gershwin [1].

$$\begin{aligned}
\mathbf{p}(n,0,0) &= (1-r_u)(1-r_d)\mathbf{p}(n,0,0) + (1-r_u)p_d\mathbf{p}(n,0,1) && (3.8)\\
&+p_u(1-r_d)\mathbf{p}(n,1,0) + p_u p_d \mathbf{p}(n,1,1)\\
\mathbf{p}(n,0,1) &= (1-r_u)r_d\mathbf{p}(n+1,0,0) + (1-r_u)(1-p_d)\mathbf{p}(n+1,0,1) && (3.9)\\
&+p_u r_d\mathbf{p}(n+1,1,0) + p_u(1-p_d)\mathbf{p}(n+1,1,1)\\
\mathbf{p}(n,1,0) &= r_u(1-r_d)\mathbf{p}(n-1,0,0) + r_u p_d\mathbf{p}(n-1,0,1) && (3.10)\\
&(1-p_u)(1-r_d)\mathbf{p}(n-1,1,0) + (1-p_u)p_d\mathbf{p}(n-1,1,1)\\
\mathbf{p}(n,1,1) &= r_u r_d\mathbf{p}(n,0,0) + r_u(1-p_d)\mathbf{p}(n,0,1) + (1-p_u)r_d\mathbf{p}(n,1,0) && (3.11)\\
&(1-p_u)(1-p_d)\mathbf{p}(n,1,1)
\end{aligned}$$

3.3.4 Upper boundary equations ($n \geq N-1$)

$$\begin{aligned}
\mathbf{p}(N-2,0,1) &= (1-r_u)(1-p_d)\mathbf{p}(N-1,0,1) && (3.12)\\
&+(1-r_u)r_d\mathbf{p}(N-1,0,0)\\
&+p_u r_d\mathbf{p}(N-1,1,0) + p_u(1-p_d)\mathbf{p}(N-1,1,1)\\
\mathbf{p}(N-1,0,1) &= (1-r_u)r_d\mathbf{p}(N,0,0) + q_u r_d\mathbf{p}(N,1,0) && (3.13)\\
\mathbf{p}(N-1,0,0) &= (1-r_u)p_d\mathbf{p}(N-1,0,1) && (3.14)\\
&+(1-r_u)(1-r_d)\mathbf{p}(N-1,0,0)\\
&+p_u(1-r_d)\mathbf{p}(N-1,1,0) + p_u p_d\mathbf{p}(N-1,1,1)
\end{aligned}$$

$$
\begin{aligned}
\mathbf{p}(N-1,1,0) &= r_u(1-r_d)\mathbf{p}(N-2,0,0) + r_u p_d \mathbf{p}(N-2,1,1) \qquad (3.15)\\
&\quad +(1-p_u)(1-r_d)\mathbf{p}(N-2,1,0)\\
&\quad +(1-p_u)p_d\mathbf{p}(N-2,1,1)\\
\mathbf{p}(N-1,1,1) &= r_u r_d \mathbf{p}(N-1,0,0) + (1-p_u)r_d\mathbf{p}(N-1,1,0) \qquad (3.16)\\
&\quad +(1-p_u)(1-p_d)\mathbf{p}(N-1,1,1) + (1-q_u)r_d\mathbf{p}(N,1,0)\\
&\quad +r_u r_d \mathbf{p}(N,0,0) + r_u(1-p_d)\mathbf{p}(N-1,0,1)\\
\mathbf{p}(N,1,0) &= r_u(1-r_d)\mathbf{p}(N-1,0,0) + r_u p_d \mathbf{p}(N-1,0,1) \qquad (3.17)\\
&\quad +r_u(1-r_d)\mathbf{p}(N-1,0,0) + (1-p_u)(1-r_d)\mathbf{p}(N-1,1,0)\\
&\quad +(1-p_u)r_d\mathbf{p}(N-1,1,1) + (1-q_u)(1-r_d)\mathbf{p}(N,1,0)\\
\mathbf{p}(N,0,0) &= (1-r_u)(1-r_d)\mathbf{p}(N,0,0) + q_u(1-r_d)\mathbf{p}(N,1,0) \qquad (3.18)
\end{aligned}
$$

Normalization

Since the sum of the probability of all states must be equal to one, we have the following normalization equation:

$$
1 = \sum_{\alpha_u=0}^{1} \sum_{\alpha_d=0}^{1} \sum_{n=0}^{N} \mathbf{p}(n, \alpha_u, \alpha_d) \qquad (3.19)
$$

Equations (3.1) through (3.19) define a probability mass function for the Markov chain model of the two-machine transfer line.

3.4 Performance Measures

The main important performance measures are *efficiency*, *average buffer level*, *probability of starvation*, and *probability of blockage*.

3.4.1 Average buffer level, probability of blockage and starvation

The average buffer level $\overline{n}$ is

$$\overline{n} = \sum_{\alpha_u=0}^{1} \sum_{\alpha_d=0}^{1} \sum_{n=0}^{N} n\mathbf{p}(n, \alpha_u, \alpha_d) \tag{3.20}$$

The downstream machine can be starved in two different ways. Suppose the upstream machine has been down and the downstream machine has depleted all the parts in the buffer. Then the downstream machine is starved while it is up. The probability that the downstream is up, but that there is no part in the buffer, denoted S_w^d, is

$$S_w^d = \mathbf{p}(0, 0, 1). \tag{3.21}$$

Since the machine can fail while it is starved, we need to quantify this probability. The probability that the downstream machine is down, but that there is no part in the buffer, denoted S_d^u is

$$S_d^d = \mathbf{p}(0, 0, 0). \tag{3.22}$$

Likewise, the probability that the upstream machine is blocked and is up, denoted B_w^u, is the probability that the machine is up, and that the intermediate buffer is full.

$$B_w^u = \mathbf{p}(N, 1, 0) \tag{3.23}$$

and the probability that the upstream machine is blocked and is up, denoted B_w^u, is the probability that the machine is down and that the intermediate buffer is full.

$$B_d^u = \mathbf{p}(N, 1, 0) \tag{3.24}$$

Note that without the idleness failure, the quantities S_d^d and B_d^u are zero.

3.4.2 Efficiency

The efficiency of the upstream machine, denoted E_u, is the probability that the upstream machine is working in time step $t+1$ and not blocked at time t. This is the fraction of time that M_u is working on a part. Since M_u is allowed to process only one part at one time step, the efficiency is actually a production rate of the machine.

$$E_u = Pr\left[\alpha_u(t+1) = 1 \cap n(t) < N\right] \tag{3.25}$$

The efficiency of the downstream machine,denoted E_u, is the probability that the downstream machine is working at time $t+1$ and not starved at time t. This is given by

$$E_d = Pr\left[\alpha_d(t+1) = 1 \cap n(t) > 0\right] \tag{3.26}$$

For our convenience, we define the following quantities: W^u and W^d, such that

$$\begin{aligned} W_u &= Pr\left[\alpha_d(t) = 1 \cap n(t) < n\right] \\ W_d &= Pr\left[\alpha_d(t) = 1 \cap n(t) > 0\right] \end{aligned}$$

Observe that these probabilities are different from the efficiency described in (3.25) and (3.26) in that each quantity is expressed with only single time step t.

Notice that (3.25) and (3.25) are awkward expressions because it involves states at two different time steps. Without the idleness failure the following equalities should hold [1]

$$\begin{aligned} W_u &= E_u \\ W_d &= E_d \end{aligned}$$

however, with the idleness failure, these equalities do not hold anymore. We need to transform them into a statement about the state of the system at a single time step. In order to do this, we need to show the following:

Repair Frequency Equals Failure Frequency

For every repair, there is a failure in steady state. This seems self-evident, so it is reassuring that the transition equations satisfy this condition. This equation is used in the decomposition described in Chapter ??. When the system is in steady state,

$$\begin{aligned} &Pr(M^u \text{ is up at } t+1 | M^u \text{ is down at } t) \times Pr(M^u \text{ is down at } t) \\ &\quad = Pr(M^u \text{ is down at } t+1 | M^u \text{ is up at } t) \times Pr(M^u \text{ is up at } t) \end{aligned}$$

That is, actually,

$$\begin{aligned} &r_u(Pr[\{\alpha_u = 0\} \cap \{n < N\}] + Pr[\{\alpha_u = 0\} \cap \{n = N\}]) \\ &\quad = p_u Pr[\{\alpha_u = 1\} \cap \{n < N\}] + q_u Pr[\{\alpha_u = 1\} \cap \{n = N\}] \end{aligned} \tag{3.27}$$

Similarly, for the downstream machine we have

$$r_d(Pr[\{\alpha_d = 0\} \cap \{n > 0\}] + Pr[\{\alpha_d = 0\} \cap \{n = 0\}]) \tag{3.28}$$
$$= p_d Pr[\{\{\alpha_u = 1\} \cap \{n > 0\}] + q_d Pr[\{\alpha_u = 1\} \cap \{n = 0\}]$$

We can use (3.27) and (3.28) to derive alternative expressions for (3.25) and (3.26). The upstream machine, E_u, produces a part in time step $t + 1$ if it is up at the end of time step $t + 1$ and was not blocked at the end of time step t. We can then write E_u as follows:

$$E_u = Pr\left[\alpha_u(t+1) = 1 \cap n(t) < N\right]. \tag{3.29}$$

This expression has both time step $t + 1$ and time step t in it. We proceed by conditioning on events occurring in time step t to write (3.29) in terms of events occurring entirely in time step t. By doing so, we will be able to express the production rate of the upstream machine entirely in terms of the state probabilities, which are defined only in one time step.

$$\begin{aligned}
E_u &= Pr[\{\alpha_u(t+1) = 1\} \cap \{n(t) < N\}] \\
&= Pr[\{\alpha_u(t+1) = 1\} | \{\alpha_u(t) = 1\} \cap \{n(t) < N\}] \\
&\quad \times Pr[\{\alpha_u(t) = 1\} \cap \{n(t) < N\}] \\
&\quad + Pr[\{\alpha_u(t+1) = 1\} | \{\alpha_u(t) = 0\} \cap \{n(t) < N\}] \\
&\quad \times Pr[\{\alpha_u(t) = 0\} \cap \{n(t) < N\}] \\
\\
&= (1 - p_u) Pr[\{\alpha_u(t) = 1\} \cap \{n(t) < N\}]
\end{aligned}$$

$$+r_u Pr[\{\alpha_u(t)=0\}\cap\{n(t)<N\}]$$

$$= Pr[\{\alpha_u(t)=1\}\cap\{n(t)<N\}] - p_u Pr[\{\alpha_u(t)=1\}\cap\{n(t)<N\}]$$
$$+r_u Pr[\{\alpha_u(t)=0\}\cap\{n(t)<N\}]$$

$$= Pr[\{\alpha_u(t)=1\}\cap\{n(t)<N\}]$$
$$+q_u Pr[\{\alpha_u(t)=1\}\cap\{n(t)=N\}]$$
$$-r_u Pr[\{\alpha_u(t)=0\}\cap\{n(t)=N\}]$$

By the equations (3.23), (3.24), and (3.27), we can express the above such that:

$$E_u = W_u + q_u B_d^u - r_u B_w^u \tag{3.30}$$

Similarly,

$$E_d = W_d + q_d S_d^d - r_u S_w^d \tag{3.31}$$

We note that those equations also reduce to the efficiency without idleness failure if the idleness failure probabilities are set to zero.

3.4.3 Flow Rate-Idle Time

We now derive the flow rate-idle time relationships for the two-machine line. First, we define the following quantities:

$$D_u = Pr\left[\alpha_u(t) = 0 \cap n(t) < N\right] \tag{3.32}$$

This quantity is the probability that the upstream machine is down and not blocked at time t. Then we can state that

$$W_u + D_u + B_w^u + B_d^u = 1 \tag{3.33}$$

Some simple algebraic manipulation of (3.27) and (3.30) yields

$$\begin{aligned} D_u &= \frac{p_u}{r_u}E_u - (1-p_u)B_d^u + \frac{q_u}{r_u}(1-p_u)B_w^u && (3.34) \\ W_u &= E_u - q_u B_w^u + r_u W_d^u && (3.35) \end{aligned}$$

Now, from Equation (3.33), we have

$$1 - B_w^u - B_d^u = W_u + D_u \tag{3.36}$$

By substituting the expressions in (3.34) and (3.35) into (3.36) and manipulating algebraically, we arrive at

$$E_u = e_u(1 - B_w^u) - e_u q_u B_w^u \left(\frac{r_u + r_d - p_u r_d - 2r_u r_d}{r_u(r_u + r_d - r_u r_d)}\right). \tag{3.37}$$

In a similar manner, it is possible to derive the following expression for the downstream machine.

$$E_d = e_d(1 - S_w^d) - e_d q_d S_w^d \left(\frac{r_u + r_d - r_u p_d - 2r_u r_d}{r_d(r_u + r_d - r_u r_d)}\right). \tag{3.38}$$

Now, observe that if the idleness failure probabilities in either of the above equations are zero, then that expression reduces to that found in Gershwin [1].

Chapter 4

Part Type 1 Analysis

4.1 Introduction

In this chapter, we analyze the behavior of the two-part type line, especially for Type 1 parts. As mentioned in Section 2.3, we take the position of an observer for Type 1 parts in each machine. We then seek to capture the Type 1 part behavior, as seen by the observer. We first try to capture the isolated machine parameters of each machine in the part type one line, denoted by $LP(1)$. Once we analyze the machine parameters in $LP(1)$, we apply decomposition method.

4.2 Part type decomposition for Type 1

4.2.1 Isolated machine parameters in Type 1 Line

Type 1 observer watches the flow of only Type 1 parts. The observer is unable to tell whether or not he is in the system of processing two part types. The decomposition of the part type one line is illustrated in Figure 4-1. A machine in $LP(1)$ is denoted by $M(i,j)$, where $i = 1, \ldots, k$ is the sequence of the machines and $j = 1$ is the part

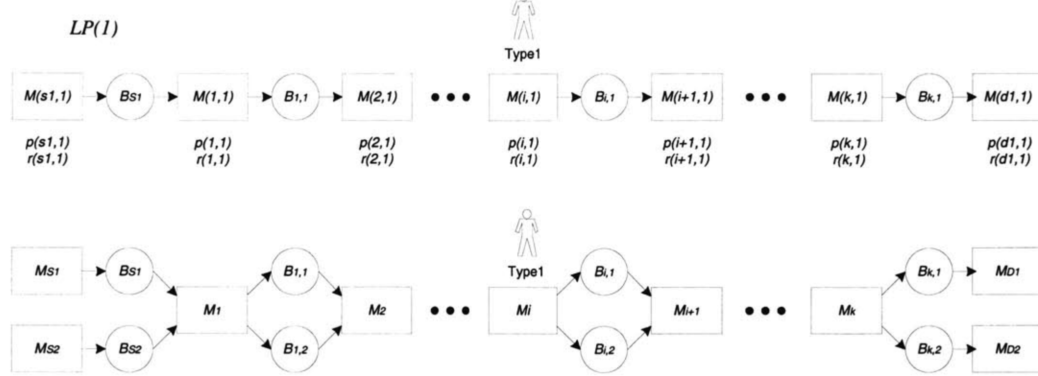

Figure 4-1: Part Type Decomposition

type number. Also we let $B(i,1)$ denote a buffer in $LP(1)$. We take the the first and last machine in $LP(1)$, denoted $M(s1,1)$ and $M(d1,1)$, to be the same as M_{s1} and M_{d1}, respectively. This is because these machines are processing only Type 1 parts in the real line. We take all the buffers, including supply and demand buffers in $LP(1)$, to be the same as all the buffers in the real line. Therefore, the size of the buffer in $LP(1)$, denoted by $N(i,1)$, is $N_{i,1}$.

4.2.2 Machine parameters in the part-type-one line

Now, we have a hypothetical transfer line, $LP(1)$, processing only Type 1 parts. We need to calculate the machine parameters in order to analyze the line. We denote the repair and failure probability of $M(i,1)$ by $r(i,1)$ and $p(i,1)$, respectively. Type 1 parts have priority over Type 2 parts. Therefore, the Type 1 part observer sees that the machine is processing Type 1 parts whenever there is a Type 1 part available in the immediate upstream buffer and whenever there is space to put a Type 1 part in the immediate downstream buffer. The parameter $p(i,1)$ is the failure probability of machine $M(i,1)$. That is, it is the probability that $M(i,1)$ does not make a part at time $t+1$, given it did make one, at time t. This is expressed as

$$p(i,1) = Pr\left[M(i,1) \text{ down at time } t+1 | M(i,1) \text{ up, at time } t\right] \quad (4.1)$$

Note that unlike the single part type processing line case, in the above conditional probability, $M(i,1)$ is not restricted to fail while it is not starved or blocked. There are two ways that $M(i,1)$ is down at time $t+1$, and $M(i,1)$ was up at time t. They are:

- M_i is down at time $t+1$, and M_i was up, and not starved and blocked for Type 1 parts at time t, or
- M_i is down at time $t+1$, and at time t, M_i was up, and either starved or blocked for Type 1 parts and not blocked and starved for Type 2 parts.

We express (4.1) such that

$$\begin{aligned}
p(i,1) &= Pr\left[\alpha_i(t+1) = 0 | \{\alpha_i(t) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t) < N_{i,1}\} \cup \right. \\
&\qquad \{\alpha_i(t) = 1 \cap (n_{i-1,1}(t) = 0 \cup n_{i,1}(t) = N_{i,1}) \\
&\qquad \left. \cap\, n_{i-1,2}(t) > 0 \cap n_{i,2}(t) < N_{i,2}\}\right] \\
&= Pr\left[\alpha_{i,1}(t+1) = 0 | \{\alpha_{i,1}(t) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t) < N_{i,1}\}\right] \\
&= p_i
\end{aligned}$$

The parameter $r_{i,1}$ is the repair probability of machine $M(i,1)$. That is, it is the probability that $M(i,1)$ makes a part at time $t+1$, given it did not make one at time t. This is expressed as

$$r(i,1) \quad = \quad Pr\left[M(i,1) \text{ up at time } t+1 | M(i,1) \text{ down at time } t\right]$$

There are two ways that $M(i,1)$ produces a Type 1 part at time $t+1$, given that it did not produce one at time t. They are

- M_i is up at time $t+1$ and M_i was down at time t, or
- M_i is processing a Type 1 part at time $t+1$ and was processing a Type 2 part at time t.

For the second case, $M_{i,1}$ can work on a Type 2 part only when the M_i is blocked or starved. The observer would believe that $M(i,1)$ is not processing a Type 1 part at time t, because of the blockage or starvation in the part type one line. Therefore, the observer would not notice the second case and we can ignore the case. Since we consider only the first term, we can express as

$$\begin{aligned} r(i,1) &= Pr\left[\alpha_{i,1}(t+1) = 1 | \alpha_{i,1}(t) = 0\right] \\ &= r_i \end{aligned}$$

It follows from the definition of r_i. In similar manner, it is possible to evaluate $r(s1,1)$, $p(s1,1)$, $r(d1,1)$, and $p(d1,1)$. Therefore, failure and repair probabilities in $LV(1)$ are

$$
\begin{aligned}
r(s1,1) &= r_{s1,1} \\
r(d1,1) &= r_{d1,1} \\
p(s1,1) &= p_{s1,1} \\
p(d1,1) &= p_{d1,1} \\
r(i,1) &= r_i \\
p(i,1) &= p_i \\
& \text{for } i = 1,\ldots,k
\end{aligned}
$$

4.2.3 Idleness failure in Type 1 part line

Suppose that the Type 1 observer does not see any inflow of parts due to starvation; there has been an actual machine failure in the upstream, starving the line downstream of it. The observer would think that the machine is idle. However, the actual machine may not be idle and maybe working on Type 2 parts. While the actual machine is working on Type 2 parts, the machine can fail. From the part type one observer's perspective, the machine can fail while it is idle. We call this idleness failure.

Unlike the machine parameters, probability of idleness failure depends on the size neighboring buffers. This is because the idleness failure is the failure rate while the machine is either blocked or starved, and these blockage and starvation depend on the size of buffers.

Instead deriving an equation for the idleness failure for each machine, we consider idleness failure to be a two machine line parameter as illustrated in Figure 3-1.

4.3 Two machine line decomposition

We present a decomposition for the line $LV(1)$ in this section. The approach is similar to the decomposition for the single part type. However, we consider idleness failure in our model.

4.3.1 Notation and assumptions

Figure 4-2 shows the decomposition of the line $LV(1)$. For each buffer $B_{i,1}$, we have a representative two-machine line, denoted $L(i,1)$. The upstream machine is denoted $M_u(i,1)$, and the downstream machine is denoted $M_d(i,1)$. We denote the repair, failure, and idleness failure parameters of $M_u(i,1)$ by $r_u(i,1)$, $p_u(i,1)$ and $q_u(i,1)$, respectively. Likewise, we denote the repair, failure, and idleness failure parameters of $M_d(i,1)$ by $r_d(i,1)$, $p_d(i,1)$ and $q_d(i,1)$, respectively. We take the size of the buffer $B(i,1)$ in $L(i,1)$, denoted $N(i,1)$, to be the same as $B_{i,1}$. We denote the current level of the buffer in $L(i,1)$ by $n(i,1)(t)$.

We will assume that the probability that a machine M_i is simultaneously starved and blocked for a given part is negligible. That is, we assume that

Approximation. $Pr\left[n_{i-1,j}(t) = 0 \cap n_{i,j}(t) = N_{i,j}\right] \approx 0$ □

In order for a machine to be both starved and blocked for a part simultaneously, it is necessary that at some point, the machine had exactly one part in the upstream buffer, and exactly one space in the downstream buffer. We expect that this is a rare occurrence because in order for it to happen, the buffer upstream of machine M_i would have to have exactly one part in it, and the buffer downstream of machine M_i would have to have exactly one space for a part.

We also define some useful quantities.

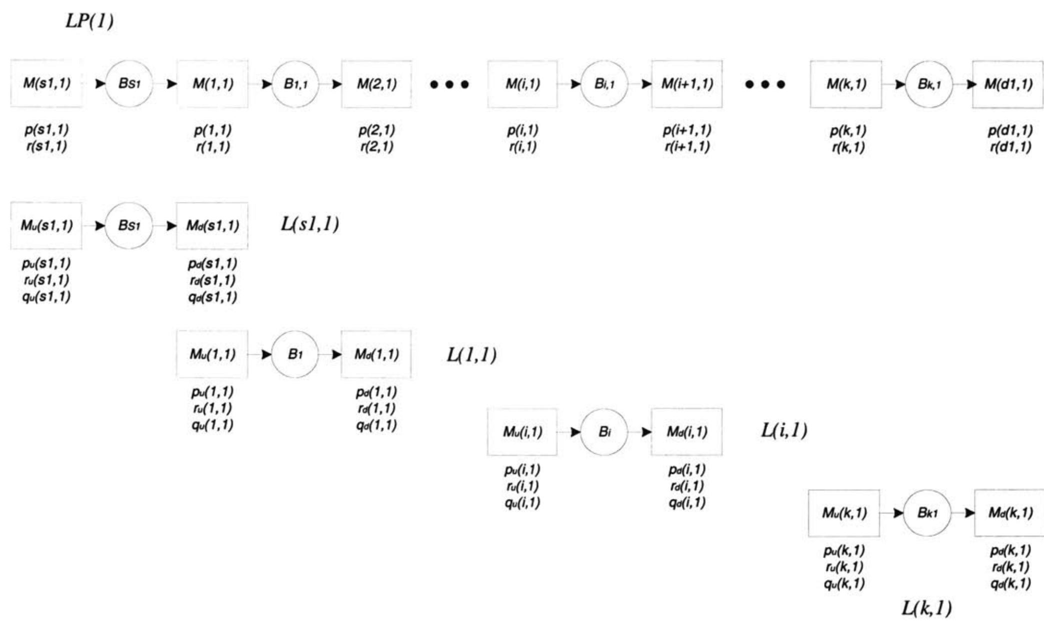

Figure 4-2: Part Type Decomposition

$$
\begin{aligned}
S_d^d(i-1,1) &= Pr[\alpha_i(t)=0 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1) < N_{i,1}] \\
S_w^d(i-1,1) &= Pr[\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1) < N_{i,1}] \\
B_d^u(i,1) &= Pr[\alpha_i(t)=0 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) = N_{i,1}] \\
B_w^u(i,1) &= Pr[\alpha_i(t)=1 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) = N_{i,1}]
\end{aligned}
$$

Notice that these probabilities are quantities for two-machine-line. The quantity $S_d^d(i-1,1)$ denotes the probability that $M_d(i-1,1)$ is down and starved, while $S_w^d(i-1,1)$ denotes the probability that $M_d(i-1,1)$ is up and starved. The quantity $B_d^u(i,1)$ denotes the probability that $M_u(i,1)$ is down and blocked, while $B_w^u(i,1)$

denotes the probability that $M_u(i,1)$ is up and blocked. We define the preceeding quantities because the main technique in deriving the decomposition is to relate events in the real line with events in the two-machine-lines. Since the two-machine-lines model the behavior of material flowing into and out of the buffers in the real line, the states when buffers in the real line are empty or full should correspond to states in the relevant two-machine sub-lines where the intermediate buffer is empty or full.

4.3.2 Conservation of Flow

Since there is no mechanism for the creation or destruction of material, we expect the flow of each type to be conserved. That is, for $j = 1, 2$, we require that

$$E(i,1) = E(s1,1) \quad \text{for } i = 1, \ldots, K. \tag{4.2}$$

4.3.3 Resumption of Flow Equations

The resumption of flow equations are used to calculate the values of $r_u(i,1)$ for $i = 1, \ldots, K$ and $r_d(i,1)$ for $i = s1, \ldots, K-1$. We begin with the calculation of an expression for $r_u(i,1)$, $i = 1, \ldots, K$.

The parameter $r_u(i,1)$ is the repair probability of the upstream machine of $L(i,1)$. That is, it is the probability that $M_u(i,1)$ makes a part in time $t+1$, given it did not make one, and was not blocked, in time t. There are three ways that M_i could have not produced a type one part in time step t, given that it was not blocked for type one at time $t-1$. They are:

- M_i was down at time t, and not starved for type one at time $t-1$,
- M_i was down at time t, and starved for type one at time $t-1$, or
- M_i was up at time t, and starved for type one at time $t-1$

If we are considering a processing line making only Type 1 parts, and assume the the machines have operation-dependent failures, then the probability of the second point occurring is zero, as the a machine being simultaneously down and starved implies that the machine failed while idle. We can express such that

$$\begin{aligned} r_u(i,1) &= Pr\left[\alpha_i(t+1) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t-1) < N_{i,1}\right| \\ &\quad (\{\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) > 0\} \cup \\ &\quad \{\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) = 0\} \cup \\ &\quad \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = 0\}) \cap n_{i,1}(t-1) < N_{i,1}] \end{aligned}$$

If we define the following events

$$\begin{aligned} U &= \{\alpha_i(t+1) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t-1) < N_{i,1}\} \\ A &= \{\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) < N_{i,1}\} \\ B &= \{\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}\} \\ C &= \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}\} \end{aligned}$$

We note that the events A, B, and C are mutually exclusive. Then we can rewrite $r_u(i,1)$ as

$$r_u(i,1) = Pr\left[U | A \cup B \cup C\right]$$

$$
\begin{aligned}
&= Pr\,[U|A]Pr\,[A|A \cup B \cup C] + Pr\,[U|B]Pr\,[B|A \cup B \cup C] \\
&\quad + Pr\,[U|C]Pr\,[C|A \cup B \cup C] \\
&= \frac{Pr\,[U|A]Pr\,[A] + Pr\,[U|B]Pr\,[B] + Pr\,[U|C]Pr\,[C]}{Pr\,[A \cup B \cup C]}
\end{aligned}
$$

We need to calculate these quantities. We first calculate the probability of the event A. That is, we calculate the probability that M_i is down, and not blocked or starved. In order to do so, we need to classify exactly when $M_u(i,1)$ is down and not blocked. There are three mutually exclusive, collectively exhaustive possibilities:

- M_i is down, and not blocked or starved for type one,
- M_i is down, and starved, but not blocked for type one, or
- M_i is up, and starved, but not blocked for type one.

That is,

$$
\begin{aligned}
&Pr[\alpha_u(i,1) = 0 \cap n_{i,1} < N_{i,1}] \\
&\quad = Pr[\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) < N_{i,1}] \\
&\quad + Pr[\alpha_i(t) = 0 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}] \\
&\quad + Pr[\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}]
\end{aligned} \tag{4.3}
$$

Using (4.2), we can write the following:

$$
Pr[\alpha_u(i,1) = 0 \cap n_{i,1} < N_{i,1}] \tag{4.4}
$$

$$= Pr[A] + S_d^d(i-1,1) + S_w^d(i-1,1)$$

Therefore,

$$Pr\left[A\right] \;=\; Pr\left[\alpha_u(i,1) = 0 \cap n_{i,1} < N_{i,1}\right] - S_d^d(i-1,1) - S_w^d(i-1,1)$$

Now, from the repair-frequency-equals-failure-frequency of the upstream machine for the two machine line stated in (3.27), together with the quantity we defined in (4.3), this can be written as

$$\begin{aligned} Pr\left[A\right] \;=\; & \frac{p_u(i,1)}{r_u(i,1)} Pr\left[\alpha_u(i,1) = 1 \cap n_{i,1} < N_{i,1}\right] + \frac{q_u(i,1)}{r_u(i,1)} B_w^u(i,1) \\ & -B_d^u(i,1) - S_d^d(i-1,1) - S_w^d(i-1,1) \end{aligned}$$

Note that in the fist term, the probability that the upstream machine is up and is not blocked can be related with the efficiency we derived in (3.31). Finally we arrive at the following expression for $Pr[A]$.

$$\begin{aligned} Pr\left[A\right] \;=\; & \frac{p_u(i,1)}{r_u(i,1)}\left[W_u(i,1) + q_u(i,1)B_d^u(i,1) - r_u(i,1)B_w^u(i,1)\right] \\ & +\frac{q_u(i,1)}{r_u(i,1)} B_w^u(i,1) \\ & -B_d^u(i,1) - S_d^d(i-1,1) - S_w^d(i-1,1) \end{aligned}$$

Next we calculate the probability of the event B. That is we need to calculate the probability that M_i is down, and not blocked, but starved. If we relate this event with the event in the two-machine-line that is defined in (4.2),

$$Pr[B] = S_d^d(i-1,1) \tag{4.5}$$

Similarly, the probability of the event C is

$$Pr[C] = S_w^d(i-1,1) \tag{4.6}$$

Next we need to calculate $Pr\left[A \cup B \cup C\right]$. Actually, they are mutually exclusive events. Therefore, probability of this event is the sum of the probabilities of the events X, Y, and Z.Therefore, we can write the following:

$$\begin{aligned}
Pr\left[A \cup B \cup C\right] &= Pr[A] + Pr[B] + Pr[C] \\
&= \frac{p_u(i,1)}{r_u(i,1)}[W_u(i,1) + q_u(i,1)B_d^u(i,1) - r_u(i,1)B_w^u(i,1)] \\
&\quad + \frac{q_u(i,1)}{r_u(i,1)}B_w^u(i,1) \\
&\quad - B_d^u(i,1) - S_d^d(i-1,1) - S_w^d(i-1,1) \\
&\quad + S_d^d(i-1,1) + S_w^d(i-1,1) \\
&= \frac{p_u(i,1)}{r_u(i,1)}[W_u(i,1) + q_u(i,1)B_d^u(i,1) - r_u(i,1)B_w^u(i,1)] \\
&\quad + \frac{q_u(i,1)}{r_u(i,1)}B_w^u(i,1) - B_d^u(i,1)
\end{aligned}$$

Next, we calculate the conditional probability that we make a part at time step $t+1$, given that we did not make one at time t because M_i was down and not blocked or starved for Type 1 parts.

$$\begin{aligned} Pr\,[U|A] &= Pr\,[\alpha_i(t+1)=1 \cap n_{i-1,1}(t)>0 \cap n_{i,1}(t-1)<N_{i,1}| \\ &\qquad \alpha_i(t)=0 \cap n_{i-1,1}(t-1)>0 \cap n_{i,1}(t-1)<N_{i,1}] \\ &= Pr\,[\alpha_i(t+1)=1|\alpha_i(t)=0] \\ &= r_i. \end{aligned}$$

Here, the last step follows from the definition of r_i.

Next, we calculate the conditional probability that M_i makes a type one part in time $t+1$, given it did not make one in time t because the machine was down at time t, and starved for type one parts in step $t-1$. This is the probability that M_i is repaired, and M_{i-1} makes a type one part.

$$\begin{aligned} Pr\,[U|B] &= Pr\,[\alpha_i(t+1)=1 \cap n_{i-1,1}(t)>0 \cap n_{i,1}(t-1)<N_{i,1}| \\ &\qquad \alpha_i(t)=0 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}] \\ &\approx r_i r_u(i-1,1). \end{aligned}$$

In this derivation, two events are occurred simultaneously. First, machine M_i is repaired, which occurs with probability r_i. Second, there is a repair of M_{i-1}, or M_{i-1} becomes not starved for Type 1 parts due to a repair upstream.

Next, we calculate the conditional probability that M_i makes a Type 1 part in time $t+1$, given it did not make one in time t because it was up, but starved for Type 1 parts.

$$\begin{aligned} Pr\left[U|C\right] &= Pr\left[\alpha_i(t+1)=1 \cap n_{i-1,1}(t)>0 \cap n_{i,1}(t-1)<N_{i,1}\right| \\ &\qquad \alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}] \\ &\approx r_u(i-1,1)(1-q_d(i-1,1)). \end{aligned}$$

Here, the approximation comes from two events occurring. First, there is a repair of M_{i-1}, or M_{i-1} becomes not starved for Type 1 parts due to a repair upstream. Second, machine M_i did not fail while idle. This corresponds to the machine not failing while making a Type 2 part. The observer in the Type 1 buffer does not even know if M_i was working on a Type 2 part, but he observes that the machine did not undergo an idleness failure. This occurs with probability $1-q_d(i-1,1)$.

If we put (4.5), (4.5), (4.6), (4.7), (??), and (??) together into (4.3), then

$$\begin{aligned} r_u(i,1) &= \Bigg(r_i\left[\frac{p_u(i,1)}{r_u(i,1)}\left\{E_u(i,1)-q_u(i,1)B_d^u(i,1)+r_u(i,1)B_w^u(i,1)\right\}\right. \\ &\quad \left.+\frac{q_u(i,1)}{r_u(i,1)}B_w^u(i,1)-B_d^u(i,1)-S_d^d(i-1,1)-S_w^d(i-1,1)\right] \\ &\quad +r_i r_u(i-1,1)S_d^d(i-1,1)+r_u(i-1,1)(1-q_d(i-1,1))S_w^d(i-1,1)\Bigg) \\ &\div \Bigg(\frac{p_u(i,1)}{r_u(i,1)}\left\{E_u(i,1)-q_u(i,1)B_d^u(i,1)+r_u(i,1)B_w^u(i,1)\right\} \\ &\quad +\frac{q_u(i,1)}{r_u(i,1)}B_w^u(i,1)-B_d^u(i,1)\Bigg) \end{aligned} \tag{4.7}$$

We observe that if we set $q_u(i,1) = q_d(i-1,1) = 0$, then $B_d^u(i,1)$ and $S_d^d(i-1)$ both become zero. Then $r_u(i,1)$ becomes

$$
\begin{aligned}
r_u(i,1) &= r_i\left(\frac{p_u(i,1)}{r_u(i,1)}\{E(i-1,1) - p_s(i-1,1)\} + r_u(i-1,1)p_s(i-1,1)\right) \\
&\div \frac{p_u(i,1)}{r_u(i,1)}E(i,1).
\end{aligned}
$$

Observing that Conservation of Flow gives $E(i-1,1) = E(i,1)$, this equation is reduced to the resumption of flow equation for the one-part transfer line decomposition given in [1].

Finally, we can similarly derive an expression for $r_d(i,1)$. The equation for $r_d(i,1)$ is given as:

$$
\begin{aligned}
r_d(i,1) &= \left(r_{i+1}\left[\frac{p_d(i,1)}{r_d(i,1)}\left\{E_d(i+1,1) - q_d(i,1)S_w^d(i,1) + r_d(i,1)S_d^d(i,1)\right\}\right.\right. \qquad (4.8) \\
&\quad \left. + \frac{q_d(i,1)}{r_d(i,1)}S_w^d(i,1) - B_d^u(i,1) - S_d^d(i,1) - B_w^u(i,1)\right] \\
&\quad \left. + r_{i+1}r_d(i+1,1)B_w^u(i+1,1) + r_d(i+1,1)(1-q_u(i+1,1))B_d^u(i+1,1)\right) \\
&\div \left(\frac{p_d(i,1)}{r_d(i,1)}\left\{E_d(i+1,1) - q_d(i,1)S_d^d(i,1) + r_d(i,1)S_w^d(i,1)\right\}\right. \\
&\quad \left. + \frac{q_d(i,1)}{r_d(i,1)}S_w^d(i,1) - S_w^d(i,1)\right)
\end{aligned}
$$

We note that this equation also reduces to the one-part type resumption of flow equation for the downstream machine if the idleness failure probabilities are set to zero.

4.3.4 Idleness failure equation

The probability $q_u(i,1)$ represents the probability that pseudo-machine $M_u(i,1)$ is down at time $t+1$, given that is was up and blocked at time t. Since we assume operation dependent failures, the only way that this is possible is if the processing machine M_i failed while making a Type 1 in step t. Therefore, this can be expressed as

$$\begin{aligned} q_u(i,1) &= Pr\left[\alpha_i(t+1)=0|\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}\right. \\ &\qquad \left. \cap\, n_{i-1,2}(t-1)>0 \cap n_{i,2}(t-1)<N_{i,1}\right] \\ &= p_i Pr\left[\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}\right. \\ &\qquad \left. \cap\, n_{i-1,2}(t-1)>0 \cap n_{i,2}(t-1)<N_{i,1}\right] \end{aligned}$$

It remains to calculate the expression in the second line above. From first principals, we have

$$Pr[A] = Pr[A \cap B] + Pr[A \cap B^c]$$

Thus,

$$\begin{aligned} & Pr\left[\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}\right. \\ &\qquad \left. \cap\, n_{i-1,2}(t-1)>0 \cap n_{i,2}(t-1)<N_{i,1}\right] \\ = \; & Pr\left[\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}\right] \\ - \; & Pr\left[\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=0 \cap n_{i,1}(t-1)<N_{i,1}\right. \\ &\qquad \left. \cap\, (n_{i-1,2}(t-1)=0 \cup n_{i,2}(t-1)=N_{i,1})\right] \end{aligned}$$

The first term is defined to be $S_w^d(i-1,1)$. We argue that the second term can be approximated by

$$\left(S_w^d(i-1,2)+B_w^u(i,2)\right)\frac{S_w^d(i-1,1)}{S_w^d(i-1,1)+B^{uw}(i,1)}.$$

This follows because, from (??), we know that the probability that machine M_i is either blocked or starved for type one, and either blocked or starved for type two is given by

$$S_w^d(i-1,2)+B_w^u(i,2).$$

It follows that the probability that M_i is starved for type one, and either blocked or starved for type two would be the probability that it is either blocked or starved for type one, and either blocked or starved for type two, times the weighted probability that the machine is starved for type one, given it is blocked or starved for type one. The result is our idleness failure equation for the upstream machine of line $L(i,1)$.

$$q_u(i,1)=p_iS_w^d(i-1,1)\left(1-\frac{S_w^d(i-1,2)+B_w^u(i,2)}{S_w^d(i-1,1)+B_w^u(i,1)}\right).$$

Similarly, we can derive

$$q_d(i-1,1)=p_iB_w^u(i,1)\left(1-\frac{S_w^d(i-1,2)+B_w^u(i,2)}{S_w^d(i-1,1)+B_w^u(i,1)}\right).$$

4.3.5 Boundary Conditions

The boundary conditions of the Type 1 part line is following:

$$
\begin{aligned}
r_u(0,1) &= r_{1,1} \\
p_u(0,1) &= p_{1,1} \\
q_u(0,1) &= 0 \\
r_d(K,1) &= r_{K+1,1} \\
p_d(K,1) &= p_{K+1,1} \\
q_d(K,1) &= 0
\end{aligned}
$$

Chapter 5

Part Type 2 Analysis

5.1 Introduction

In this chapter, we analyze the behavior of the two-part type line, especially for Type 2 parts. As mentioned in Section 2.3, we take the position of an observer for Type 2 parts in each machine. We then seek to capture the Type 2 part behavior, as seen by the observer. We first try to capture the isolated machine parameters of each machine in Type 2 line, denoted by $LP(2)$. Once we analyze the machine parameters in $LP(2)$, we apply decomposition method.

5.2 Part type decomposition for Type 2

5.2.1 Isolated machine parameters in Type 2 Line

Type 2 observer watches the flow of only Type 2 parts. The observer is unable to tell whether or not he is in the system of processing two part types. The decomposition of the part type one line is illustrated in Figure 5-1. A machine in $LP(2)$ is denoted by $M(i,j)$, where $i = 1, \ldots, k$ is the sequence of the machines and $j = 2$ is the part

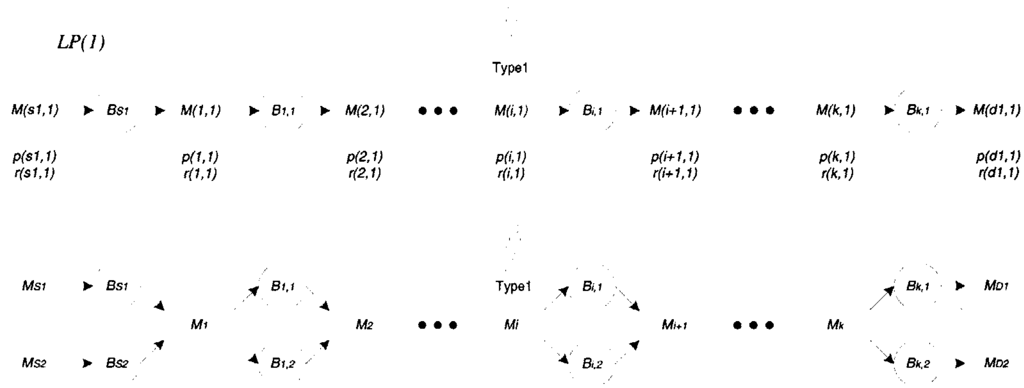

Figure 5-1: Part Type Decomposition

type number. Also we let $B(i,2)$ denote a buffer in $LP(2)$. We take the the first and last machine in $LP(2)$, denoted $M(s1,2)$ and $M(d1,2)$, to be the same as M_{s2} and M_{d2}, respectively. This is because these machines are processing only Type 2 parts in the real line. We take all the buffers, including supply and demand buffers in $LP(2)$, to be the same as all the buffers in the real line. Therefore, the size of the buffer in $LP(2)$, denoted by $N(i,2)$, is $N_{i,2}$.

5.3 Two machine line decomposition

5.3.1 Machine parameters in Type 2 line

Now, we have a hypothetical transfer line, $LP(2)$, processing only Type 2 parts. We need to calculate the machine parameters in order to analyze the line. We denote the repair and failure probability of $M(i,2)$ by $r(i,2)$ and $p(i,2)$, respectively. Type 1 parts have priority over Type 2 parts. Therefore, the Type 2 part observer sees that the machine is processing Type 2 parts when there is a blockage or starvation for Type 1 and also there is a Type 2 part available in the immediate upstream buffer and there is space to put a Type 2 part in the immediate downstream buffer.

5.3.2 Interruption of Flow

The parameter $p(i,2)$ is the failure probability of machine $M(i,2)$. That is, it is the probability that $M(i,2)$ does not make a part at time $t+1$, given it did make one, at time t. This is expressed as

$$p(i,2) = Pr\left[M(i,2) \text{ down at time } t+1 | M(i,2) \text{ up, at time } t\right] \tag{5.1}$$

There are four ways that $M(i,2)$ is down at time $t+1$, and $M(i,2)$ was up at time t. They are:

- M_i is up and not starved or block for Type 1 at time $t+1$, and M_i was up, and was starved for Type 1 parts at time t, or
- M_i is up and not starved or block for Type 1 at time $t+1$, and M_i was up, and was blocked for Type 1 parts at time t, or
- M_i is down at time $t+1$, and M_i was up, and was starved for Type 1 parts at time t, or
- M_i is down at time $t+1$, and M_i was up, and was blocked for Type 1 parts at time t.

We define the following events:

$$\begin{aligned} A &= \{\alpha_i(t) = 0\} \\ B &= \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) < N_{i,1}\} \end{aligned}$$

$$
\begin{aligned}
D &= \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}\} \\
E &= \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) = N_{i,1}\}
\end{aligned}
$$

Note that the event A,B,C,D, and E are mutually exclusive. We express (5.1) such that

$$
\begin{aligned}
p(i,2) &= Pr\left[A \cup B | D \cup E\right] \\
&= \frac{Pr\left[(A \cup B) \cap (D \cup E)\right]}{Pr\left[D \cup E\right]} \\
&= \frac{Pr\left[(A \cap (D \cup E)) \cup (B \cap (D \cup E))\right]}{Pr\left[D \cup E\right]} \\
&= Pr\left[A | D \cup E\right] + Pr\left[B | D \cup E\right] \\
&= \frac{Pr\left[D\right]}{Pr\left[D\right] + Pr\left[E\right]}(Pr[A|D] + Pr[B|D]) \\
&= \frac{Pr\left[E\right]}{Pr\left[D\right] + Pr\left[E\right]}(Pr[A|E] + Pr[B|E])
\end{aligned}
$$

We need to calculate these quantities. By the definition of the probability of failure, we know that

$$
Pr[A|D] = Pr[A|E] = p_i \tag{5.2}
$$

Next, we calculate the conditional probability that M_i makes a Type 1 part in time $t+1$, given it did not make one in time t because it was up, but starved for Type 1 parts.

$$
\begin{aligned}
Pr\,[B|D] &= Pr\,[\alpha_i(t+1) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t-1) < N_{i,1}| \\
&\qquad \alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = 0 \cap n_{i,1}(t-1) < N_{i,1}] \\
&\approx r_u(i-1,1)(1-q_d(i-1,1)).
\end{aligned}
$$

Here, the approximation comes from two events occurring. First, there is a repair of M_{i-1}, or M_{i-1} becomes not starved for Type 1 parts due to a repair upstream. Second, machine M_i did not fail while idle. This corresponds to the machine not failing while making a Type 2 part. The observer in the Type 1 buffer does not even know if M_i was working on a Type 2 part, but he observes that the machine did not undergo an idleness failure.

Likewise,

$$
\begin{aligned}
Pr\,[B|E] &= Pr\,[\alpha_i(t+1) = 1 \cap n_{i-1,1}(t) > 0 \cap n_{i,1}(t-1) < N_{i,1}| \\
&\qquad \alpha_i(t) = 1 \cap n_{i-1,1}(t-1) > 0 \cap n_{i,1}(t-1) = N_{i,1}] \\
&\approx r_d(i,1)(1-q_u(i,1)).
\end{aligned}
$$

Also, by the definition of the two-machine line, we know that

$$
\begin{aligned}
Pr[D] &= S_w^d(i-1,1) \\
Pr[E] &= B_w^u(i,1)
\end{aligned}
$$

Therefore, $p(i,2)$ is

$$p(i,2) = \frac{S_w^d(i-1,1)}{S_w^d(i-1,1)+B_w^u(i,1)}(r_u(i-1,1)(1-q_d(i-1,1))+p_i) + \frac{B_w^u(i,1)}{S_w^d(i-1,1)+B_w^u(i,1)}(r_d(i,1)(1-q_u(i,1))+p_i) \quad (5.3)$$

5.3.3 Resumption of Flow

The parameter $r_{i,2}$ is the repair probability of machine $M(i,2)$. That is, it is the probability that $M(i,2)$ makes a part at time $t+1$, given it did not make one at time t. This is expressed as

$$r(i,2) = Pr\left[M(i,2) \text{ up at time } t+1 | M(i,2) \text{ down at time } t\right]$$

We define the following quantities:

$$\begin{aligned}
U_1 &= \{\alpha_i(t+1)=1 \cap n_{i-1,1}(t)=0\} \\
D_1 &= \{\alpha_i(t)=0 \cap n_{i-1,1}(t-1)=0\} \\
V_1 &= \{\alpha_i(t)=1 \cap n_{i-1,1}(t-1)=1\} \\
U_2 &= \{\alpha_i(t+1)=1 \cap n_{i-1,1}(t)=N_{i,1}\} \\
D_2 &= \{\alpha_i(t)=0 \cap n_{i-1,1}(t-1)=N_{i,1}\}
\end{aligned}$$

$$V_2 = \{\alpha_i(t) = 1 \cap n_{i-1,1}(t-1) = N_{i,1} - 1\}$$

Then we can refine $r(i,2)$ such that

$$\begin{aligned}
r(i,2) &= Pr\left[U_1 \cup U_2 | D_1 \cup V_1 \cup \cup D_2 \cup V_2\right] \\
&= Pr\left[U_1 | D_1 \cup V_1 \cup \cup D_2 \cup V_2\right] \\
&\quad + Pr\left[U_2 | D_1 \cup V_1 \cup \cup D_2 \cup V_2\right] \\
&= \frac{Pr[U_1|D_1]Pr[D_1] + Pr[U_1|V_1]Pr[V_1]}{Pr[D_1] + Pr[V_1]} \\
&= \frac{Pr[U_2|D_2]Pr[D_2] + Pr[U_2|V_2]Pr[V_2]}{Pr[D_2] + Pr[V_2]}
\end{aligned}$$

From the previous analysis we know that

$$\begin{aligned}
Pr[U_1|D_1] &= r_i(1 - r_u(i-1,1)) \\
Pr[D_1] &= S_d^d(i-1,1) \\
Pr[U_1|V_1] &= \frac{(1 - r_u(i-1,1))P(i-1,1;101) + p_u(i-1,1)P(i-1,1;111)}{P(i-1,1;101) + P(i-1,1;111)} \\
Pr[V_1] &= P(i-1,1;101) + P(i-1,1;111) \\
Pr[U_2|D_2] &= r_i(1 - r_d(i,1)) \\
Pr[D_2] &= B_u^d(i,1) \\
Pr[U_1|V_1] &= \frac{(1 - r_d(i,1))P(i,1;N-110) + p_d(i,1)P(i,1;N-111)}{P(i,1;N-110) + P(i,1;N-111)} \\
Pr[V_1] &= P(i,1;N-110) + P(i,1;N-111)
\end{aligned}$$

Therefore, we can express

$$
\begin{aligned}
r(i,2) &= (r_i(1-r_u(i-1,1))S_d^d(i-1) \\
&\quad +(1-r_u(i-1,1))P(i-1,1;101)+p_u(i-1)P(i-1,1,111)) \\
&\quad \backslash(S_d^d(i-1,1)+P(i-1,1;101)+P(i-1,1;111))
\end{aligned}
$$

Chapter 6

Algorithms and Preliminary Results

6.1 Introduction

It is now necessary to develop an algorithm that will solve the equations. This chapter presents an algorithm, classifies its effectiveness, and provides numerical data for determining its accuracy.

6.2 Algorithm

In this section we present an algorithm for solving the decomposition equations derived in Chapter 4 and Chapter 5. The basic idea of the algorithm is to run DDX algorithm to the Type 1 line, calculating the upstream two-machine parameters for type one, and then apply DDX algorithm for Type 2 line, using the parameter calculated from Type 1 line. We repeat the process until the parameters are converged.

Step 0: Initialization

initialize upstream parameters for Type 1

```
for i = 1 to NumMachines do;
    p_u[i][j] = p[i][1];
    r_u[i][j] = r[i][1];
    q_u[i][j] = 0;
end;
```

initialize upstream parameters for Type 2

```
for i = 1 to NumMachines do;
    p_u[i][j] = p[i][2];
    r_u[i][j] = r[i][2];
    q_u[i][j] = 0;
end;
```

initialize downstream parameters for Type 1

```
for i = 0 to NumMachines-1 do ;
    p_d[i][j] = p[i + 1][1];
    r_d[i][j] = r[i + 1][1];
    q_d[i][j] = p_d[i][1];
end;
```

initialize downstream parameters for Type 2

for i = 0 to NumMachines-1 **do** ;

$p_d[i][j] = p[i+1][2];$

$r_d[i][j] = r[i+1][2];$

$q_d[i][j] = p_d[i][2];$

end;

While the termination criterion of step 5 is not met, do steps one through four, in order, until the termination criterion of step 5 is met, or until the algorithm iterates for a preset number of iterations.

Step 1: Upstream Sweep for Type One

for i = 1 to NumMachines **do**;

Evaluate Two Machine Line $L(i-1, 1)$;

Calculate $q_u[i][1]$ using Equation (??)

Calculate $p_u[i][1]$ using Equation (??)

Calculate $r_u[i][1]$ using Equation (??)

end;

Step 2: Downstream Sweep for Type One

for i = NumMachines-1 to 0 **do**;

Evaluate Two Machine Line $L(i+1, 1)$;

Calculate $q_d[i][1]$ using Equation (??)

Calculate $p_d[i][1]$ using Equation (??)

Calculate $r_d[i][1]$ using Equation (??)

end;

Step 3: Upstream Sweep for Type Two

for i = 1 to NumMachines **do**;

Evaluate Two Machine Line $L(i-1,2)$;

Calculate $q_u[i][2]$ using Equation (??)

Calculate $p_u[i][2]$ using Equation (??)

Calculate $r_u[i][2]$ using Equation (??)

end;

Step 4: Downstream Sweep for Type Two

for i = NumMachines-1 to 0 **do**;

Evaluate Two Machine Line $L(i+1,2)$;

Calculate $q_d[i][2]$ using Equation (??)

Calculate $p_d[i][2]$ using Equation (??)

Calculate $r_d[i][2]$ using Equation (??)

end;

Step 5: Evaluate Stopping Criterion

Terminate the algorithm when the maximum value of

$$||E(i,j) - E(0,j)||$$

for $i = 1, \ldots, NumMachines$ is less than some pre-specified ϵ *for each* part type j.

6.3 Preliminary Results

With the above parameters, the cases were randomly generated using a program written in C. A script then simulated each case, and recorded the results, using a discrete-event simulator written in C. Simulation consisted of 100 independent simulations runs of 1,500,000 time periods each, where the first 500,000 time periods were discarded to ensure data was only collected on a system in steady state. The length of the simulation runs was chosen so that we could be sure that the transient period did not affect the results and that the period of data collection was long enough to give confidence intervals that gave at least three significant digits of accuracy to the right of the decimal place. To do so, we followed a simple procedure of picking a simulation length and transient period at random, and the doubling it until the results from one iteration to the next were essentially identical.

For production rates, we calculated the percent error of the approximated production rate from the simulated production rate in the following manner.

$$\% \text{ Error} = 100 \times \frac{E_{decomp} - E_{sim}}{E_{sim}}. \tag{6.1}$$

This metric is standard in the literature, and provides easy recognition of whether the approximation is under- or over-estimating the simulated production rate.

The average absolute percent error for the 269 cases where the algorithm converged was 0.15% for Type 1 parts, and 4.51% for Type 2 parts.

NO FILE: eps/decomp_case5plot1.eps

Figure 6-1: The errors in the decomposition approximation for Type 1 and Type 2 production rates in a production line with eight processing machines.

Bibliography

[1] Stanley B. Gershwin. *Manufacturing Systems Engineering.* Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[2] Joe Nemec. Multiclass processing line. Master's thesis, Massachusetts Institute of Technology, May 1998.

[3] Tolio T. and A. Matta. A method of performace evaluateion of automated flow lines. 1998. Technical Report.