Scheduling Manufacturing Systems With Work-In-Process Inventory Control: Single-Part-Type Systems

by

X. Bai and S. B. Gershwin

OR 218-90 June 1990

Scheduling Manufacturing Systems With Work-In-Process Inventory Control:

Single-Part-Type Systems

by

X. Bai and S. B. Gershwin

Operations Research Center

and

Laboratory for Manufacturing and Productivity

Massachusetts Institute of Technology

77 Massachusetts Avenue, Cambridge, MA 02139

June, 1990

Scheduling Manufacturing Systems With Work-In-Process Inventory Control:

Single-Part-Type Systems

by

X. Bai and S. B. Gershwin

Abstract

In this paper, a real-time feedback control algorithm is developed for scheduling single-part-type production lines in which there are three important classes of activities: operations, failures, and starvation or blockage. The scheduling objectives are to keep the actual production as close to the demand as possible, and to keep the level of work-in-process inventory as low as possible. By relating the starvation and blockage to the system capacity, the buffer sizes and the target buffer levels are chosen according to the demands and machine parameters.

The processing time for each operation is deterministic. Failure and repair times are random. Whenever a machine fails or is starved or blocked, the scheduling system recalculates short term production rates.

To begin with, we study a very simple case, a two machine and one part type system, to get insight into the buffer effects and production control policies. Using the relationship between system capacity and starvation or blockage, we find desirable buffer levels and buffer sizes. The production control policy is determined to meet the system performance requirements concerning low WIP inventory and tardiness. The results from the simple case are extended to N-machine, one-part-type systems.

Contents

List of Figures

List of Tables

1 Introduction

Manufacturing systems are complex. Large numbers of machines, workers, and part types are often involved. The large number of random events make the scheduling of manufacturing systems difficult. For example, in a semiconductor fabrication factory, hundreds of part types are produced simultaneously by hundreds of workers on dozens of machines. Each part type follows a predefined process which consists of hundreds of operations. Machines are subject to random failures, and need set-up changes for different part types. Maintenance and rework must be considered. Workers are absent at random. These factors result in long throughput time, large work-in-process (WIP) inventory, and significant tardiness.

Tardiness is the time difference between actual production and demand. If the actual production is ahead of the demand (negative tardiness or earliness), final product inventory accumulates. If the actual production is behind the demand (positive tardiness), customers are unsatisfied, and sales may be lost.

Throughput time (sometimes called cycle time or lead time) is the time that a part spends in the system. The shorter the throughput time is, the faster the system can respond to customer orders, and the sooner that tardiness can be reduced. Throughput time consists of waiting times in buffers and processing times on machines.

Work-in-process (WIP) inventory is the number of unfinished parts in the system, which consists of the material in buffers and the pieces being processed on machines. The less the WIP, the shorter the throughput time. However, too little inventory will reduce the system capacity, which will increase the tardiness.

To improve the efficiency of production, we would like to reduce inventory, throughput time, and tardiness simultaneously. In this paper, a real-time feedback control algorithm is developed for scheduling manufacturing systems. The scheduling objectives are to keep the actual production as close to the demand as possible, and to keep the level of WIP as low as possible.

1.1 The role of WIP in manufacturing systems

WIP inventory in manufacturing systems is not always bad. It is usually regarded as a bad thing because it takes space, costs money for handling, and increases the throughput time. In addition, parts must pass through several operations before being inspected. If an operation produces defective parts, and there is much WIP inventory between operations, many parts will be produced before the faulty operation

is discovered. But inventory does have some properties from which the production managers can benefit. They are listed in the following.

Operation independence: In serial production systems, two machines in series without intervening WIP must be perfectly synchronized to operate effectively. Otherwise, even if they have the same average variable processing times, the first machine sometimes finishes an operation before the second. The first must wait to unload the finished piece before it begins the next piece. Putting a buffer and some amount of WIP between the two machines will provide independence of their operations. The two machines do not have to finish operations at exactly the same instant to operate effectively.

Breakdown impact absorption: In real manufacturing systems, all machines are subject to random failures. In the case of two machines in series without a buffer between them, if first machine is broken, the second machine will be starved after it finishes its current operation since there is no part available for it to work on next. Similarly, if the second machine is down, the first machine will be blocked when it finishes its current operation since there is no space to unload the finished part. However, putting a buffer and some amount of WIP between the two machines allows an operation to continue when the another machine is down.

Setup changes: WIP inventory allows two machines in series to work on different part types, even if there is a significant setup time required to change from one part type to another.

Spatial decomposition: The huge sizes of manufacturing systems and the variety of random events involved are always hard to deal with in real-time decision making. WIP inventory allows a system to be divided into several sub-systems and to be scheduled separately to some extent. It is like warehouses between factories.

Thus, the WIP inventory in a manufacturing system affects significantly the throughput time and the tardiness. The properties of WIP and the effects of buffers in manufacturing systems have drawn a lot of attention and interest from researchers. In this paper, we study how WIP can be used with a sophisticated control policy. One key question is: what is the minimal necessary WIP and how should it be distributed in a manufacturing system to make the production effective?

1.2 Previous research

There is a large body of literature in production scheduling. Much of it is surveyed in Graves [1]. Many of the works before the early 80's are based on combinatorial

optimization/ integer programming or mixed integer methods ([2], [3], [4], [5], [6], [7], and [8]). Some other works are based on queuing network models ([9], [10], and [11]).

Since the large number of machines, workers, part types, and operations are involved in real production systems, hierarchical structures have been proposed for production control in order to reduce the problem size and complexity ([12], [13], [14], [15], and [16]). The goal is to replace one large problem by a set of many small ones because latter is invariably easier to solve. Even still, the variety of random events associated with the manufacturing procedures make the traditional optimization methods, in many cases, inadequate or inappropriate for production scheduling, especially in real time.

Since the early 80's, production flow models have been developed to further reduce the complexity of the scheduling problems. In those formulations, the part movement in a production system is treated as continuous flow so that the dimension of the model is reduced dramatically. Furthermore, the system dynamics of the production flow models are in a form that is appropriate for control theory and techniques.

Using Rishel's methodology [17], Kimemia and Gershwin [18] investigated the optimal flow controller's structure and determined that it is a hedging point feed-back control policy. Tsitsiklis [19] proved the convexity of the value function that satisfies the Hamilton-Jacobi-Bellman equations and determines the optimal controller. Gershwin, Akella, and Choong [20] proposed a heuristic approximation of the value function. Akella and Kumar [21] solved analytically the Hamilton-Jacobi-Bellman equation to obtain the optimal value function for a simple one-part-type, one-machine system. Van Ryzin [22] studied the delay of the production flow in a buffer and obtained a numerical solution for a one-part-type, two-machine system. In short, much effort has been directed to the development of the production flow control models, for both analytical solutions and approximation methods. (Also see [23], [24], [25], [26], and [27].)

Work-in-process (WIP) inventory plays a very important role in production scheduling. It has drawn a great deal of attention from researchers. Conway et al. [28] studied the effects of WIP in serial production lines. Burman et al. [29] investigated the relation between the WIP level and the system performance of integrated circuit manufacturing lines. Zeghmi studied inventory buffers in a producion line [30]. Because of the complex way that WIP interacts with all the random events, the WIP control in a dynamic environment, such as real-time scheduling production systems, is still not well solved and understood.

1.3 Results of the paper

A real-time feedback control algorithm is developed for scheduling single-part-type production lines in this paper. Three important classes of activities are considered. They are operations, machine failures, and starvation or blockage. The buffer levels and sizes (and therefore the WIP) are allocated according to the demands and machine parameters, by solving a non-linear program.

To begin with, we study a very simple case, a two-machine, one-part- type system, to get insight into the buffer effects and production policies. Using the relation between system capacity and starvation or blockage, we find the most desirable buffer level and size. The production control policy is determined to meet the system performance requirements concerning low WIP inventory and tardiness. The result from the simple case is extended, then, to N-machine, one-part-type systems.

The method developed in this paper is extended to multiple-part-type and reentrant systems in [31] and [32]. The algorithm can also be modified such that the formulation for buffer levels and sizes becomes linear [33].

The results of this paper are an extension of those by Kimemia and Gershwin [18] and Van Ryzin [22].

1.4 Outline of the paper

Section 2 describes the systems which are under study. The system assumptions are described. Notation and terminology are defined there. The activities, constraints, and objectives are discussed. In Section 3 we study the two-machine, one-part-type system. In Section 4 we study N-machine, one-part-type systems. The results are summarized in Section 5.

2 The model of a manufacturing system

In this section, we introduce a model of a production line. In Section 3, we construct a real-time scheduler for the simplest example of this model. In Section 4, we construct a controller for N-machine, one-part-type systems.

2.1 Time

In a manufacturing system, many measurements are based on different time frames. For instance, the time to fail is measured only when a machine is operational, and the frequency of failure is based on the time during which the manufacturing facility is functional for production activities. Usually, a measurement is defined only on a specified time frame. It often becomes practically meaningless when the underlying time frame is changed. In the following, we define three time frames and the complementary frames associated with them.

2.1.1 Absolute time

Absolute time is also called clock time. It is identical to the time measured by a clock. Define T_a to be the set of absolute time, which satisfies

$$T_a = \{t \in \mathbf{R} \bigcup \{-\infty, +\infty\}\} \tag{1}$$

Define C_a to be the complement of the absolute time, which is an empty set:

$$C_a = \emptyset$$

2.1.2 Working time

Working time is a subset of absolute time. It is the time during which the manufacturing system is functional for production activities. Since most manufacturing facilities are closed on holidays and some are run only one or two shifts a day, it is convenient, sometimes, to make measurements based on the working time.

Define T_w to be the set of working time, which satisfies

$$T_w = \{t \in T_a | \text{the system is functional for production}\} \tag{2}$$

Define C_w to be the complement of the working time, which satisfies

$$T_w + C_w = T_a$$

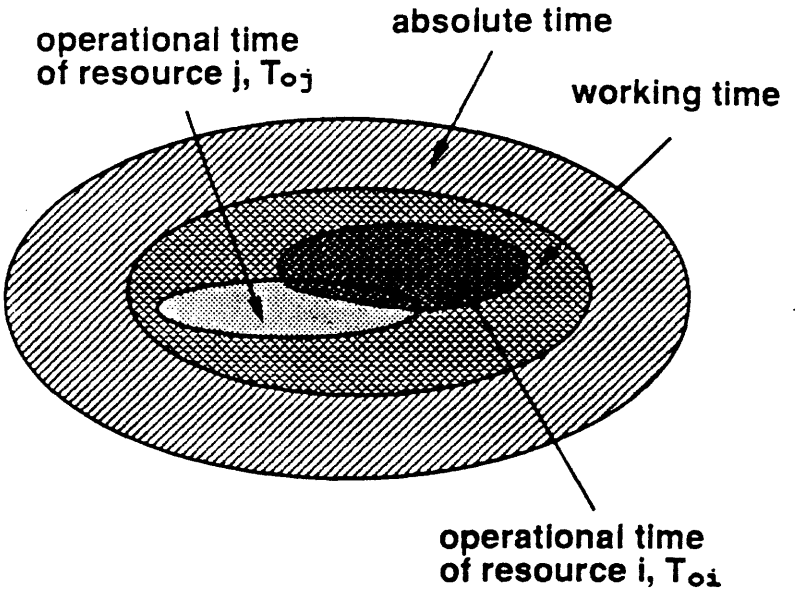

Figure 1: The relationship among the frames of time

In general, Working time consists of shift hours and overtime. The complement of the working time consists of holidays and off time (such as weekends).

2.1.3 Operational time

Operational time is a subset of the working time. It is associated with an individual resource. Operational time is the time during which the resource is able to perform production activities. A machine can change setup, or produce parts only when it is operational. Most resources in a manufacturing system subject to disruptions. For example, all machines fail randomly and need preventative maintenance. Operational time is needed to define some important quantities, such as starvation fraction and blockage fraction since a machine can be starved or blocked only when it is operational.

Define T_{oi} to be the set of operational time associated with Resource i, which satisfies

$$T_{oi} = \{t \in T_w | \text{Resource i is operational}\} \quad (3)$$

Define C_{oi} to be the complement of the operational time of Resource i within T_w, which satisfies

$$T_{oi} + C_{oi} = T_w$$

2.1.4 The relationship among the time frames

Fig.1 illustrates the relationship among the three time frames defined above. The absolute time, T_a, is the universal set, which consists of the working time, T_w, and its complement, C_w. The operational time, T_{oi}, and its complementary, C_{oi}, are complementary subsets of working time.

Since we do not consider issues like holidays and over time, in this paper, we assume that the absolute time and working time are identical. That is, we assume that $C_w = \emptyset$. In the following, the time axis is the working time unless we redefine it explicitly.

2.2 Material flow

For each part type, the parts go through the system following a predefined operation sequence. The operation sequence is called a process, which contains the route and operation information. For example, a semiconductor fabrication process consists of the machine name, recipe number, processing time, temperature, gas configuration, and so on, for each operation [34].

To reduce the complexity of the problem, we model the movement of parts in the system as a continuous flow. We model a machine as a valve with a switch which is randomly on and off. When the switch is on, the material flows through the machine. The flow is incompressible so no material can be accumulated in the machine. Therefore, at any given time, the flow rates at the two ends of the machine must be the same. A buffer can be viewed as a tank in which material is allowed to accumulate.

In contrast, material is allowed to accumulate inside a machine in the compressible flow model. In this case, the production flow is not only delayed in buffers, but also in machines. The compressible flow model is appropriate for "pizza-oven-type" machines. That is, a machine can handle more than one job at a time, and the processing times may be different. An individual part can be loaded or unloaded when the machine is processing some other parts. Van Ryzen [22] studied the phenomena of delay in machines as well as in buffers.

2.3 Resources

A resource is any part of the manufacturing system that is used to perform or to support an operation. Machines, buffers and workers are resources. By function, ma-

chines are divided into two groups: operation machines and support equipment [34]. The operation machines are those which perform operations directly. The support equipment, such as the DI water and gas supplies in a wafer fabrication factory, are never visited by parts.

We assume that the human resources and the support equipment are always available. Consequently, the operation machines and buffers are the resources which impose capacity constraints to the scheduling problem. However, the methods developed in this paper can be extended to take the human resources and the support equipment into account. In the following, we will simply use "machine" to indicate an operation machine.

2.3.1 Machines

The manufacturing system under study includes a total of N machines. All machines are subject to random failures and need random repair times. Define $\alpha_i(t)$ to represent the state of Machine i ($i = 1, 2, \ldots, N$). It is a binary variable which is 1 if the machine is operational and 0 otherwise. We define the machine state vector

$$\alpha(t) = (\alpha_1(t), \ldots, \alpha_N(t)).$$

Given that Machine i is operational, the probability of a failure in an interval of length δt is $p_i \delta t$. The probability that a failed machine is repaired during a δt time interval is given by $r_i \delta t$. The parameters p_i and r_i are the failure and repair rates for machine i ($i = 1, 2, \ldots, N$). The dynamics of the machine state are therefore governed by

$$\begin{aligned} p[\alpha_i(t+\delta t) = 1 | \alpha_i(t) = 0] &= r_i \delta t \\ p[\alpha_i(t+\delta t) = 0 | \alpha_i(t) = 1] &= p_i \delta t \\ (i = 1, 2, \ldots, N) & \end{aligned} \tag{4}$$

For Machine i, the time to fail is thus modeled by exponentially distributed random variable with mean $1/p_i$, which is measured in the frame of operational time, T_{oi}. The time to repair is also an exponentially distributed random variable with mean $1/r_i$, which is defined on the complement of the operational time, C_{oi}. These two random variables are independent. The average time interval during which Machine i is up once and down once is measured in the working time frame. The length of such an interval is $1/r_i + 1/p_i$.

Define F_i to be the frequency of failure of Machine i, which is given by

$$F_i = \frac{1}{\frac{1}{r_i} + \frac{1}{p_i}} = \frac{r_i p_i}{r_i + p_i}$$

The model assumes that machine failure rates do not depend on the part flow rates, starvation, or blockage. That is, we assume time-dependent, rather than operation-dependent failures.

We also assume that all machines are flexible enough so that we can neglect setup change times, and that the preventative maintenance activities do not occur on the time scale of repairs and failures. The only activities that we are considering are operations and failures and the effects of emptying and filling of buffers. (The term activity is defined in Section 2.4.)

Finally, we assume that the frequency of operations is an order of magnitude greater than the frequency of failures, and that the durations of operations are an order of magnitude less. We focus our attention on the time scale in which individual failures are important, but individual operations are not. Thus we study production rates, and approximate cumulative productions and buffer levels as continuous quantities (and represent them with real numbers rather than integers).

2.3.2 Buffers

There is only one part type in the system. A total of N operations are required. The i^{th} operation is performed on the i^{th} machine. We assume that there are buffers between every two consecutive operations. Therefore, there are (N-1) buffers to store the work-in-process (WIP).

Let Buffer i be the buffer between the i^{th} and $i+1^{th}(i = 1, 2, \ldots, N-1)$ operation. We use b_i to represent the buffer level, i.e., the number of parts in Buffer i. Since we represent material as continuous flow, b_i is a real number, not restricted to the integers.

The buffer level b_i is a part of the WIP inventory. It is a key factor for production control. The buffer level is directly related to how long the production can last without starving the adjacent downstream machine when a machine is down. If the downstream machine is starved too much, the production demand cannot be satisfied, and if the downstream machine is starved too little, then excess WIP inventory exists.

Define u_i to be the production rate of the i^{th} operation at Machine i. The dynamics

of the buffer level are governed by

$$\dot{b}_i = u_i - u_{i+1} \tag{5}$$

$$(i = 1, 2, \ldots, N-1)$$

Define B_i to be the size of Buffer i. The buffer size B_i is not necessarily the physical buffer size. It is a control parameter (which we determine below) which is used as a threshold to block the upstream machine. We choose it to limit WIP when more WIP does not lead to better performance. Although the model can be easily extended to include physical buffer size, to focus our attention to WIP inventory allocation, we assume that there is an unlimited amount of physical space for each buffer in the system. That means that there is no upper limit for B_i. B_i and b_i must satisfy

$$B_i \geq b_i \geq 0 \tag{6}$$

$$(i = 1, 2, \ldots, N-1).$$

Define s_i to be the empty space in Buffer i, which satisfies

$$b_i + s_i = B_i$$

$$(i = 1, 2, \ldots, N-1)$$

Here, we would like to emphasize that the empty space s_i is equally important as the buffer level b_i for production control. The empty space s_i is the decisive factor that determines how long the production can last without blocking the adjacent upstream machine when a machine is down. If the upstream machine is blocked too much, the demand cannot be achieved, and if the upstream machine is blocked too little, excess WIP inventory is accumulated.

2.4 Activities

An activity is a pair of events associated with a resource [14]. The first event corresponds to the start of the activity, and the second is the end of the activity. Only one activity can appear at a resource at any time. The three important classes of activities which are included in the model are listed as follows:

Operations: The major activities in a manufacturing system are production operations. It takes a certain amount of time to perform an operation on a machine.

Sometimes, operation times are random. However, for highly automated machines, the variances of operation times are usually very small. We treat the operation times as deterministic. Production operations are controllable activities. That is, the decision-maker can decide when and where to perform the activities, as long as machines are not occupied with any other activities.

Machine failure and repair: All machines are subject to random failures and need random repair times. Two parameters, the mean time to fail (MTTF) and the mean time to repair (MTTR), are used to describe the failures and repairs for each machine. The $MTTF_i$ is the average length of the time period from a repair to the next failure, which is measured in the operational time frame, T_{oi}. The $MTTR_i$ is the average length of the time period from a failure to next repair, which is measured in the complementary frame of the operational time, C_{oi}. Machine failure and repair are uncontrollable and unpredictable activities. We assume that the time to fail and the time to repair are exponentially distributed random variables and that failures may occur even when the machine is not being used (time-dependent failures).

Starvation and blockage: A machine is starved when it is idle because there are no parts in its upstream buffer. A machine is blocked when it is idle because its downstream buffer is full. Starvation and blockage are uncontrollable and unpredictable activities. That is, the decision-maker cannot know in advance when and where starvation or blockage will occur.

We assume that the first machine is never starved due to a shortage of the raw parts, and the last machine is never blocked due to a lack of space to unload the final product. *A machine can be starved or blocked only when it is operational.*

Define f_i^b to be the *blockage fraction*. It is the fraction of operational time, T_{oi}, during which Machine i is operational and blocked.

Define f_i^s to be the *starvation fraction*. It is the fraction of operational time, T_{oi}, during which Machine i is operational and starved.

The starvation and blockage fractions are quantities defined in the frame of operational time. We show in Section 3 that the starvation and blockage fraction are functions of machine parameters, buffer levels, buffer sizes, and production demands.

2.5 Constraints

Production is subject to many constraints. Some of them are common to all manufacturing procedures, such as capacity constraints and feasible demand constraints. Others only can appear in specific manufacturing environments, such as the limited furnace chamber size constraint in semiconductor fabrication. In our model, two kinds of constraints are considered. They are

Capacity constraints: It takes a certain amount of time for a machine to perform an operation and a machine is only available for so many hours a day. The production rates are constrained by the current capacity of the system.

Define τ_i to be the processing time of the i^{th} operation on Machine i; and u_i to be the production flow rate of the i^{th} operation through Machine i at time t. The current or instantaneous capacity is then defined by

$$\tau_i u_i \leq \alpha_i \qquad (i = 1, 2, \ldots, N). \tag{7}$$

As a machine fails or is repaired, i.e., as the machine state changes, the set of feasible instantaneous production rates change.

An instantaneous production rate is feasible only if it is a member of the capacity constraint set

$$\Omega(\alpha) = \{u_i, i = 1, 2, \ldots, N \mid \tau_i u_i \leq \alpha_i, \quad \text{for all i and} \quad u_i \geq 0\}. \tag{8}$$

Note that the capacity set is independent of the control policy. That is, the system can at most have so much capacity no matter what kind of control policies we use.

Operation sequence constraints: We assume that for each part type, operations have to be performed one after another following a pre-defined sequence. That means that there is only one path for each part type to go through the system. We do not consider the multiple route case here.

2.6 Problem feasibility

A manufacturing system has certain capacity. It only can achieve demand within a limited range. This range represents the long term capacity of the system. It is useful information for long term planning and marketing decisions.

By taking the time average of (7), we have

$$\frac{1}{T}\int_0^T \tau_i u_i dt \leq \frac{1}{T}\int_0^T \alpha_i dt \quad (i = 1, 2, \ldots, N). \tag{9}$$

If the system is ergodic and in steady state, then

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T \alpha_i dt = \frac{r_i}{r_i + p_i} \quad (i = 1, 2, \ldots, N). \tag{10}$$

Let

$$\bar{u} = \lim_{T\to\infty} \frac{1}{T}\int_0^T u_i dt$$

Plugging (10) into (9), the long term capacity is given by

$$\tau_i \bar{u} \leq \frac{r_i}{r_i + p_i} \quad (i = 1, 2, \ldots, N). \tag{11}$$

The long term capacity set is

$$\bar{\Omega} = \{\bar{u} \mid \tau_i \bar{u} \leq \frac{r_i}{r_i + p_i}, \quad \text{for all i and } \bar{u} \geq 0\}. \tag{12}$$

Define d to be the production demand which usually is a function of time. We assume that the frequency of the demand change is an order of magnitude smaller than that of failures. That is, the amount of time during which the system is in steady state is much greater than its time in transient states. A demand is feasible only if it is a member of the long term capacity constraint set (12).

2.7 Objectives

In different manufacturing environments, the production control objectives may be different. We emphasize the following objectives:

Tardiness and inventory: To increase sales and keep good business relations with customers, we want to deliver products on time. At the same time we do not want excess inventory. Consequently, we must keep production close to demand.

Define x_i to be the production surplus of the i^{th} operation on machine i, which satisfies

$$\dot{x}_i = u_i - d \quad (i = 1, 2, \ldots, N) \tag{13}$$

Define the production surplus vector

$$x(t) = (x_i, i = 1, \ldots, N).$$

It should be noticed that the production surplus is not the same as WIP inventory. The production surplus is the cumulative difference between production and demand. Large surplus does not always indicate high WIP inventory. Also, the surplus can be negative (backlog) but WIP cannot.

For the final operation, if the surplus x_N is positive, more material has been produced than is required. This surplus or safety stock helps to reduce the impact of machine failures. However, it has a cost. Expensive floor space and a material handling system must be devoted to storage. In addition, working capital has been expended in the acquisition and processing of stored materials. This capital is not recovered until the final product inventory is sold. If the surplus x_N is negative, there is a backlog, which is even more costly. Backlog represents unsatisfied customers. In this case, sales and goodwill may be lost.

It should also be noticed that the production surplus is different from tardiness. Tardiness is the time difference between the due date and the actual shipment from the system. Since we represent material as continuous flow, tardiness is a continuous variable. Fig.2 illustrates the relations among demand, production, surplus, and tardiness, at the final stage of a production process. Positive surplus always indicates negative tardiness (actual shipment is ahead of due date). Negative surplus always indicates positive tardiness (actual shipment is behind due date). Most often, large surplus indicates large tardiness. At any given time, the production surplus is independent of the future production. However, when production surplus is negative, the tardiness depends on the future production (see Fig.2). This is the major reason that we choose the production surplus as the feedback variable instead of tardiness.

The objective of minimizing the tardiness and final product inventory is equivalent to minimizing the absolute value of the surplus of the final operation, x_N. That is because both objectives minimize the area between the actual production and demand.

The work-in-process (WIP) inventory: Whenever possible, we want to keep WIP inventory in a manufacturing system as small as possible, because it takes space, costs money for handling, and increases the throughput time. However, too little work-in-process inventory will increase starvation and blockage, and therefore reduce production rates.

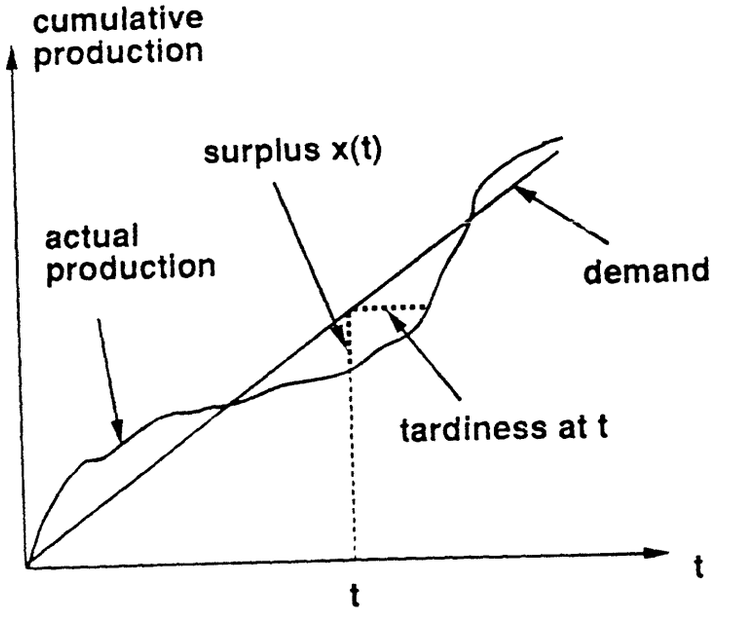

Figure 2: Demand, production, surplus, and tardiness

The work-in-process inventory consists of the parts in buffers and the working pieces on machines. Minimizing WIP inventory is equivalent to choosing the smallest buffer sizes and average buffer levels such that we just have enough capacity to achieve the demand.

The throughput time: This is the time a part spends in the system. It is also called cycle time or lead time. The shorter the throughput time is, the faster the system can respond to customer orders, and the faster the firm can develop new products and processes. The throughput time consists of the waiting times in buffers and the processing times on machines. The waiting times in buffers are proportional to the buffer levels. Consequently, minimizing the buffer levels and buffer sizes also minimizes throughput time.

Therefore, we formulate an optimization problem in which we minimize the average WIP level and are constrained to meet demand. The decision variables are the instantaneous production rates and buffer levels.

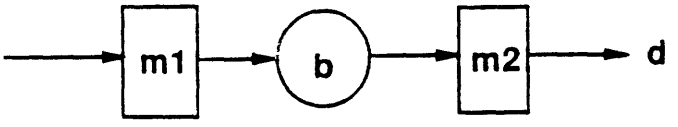

Figure 3: Two-machine, one-part-type system

3 Two machine, one part type system without reentry

In this section, we study the simplest case, the two-machine, one-part-type system. The results in this section will be extended to N-machine, one-part-type systems in Section 4 and more general systems in [33], [31], and [32].

As illustrated in Fig.3, the system consists of two machines (i=1,2) and one buffer. For Machine i, the failure rate is p_i and the repair rate is r_i. The mean time to fail $MTTF_i = 1/p_i$ and the mean time to repair $MTTR_i = 1/r_i$. One part type is produced. Each part needs an operation with processing time τ_1 on Machine 1 and an operation with time τ_2 on Machine 2. A buffer is located between the two machines. We assume that Machine 1 is never starved and Machine 2 is never blocked.

3.1 Dynamic optimization

The production flow rate control can be formulated as a dynamic optimization problem. Given an initial surplus state $x(t_0)$, and machine state $\alpha(t_0)$, we wish to specify a feedback control strategy for production during $t_0 \leq t \leq T$ that satisfies

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \qquad (14)$$

subject to:

$$\tau_1 u_1 \leq \alpha_1$$
$$\tau_2 u_2 \leq \alpha_2$$
$$u_1 \geq 0, u_2 \geq 0$$

where the system dynamics are

$$\begin{aligned}\dot{x}_1 &= u_1 - d \\ \dot{x}_2 &= u_2 - d \\ \dot{b} &= u_1 - u_2 \\ B &\geq b \geq 0\end{aligned}$$

in which B is the buffer size to be determined. The constraints are specified in the form of $u \in \Omega(\alpha)$, where $\Omega(\alpha)$ is given by (8). The function $g(x, b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive or too negative.

Assume that the initial buffer level $b(t_0)$ satisfies

$$b(t_0) = x_1(t_0) - x_2(t_0) \tag{15}$$

The buffer level, b, is a function of the surplus x, which can be determined by (5), (13), and (15),

$$b(t) = x_1(t) - x_2(t) \tag{16}$$

Therefore, by plugging (16) and (15) into (14), the cost-to-go J is not an explicit function of the buffer level, $b(t)$.

It is impossible to solve this dynamic optimization problem analytically. A numerical solution was obtained by Van Ryzin [22] for the two-machine, one-part-type case. It was shown that the production surplus x-space is divided into regions. For each α, each region in x-space corresponds to a specific production decision rule.

Unfortunately, the numerical method is very time consuming and not efficient enough to be extended to more complicated systems. Instead, we develop an approximation method to solve the production control problem, which can be extended for more complicated systems.

3.2 Feedback control law and the quadratic J function

Based on the methods of Kimemia and Gershwin [18], it can be verified that the optimal production flow rate, u, can be determined, if the optimal cost-to-go (or value function) $J(x, \alpha, t_0)$ is known, by solving the linear programming problem

$$\min_u \{\frac{\partial J}{\partial x_1} u_1 + \frac{\partial J}{\partial x_2} u_2\} \tag{17}$$

subject to:

$$\begin{aligned}\tau_1 u_1 &\leq \alpha_1 \\ \tau_2 u_2 &\leq \alpha_2 \\ u_1 \geq 0, u_2 &\geq 0\end{aligned}$$

where

$$\dot{x}_1 = u_1 - d$$
$$\dot{x}_2 = u_2 - d$$
$$\dot{b} = u_1 - u_2$$
$$B \geq b \geq 0$$

The solution of (17) is a real-time feedback control law since the LP is determined by x and α. For complicated manufacturing systems, it is impossible to get the closed form optimal cost-to-go $J(x, \alpha, t_0)$. As an approximation, we present a quadratic function for the cost-to-go of the linear programming problem (17) and we assume that J is not a explicit function of α. Not only does this reduce data requirements and simplify the computation of the production flow rates , but it also makes it possible to solve the production control problem for more complicated systems.

Let $\tilde{x} = (x_1, x_2, b)^t = (x_1, x_2, x_1 - x_2)^t$. The value function is

$$J = \frac{1}{2}\tilde{x}^t A \tilde{x} + C^t \tilde{x} + D \tag{18}$$

where A is a 3×3 positive semi-definite and symmetric matrix, C is a 3×1 vector, and D is a scalar.

Let

$$L_1(x_1, x_2) = \frac{\partial J}{\partial x_1} = (1 \quad 0 \quad 1)A \begin{pmatrix} x_1 \\ x_2 \\ x_1 - x_2 \end{pmatrix} + C^t \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \tag{19}$$

$$L_2(x_1, x_2) = \frac{\partial J}{\partial x_2} = (0 \quad 1 \quad -1)A \begin{pmatrix} x_1 \\ x_2 \\ x_1 - x_2 \end{pmatrix} + C^t \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \tag{20}$$

Note that $L_1(x_1, x_2)$ and $L_2(x_1, x_2)$ are linear functions of production surplus x_1 and x_2.

Plugging (19) and (20) into (17), the linear programming problem becomes

$$\min_u \{L_1(x_1, x_2)u_1 + L_2(x_1, x_2)u_2\} \tag{21}$$

subject to:

$$\tau_1 u_1 \leq \alpha_1$$
$$\tau_2 u_2 \leq \alpha_2$$
$$u_1 \geq 0, u_2 \geq 0$$

where

$$\dot{x}_1 = u_1 - d$$
$$\dot{x}_2 = u_2 - d$$
$$\dot{b} = u_1 - u_2$$
$$B \geq b \geq 0$$

By inspecting (21), we can make the following observations.

Observation 1: The coefficients of u are functions of x. Consequently, (21) divides the x-space into mutually exclusive regions (Fig.4). For instance, suppose that both machines are operational. If both coefficients are positive, the solution will be $u_1 = 0$ and $u_2 = 0$; if the coefficient of u_2 is positive and the other is negative, the solution will be $u_1 = 1/\tau_1$ and $u_2 = 0$; and so on. We can thus define regions of x-space in which the production rate u is constant. Because the coefficients are linear functions of x, the boundaries are straight line segments. When $x(t)$ is on a boundary, u is also constant. Consequently, the linear program yields a piecewise constant solution, $u(t)$.

Observation 2: The production rates $u(t)$ do not have to be calculated at every time instant. They need only to be computed when α changes or when $x(t)$ reaches a boundary.

When both machines are operational, the solution of (21) in x-space is illustrated in Fig.4. The straight line which goes through the origin is the zero buffer boundary, which is the set of points in which the buffer is empty. The straight line which is parallel to the zero buffer boundary is the full buffer boundary, which is the set of points in which the buffer is full. Since we have to respect (6) and (16), the feasible solution lies between the zero buffer boundary ($b = 0$) and the buffer size boundary ($b = B$). The other two boundaries are the coefficient boundaries, which are the sets of points in which one of the coefficient functions of u_1 or u_2 in (21) is zero. The coefficient boundaries divide the feasible area into four regions. In Region 1, the coefficient functions are both positive, so the production decision is to shut down both machines. In Region 2, Machine 1 produces at its maximum rate, while Machine 2 produces nothing. In Region 3, both machines produce at their maximum rates. In Region 4, Machine 2 produces as much as possible, while Machine 1 produces nothing.

The intersection of the coefficient boundaries is called the *hedging point*, which is the desirable operating state of the system. The feedback controller (21) always attempts to drive the system to the hedging point, and to stay there. Since we have chosen a quadratic cost-to-go J which is not a function of α, then the hedging point is not a function of α.

Both the coefficient boundaries are attractive. (See [20] for a similar problem.)

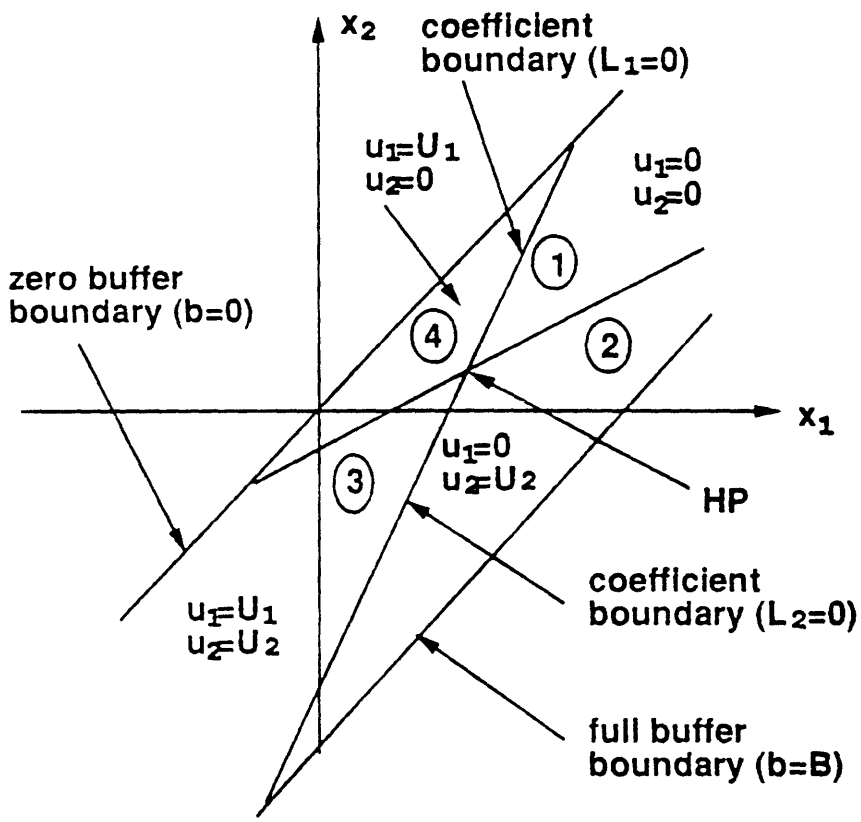

Figure 4: The linear program solution in x-space

That is, when the system state reaches a boundary, it moves along the boundary towards the hedging point. The solutions on the boundaries are different from those in the interior of the regions. That difference leads to a set of conditional constraints which we discuss later in this section. In the following text, we will use "boundary" to indicate the coefficient boundary.

A similar graph as Fig.4 was obtained by Van Ryzin from the numerical solution of the dynamic optimization problem (14) [22]. The quadratic J function was not used in that work.

3.3 System behavior and performance specification

In the previous section, we introduced the quadratic approximation of the optimal cost-to-go J and obtained a feedback controller in the form of linear programming problem. But the coefficients of the quadratic J function, namely A, C, and D, are unknown. Different sets of coefficients correspond to different boundary positions in the x-space. Consequently, the system behaves differently.

To choose the unknown parameters of the quadratic J function, in this section, we specify the desirable system behavior and performance requirements which reflect the scheduling objectives. In the following sections, we determine the coefficients of the quadratic J function such that the system behaves as specified.

The desirable system behavior and performance specifications are:

(a) When Machine 1 fails, keep Machine 2 producing without changing its production plan until the buffer is empty.
(b) When Machine 2 fails, keep Machine 1 producing without changing its production plan until the buffer is full.
(c) Keep the absolute value of x_N as small as possible. That is, keep the production close to demand.
(d) Keep the buffer size and average buffer level as small as possible. That is, keep WIP low and throughput time short.

The behavior requirements (a) and (b) are the considerations of spatial decomposition. For example, when a single machine fails, we do not want to change the production plan of other machines unless we have to. Consequently, we would like to separate the machines as much as possible to reduce the effects of machine failures. This consideration is essential for dividing a system into several sub-systems in a

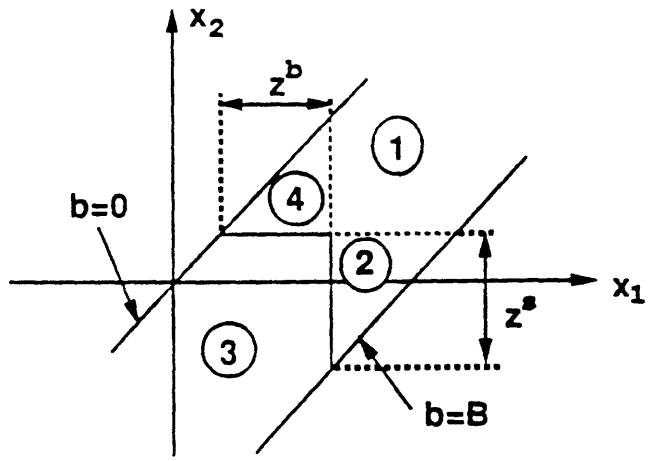

Figure 5: The desirable boundary shape in x-space

hierarchical structure. In this case, the state of Machine 1 does not affect Machine 2 if the buffer is not empty, and the state of Machine 2 does not affect Machine 1 if the buffer is not full.

It is important to note that these specifications are not necessarily optimal according to (14). They are heuristic approximations. In latter sections, we will see that these specifications reduce the complexity of the production control problem.

3.4 The desirable boundary shape in x-space

By using the system behavior requirements (a) and (b) of the previous section, we can determine the desirable boundary shape in x-space. When the system reaches the hedging point (the intersection of the coefficient boundaries), it will stay there if no failure occurs. This is because that the system attempts to move towards the intersection all the time (see Fig.4). Therefore, when the system stays at the hedging point, the production flow rates are equal to the demand since the surpluses do not change.

Suppose that the system has reached the hedging point. The production rate decision is then $u_1 = u_2 = d$, so the system stays at the hedging point indefinitely. When Machine 1 fails ($\alpha_1 = 0$) and Machine 2 is operational ($\alpha_2 = 1$), according to the behavior requirement (a) and the capacity constraints (8), the production decision

should be

$$u_1 = 0$$
$$u_2 = d \qquad (22)$$

until the buffer is empty. Before the buffer is empty, x_1 decreases at rate d and x_2 is constant. Therefore the system state (x_1, x_2) moves along a horizontal line towards the zero buffer boundary ($x_1 = x_2$). Since the solution (22) is different from those in the interior of the regions, one of the boundaries must be horizontal and must go through the hedging point (see the discussion at the end of Section 3.2).

Similarly, the other boundary must be vertical and must go through the hedging point.

The desirable boundary shape in x-space is illustrated in Fig.5. Regions (1), (2), (4) and the dashed boundaries are the transient states. This is because, after the system reaches the hedging point, it will never go above the horizontal boundary and to the right of the vertical boundary. On the horizontal boundary, x_2 is constant ($u_2 = d$). On the vertical boundary, x_1 is constant ($u_1 = d$). When the buffer is empty, Machine 2 cannot produce faster than Machine 1. When the buffer is full, Machine 1 cannot produce faster than Machine 2.

3.5 The conditional constraints

As we mentioned earlier, the solutions on the boundaries are different from those in the interior in the x-space. These properties are modeled as conditional constraints of the linear program.

Define (z_1, z_2) to be the hedging point which is the desirable value of (x_1, x_2). The components of the hedging point are unknown and will be determined by the performance specifications (c) and (d) of Section 3.3. This is described in Section 3.9.

The conditional constraints are

$$\begin{array}{lll} \text{if } x_1 = z_1, & \text{then } u_1 = d \\ \text{if } x_2 = z_2, & \text{then } u_2 = d \\ \text{if } b = 0. & \text{then } u_1 \geq u_2 \\ \text{if } b = B, & \text{then } u_1 \leq u_2 \end{array}$$

which say that when the system reaches the vertical boundary, the production rate of Machine 1, u_1, should be equal to the demand, d. When the system reaches the horizontal boundary, the production rate of Machine 2, u_2, should be equal to the demand, d. When the buffer is empty, Machine 2 cannot produce faster than Machine 1. When the buffer is full, Machine 1 cannot produce faster than Machine 2.

3.6 The linear program for real-time feedback control

To ensure that the coefficient boundaries in x-space are horizontal and vertical and go through the hedging point, linear program (21) becomes

$$\min_{u}\{a_1(x_1 - z_1)u_1 + a_2(x_2 - z_2)u_2\} \tag{23}$$

subject to:

$$\begin{aligned} &\tau_1 u_1 \leq \alpha_1 \\ &\tau_2 u_2 \leq \alpha_2 \\ &u_1 \geq 0, u_2 \geq 0 \end{aligned}$$

$$\begin{aligned} &\text{if } x_1 = z_1, \quad \text{then } u_1 = d \\ &\text{if } x_2 = z_2, \quad \text{then } u_2 = d \\ &\text{if } b = 0, \quad \text{then } u_1 \geq u_2 \\ &\text{if } b = B, \quad \text{then } u_1 \leq u_2 \end{aligned}$$

where

$$\begin{aligned} \dot{x}_1 &= u_1 - d \\ \dot{x}_2 &= u_2 - d \\ \dot{b} &= u_1 - u_2 \\ B &\geq b \geq 0 \end{aligned}$$

in which z_1 and z_2 are the unknown components of the hedging point which are to be determined later.

Comparing (23) and (21), we have

$$L_1(x_1, x_2) = \frac{\partial J}{\partial x_1} = a_1(x_1 - z_1) \tag{24}$$

$$L_2(x_1, x_2) = \frac{\partial J}{\partial x_2} = a_2(x_2 - z_2)$$

The vertical boundary in the x-space corresponds to $L_1(x_1, x_2) = 0$ (or $x_1 = z_1$. The horizontal boundary corresponds to $L_2(x_1, x_2) = 0$ (or $x_2 = z_2$).

The choice of the quadratic J function is not unique. (24) leads to a family of quadratic functions. The level set of the J function, $s_\beta = \{x \in \mathbf{R}^2 \mid J(x, \alpha) \leq \beta\}$, is an ellipse centered at (z_1, z_2). For simplicity, we choose $a_1 = a_2 = 1$.

Note that the coefficient of u_1 is not a function of x_2. Therefore, if the buffer is neither empty nor full, (23) indicates that the production flow rate, u_1, is independent of the flow rate u_2, the machine state α_2, and the surplus state x_2. The same observation can be made for u_2. Coupling occurs only when the buffer is either empty or full as specified in the conditional constraints.

In the linear program, there are three unknown parameters we need to determine. They are the components of the hedging point (z_1, z_2) and the buffer size B. By relating starvation and blockage to the system capacity, we determine the smallest buffer size which satisfies performance specification (d) of Section 3.3. The components of the hedging point are determined to meet performance requirement (c).

3.7 Starvation and blockage

If the buffer size is small, Machine 1 will be blocked soon after Machine 2 fails. If the amount of material in the buffer is small, Machine 2 will be starved soon after Machine 1 fails.

Define z^b to be the buffer hedging level (see Fig.5). It is the buffer level when the system reaches the hedging point, which satisfies

$$z^b = z_1 - z_2. \tag{25}$$

Define z^s to be the buffer hedging space. It is the room left for more parts in the buffer when the system reaches the hedging point, which satisfies

$$z^s = B - z^b. \tag{26}$$

Fig.6 illustrates a sample cumulative production trajectory for a system in which $U_1 \geq U_2$, where U_i is the maximum service rate of Machine i (i=1,2). We start with a empty buffer. At the beginning, both machines produce at maximum rates, while Machine 2 starts after Machine 1 finishes the first part. When the system reaches the hedging point, both cumulative production graphs are parallel to the cumulative demand graph. When Machine 1 fails at time t_1, Machine 2 continues to produce and starts to consume the material in the buffer. The buffer becomes empty at time t_2, and Machine 2 is starved until Machine 1 is repaired at time t_3. In this case, the length of the period of starvation $[t_2, t_3]$ is a function of the demand, the buffer level, and the time to repair Machine 1. In general, during the time Machine 1 is down, Machine 2 can fail. Therefore, the amount of time that Machine 2 is starved is affected by the failures of Machine 2.

A similar observation can be made for the blockage of Machine 1. That is, the amount of time that Machine 1 is blocked is a function of demand, the buffer space, the time to repair Machine 2, and the failures of Machine 1.

In the following, we formulate the relationship among the starvation fractions, blockage fractions, demand, and machine parameters. The starvation and blockage

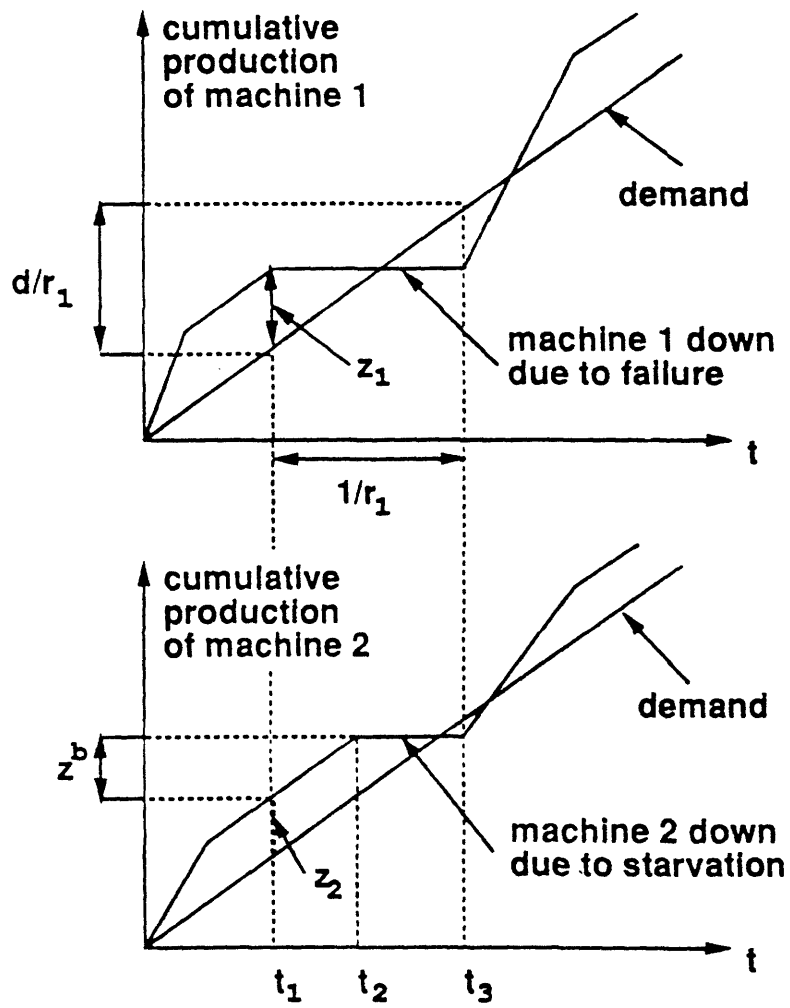

Figure 6: A typical trajectory of the cumulative production

fractions are defined in Section 2.4.

The starvation fraction of Machine 1: Since we assumed that Machine 1 is never starved, the starvation fraction of Machine 1 is

$$f_1^s = 0. \tag{27}$$

The starvation fraction of Machine 2: Assume that the demand is a member of the long term capacity set $\bar{\Omega}$ (12). Then the system has enough capacity to recover from machine breakdowns. That is, when the system leaves the hedging point due to Machine 1 going down, it is very likely that the system will come back to the hedging point after Machine 1 is repaired and before the next failure of Machine 1. As an estimate, we assume that z^b is the amount of material in the buffer at the instant that Machine 1 goes down. Consider the average time interval during which Machine 1 is up once and down once. The length of the average interval is $1/r_1 + 1/p_1$. While Machine 1 is down, Machine 2 can be down, or produce, or be starved (Fig.7).

Let β_1 be the average amount of time that both machines are down during an average Machine 1 up-down period. Note that the Machine i down-time is measured in the complementary frame of the operational time associated with Machine i (i=1,2). The length of the average Machine 1 up-down interval is measured in the working time frame. Therefore, to calculate β_1, we need to convert the machine down times to the frame of working time.

In the working time frame, the fraction of time that Machine i is down is given by

$$\frac{p_i}{r_i + p_i} \qquad (i = 1, 2)$$

The amount of time that both machines are down during $(1/r_1 + 1/p_1)$ is

$$\begin{aligned} \beta_1 &= (\frac{1}{r_1} + \frac{1}{p_1})(\frac{p_1}{r_1 + p_1})(\frac{p_2}{r_2 + p_2}) \\ &= \frac{1}{r_1}(\frac{p_2}{r_2 + p_2}). \end{aligned}$$

Let β_2 be the average amount of time that Machine 2 produces when Machine 1 is down during an average period in which Machine 1 is up once and down once. When Machine 1 is down, the production at Machine 2 is maintained by the material in the buffer. To calculate β_2, we need to know how much material is in the buffer at the

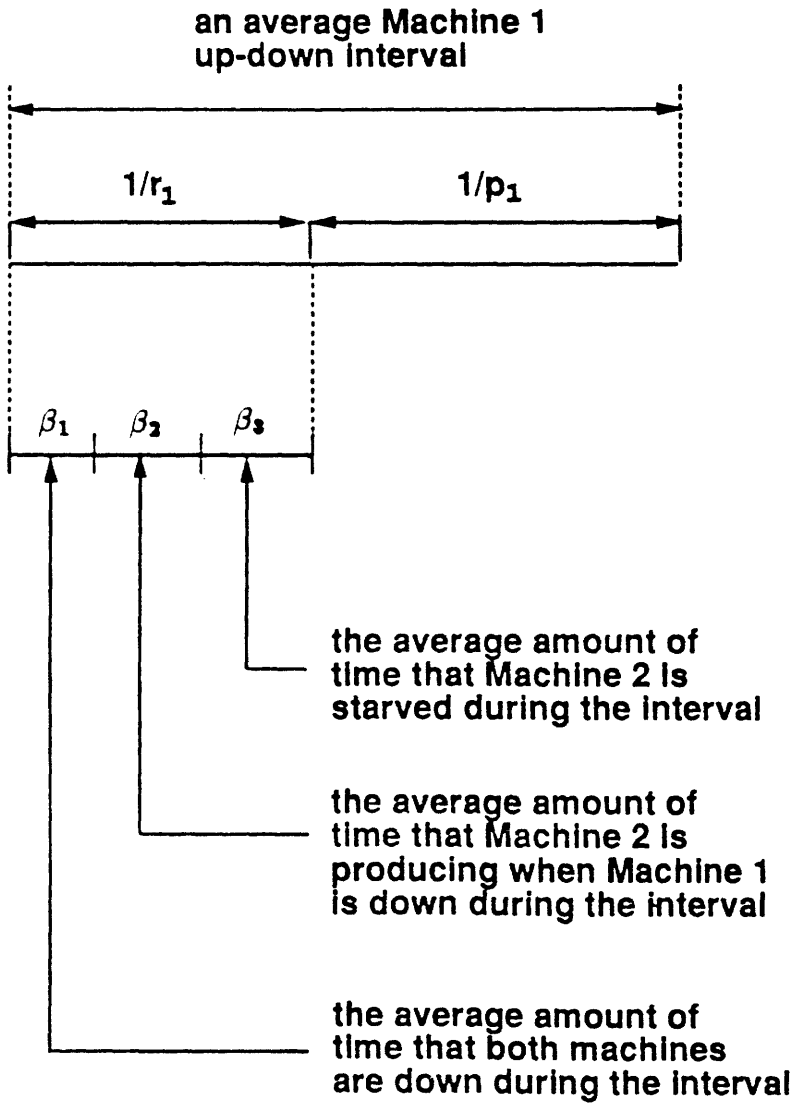

Figure 7: The average cycle time of Machine 1 breakdown

instant that Machine 1 goes down and what the average production rate of Machine 2 is during that time.

Let $\bar{u}_2$ be the average production rate when Machine 2 is producing, which is governed by

$$\frac{1}{p_2}(1 - f_2^s)\bar{u}_2 = (\frac{1}{r_2} + \frac{1}{p_2})d \tag{28}$$

or

$$\bar{u}_2 = \frac{(r_2 + p_2)d}{r_2(1 - f_2^s)}.$$

where $\frac{1}{p_2}(1 - f_2^s)$ is the amount of time that Machine 2 produces during an average Machine 2 up-down interval. Equation (28) says that the cumulative production at Machine 2 equals the cumulative demand during an interval of length $(1/r_2 + 1/p_2)$.

Since we assumed that the average amount of material in the buffer is z^b at the instant that Machine 1 goes down, we have approximately

$$\beta_2 = \frac{z^b}{\bar{u}_2} = \frac{r_2(1 - f_2^s)z^b}{(r_2 + p_2)d}$$

Let β_3 be the average amount of time that Machine 2 is starved when Machine 1 is down during an interval of length $1/r_1 + 1/p_1$. Since the starvation fraction of Machine 2 is defined in the operational time frame, we have to convert it to the frame of working time in order to calculate β_3.

In the working time frame, the fraction of time that Machine 2 is starved is

$$(\frac{r_2}{r_2 + p_2})f_2^s$$

Therefore, the amount of time that Machine 2 is starved during an up-down cycle of Machine 1 is

$$(\frac{1}{r_1} + \frac{1}{p_1})(\frac{r_2}{r_2 + p_2})f_2^s$$

But, since Machine 2 cannot be starved when Machine 1 is up,

$$\beta_3 = (\frac{1}{r_1} + \frac{1}{p_1})(\frac{r_2}{r_2 + p_2})f_2^s$$

The β's satisfy

$$\beta_1 + \beta_2 + \beta_3 = \frac{1}{r_1}. \tag{29}$$

After manipulation, this leads to

$$\frac{1}{d}z^b + \frac{r_1 + p_1}{r_1 p_1}f_2^s - \frac{1}{d}z^b f_2^s = \frac{1}{r_1} \tag{30}$$

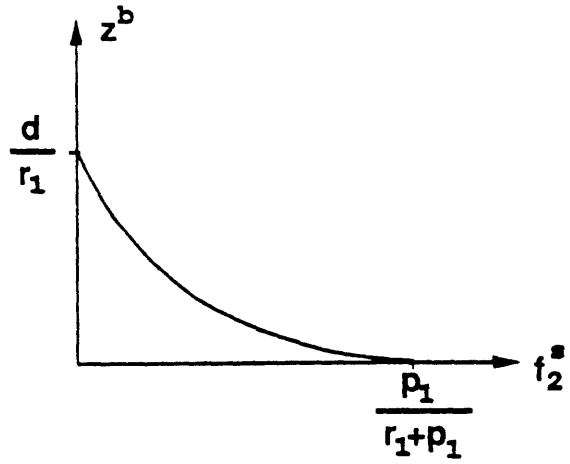

Figure 8: The relationship among z^b, f_2^s, d, r_1, and p_1

or

$$f_2^s = \frac{1}{\frac{1}{r_1} + \frac{1}{p_1} - \frac{z^b}{d}} \left(\frac{1}{r_1} - \frac{z^b}{d}\right) \tag{31}$$

Equation (30) describes the relationship among the buffer hedging level z^b, the starvation fraction f_2^s, the demand, and the machine parameters. In this case, the demand and machine parameters are known. The buffer hedging level and the starvation fraction are decision variables.

Fig.8 depicts the relationship described by Equation (30) in the space of decision variables (f_2^s, z^b), for given demand and machine parameters, r_1 and p_1. The following observations can be made.

Observation 1: Since the buffer hedging level z^b is non-negative, the starvation fraction of Machine 2, f_2^s, is bounded from above by $p_1/(r_1 + p_1)$. This coincides with our intuition. To see this, suppose that Machine 2 is a perfect machine which never fails. Then, the operational time frame of Machine 2 is identical to the frame of working time. Machine 2 is starved whenever Machine 1 fails, which leads to

$$f_2^s = \frac{p_1}{r_1 + p_1}$$

if the buffer hedging level is zero.

When Machine 2 is not perfect, the same result holds since

$$(\frac{r_2}{r_2+p_2})f_2^s \leq (\frac{r_2}{r_2+p_2})(\frac{p_1}{r_1+p_1})$$

where the left-hand-side is the starvation fraction of Machine 2 in the working time frame. The right-hand-side is the time fraction in the working time frame that Machine 1 is down and Machine 2 is up.

Observation 2: Since the starvation fraction of Machine 2 is non-negative, the feasible region of z^b in the equality constraint (30) is bounded from above by d/r_1.

Observation 3: The buffer hedging level z^b is a convex function of the starvation fraction of Machine 2 on its feasible region $f_2^s \in [0, p_1/(r_1+p_1)]$.

The blockage fraction of Machine 1: By similar reasoning, the blockage fraction of Machine 1, f_1^b, satisfies

$$\frac{1}{d}z^s + \frac{r_2+p_2}{r_2p_2}f_1^b - \frac{1}{d}z^s f_1^b = \frac{1}{r_2} \tag{32}$$

or

$$f_1^b = \frac{1}{\frac{1}{r_2}+\frac{1}{p_2}-\frac{z^s}{d}}(\frac{1}{r_2}-\frac{z^s}{d}) \tag{33}$$

Equation (32) describes the relationship among the blockage fraction of Machine 1, the buffer hedging space, the demand, and machine parameters. Fig.9 illustrates the equation. It leads to more observations:

Observation 4: The blockage fraction of Machine 1, f_1^b, is bounded from above by $p_2/(r_2+p_2)$.

Observation 5: Since the blockage fraction of Machine 1 is non-negative, the feasible region of z^s in the equality constraint (32) is bounded from above by z^s/r_2.

Observation 6: The buffer hedging space, z^s, is a convex function of the blockage fraction of Machine 1 on its feasible region, $f_1^b \in [0, p_2/(r_2+p_2)]$.

The blockage fraction of Machine 2: Since we assumed that Machine 2 is never blocked, the blockage fraction of Machine 2 is

$$f_2^b = 0. \tag{34}$$

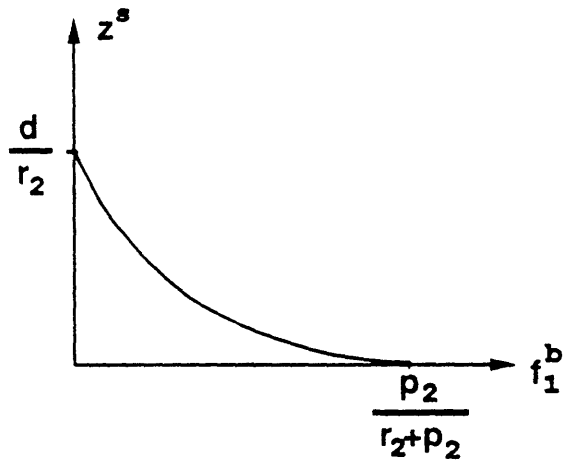

Figure 9: The relationship among z^s, f_1^b, d, r_2, and p_2

When we have excess capacity, we can keep WIP as low as possible by making **starvation** and blockage as large as possible (and meeting demand). We do that by **keeping** buffer size and buffer level small. However, we have to ensure that there is **enough** capacity to maintain production. The starvation and blockage fractions **must therefore** satisfy

$$\frac{\frac{1}{p_1}(1-f_1^b)}{\frac{1}{r_1}+\frac{1}{p_1}}U_1 = \frac{r_1(1-f_1^b)}{r_1+p_1}U_1 \geq d, \tag{35}$$

$$\frac{\frac{1}{p_2}(1-f_2^s)}{\frac{1}{r_2}+\frac{1}{p_2}}U_2 = \frac{r_2(1-f_2^s)}{r_2+p_2}U_2 \geq d, \tag{36}$$

where U_i is the maximum service rate of Machine i. In this case, $U_i = 1/r_i$ (i=1,2). We assume that the demand d is a member of the constraint set (12). **Conditions** (35) and (36) ensure that both machines have enough capacity to achieve the **demand** given the blockage and starvation fractions. Let d_{max_i} be the isolated capacity of Machine i which is given by

$$d_{max_i} = \frac{r_i}{r_i+p_i}U_i \quad (i=1,2).$$

After a rearrangement, the starvation and blockage constraints, (35) and (36), become

$$f_1^b \leq 1 - \frac{d}{d_{max1}}, \tag{37}$$

$$f_2^s \leq 1 - \frac{d}{d_{max2}}. \tag{38}$$

Note that since $f_1^b, f_2^s \in [0,1]$, feasible demand must satisfy

$$0 \leq d \leq \min\{d_{max1}, d_{max2}\}.$$

3.8 The nonlinear program for buffer hedging level and hedging space

In the previous section, we formulated the relations among the starvation and blockage fractions, the buffer hedging level and space, the demand, and the machine parameters. In this section, we establish a nonlinear programming problem to determine the optimal buffer size and average buffer level.

As we discussed earlier, one of the objectives is to minimize the WIP inventory, which is equivalent to minimizing the average buffer level and buffer size. That can be formulated as an optimization problem, by putting (30), (32), (37), and (38) together as follows

$$\min\{z^b + z^s\} \tag{39}$$

subject to:

$$\frac{1}{d}z^b + \frac{r_1 + p_1}{r_1 p_1} f_2^s - \frac{1}{d} z^b f_2^s = \frac{1}{r_1}$$

$$\frac{1}{d}z^s + \frac{r_2 + p_2}{r_2 p_2} f_1^b - \frac{1}{d} z^s f_1^b = \frac{1}{r_2}$$

$$f_1^b \leq 1 - \frac{d}{d_{max1}}$$

$$f_2^s \leq 1 - \frac{d}{d_{max2}}$$

$$f_1^b \geq 0, \quad f_2^s \geq 0$$

$$z^b \geq 0, \quad z^s \geq 0$$

Solving (39) is equivalent to minimizing both the buffer size and average buffer level, since the buffer size is defined in Section 2.2.2 as

$$B = z^b + z^s$$

The buffer hedging level is different from the average buffer level. Let $\bar{b}$ be the average buffer level, which can be obtained by taking time average of (16)

$$\bar{b} = \bar{x}_1 - \bar{x}_2 \tag{40}$$

The relation between the hedging buffer level and the average buffer level is given by

$$\bar{b} = z^b + (\Delta_2 - \Delta_1)$$

where $\Delta_i (i = 1, 2)$ are the average surplus losses, which are discussed in next section.

3.9 The hedging point and surplus loss

In the previous section, we determined the optimal buffer size needed in the feedback controller (21). In this section, we are going to determine the hedging point according to the system performance requirement (c) specified in Section 3.3. Since the system performance is based on the average surplus of the final operation, we have to formulate the relations among the components of the hedging point and average surpluses.

Since both machines are unreliable and can be starved or blocked, there is a difference between the hedging point (z_1, z_2) and the average surplus $(\bar{x}_1, \bar{x}_2)$, which is the time average over the planning horizon of (x_1, x_2). The relation can be written

$$z_i = \bar{x}_i + \Delta_i \quad (i = 1, 2) \tag{41}$$

where $\Delta_i (i = 1, 2)$ is the average surplus loss at Machine i, which is the average amount that x_i deviates from z_i.

The average surplus loss $\Delta_i (i = 1, 2)$ consists of three components caused by failure, starvation, and blockage. For simplicity, we assume that the three components are independent of each other. That is, the three components can be calculated separately. Note that the assumption is a heuristic approximation.

Fig.10 illustrates the typical surplus loss due to failures. The shaded area is the total surplus loss due to failure during an average Machine i up-down interval. The area of the shaded region is equal to

$$\frac{1}{2}(\frac{1}{r_i})^2 d + t_{ci}(\frac{d}{r_i}) + \frac{1}{2} t_{ci}^2 d - \frac{1}{2} t_{ci}^2 U_i$$

where t_{ci} is the average catch-up time needed for Machine i to recover from failures. It is given by

$$t_{ci} = \frac{d}{r_i(U_i - d)}$$

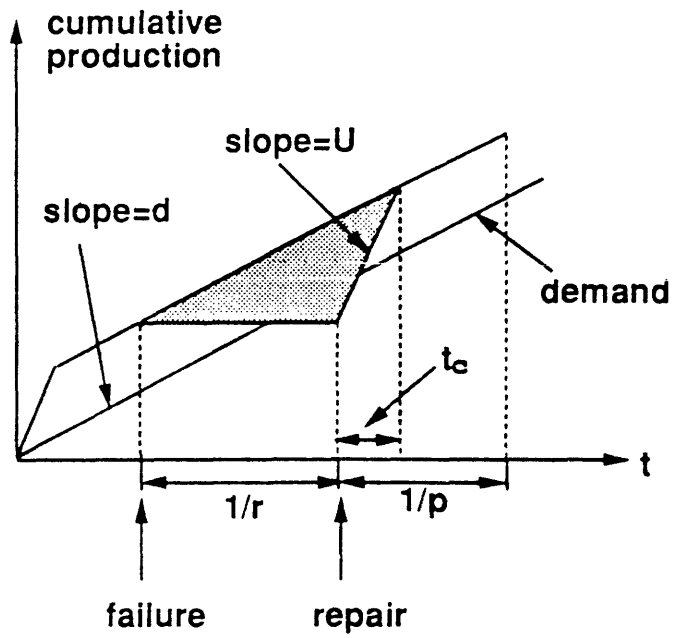

Figure 10: The surplus loss due to failure

Also, U_i is the maximum service rate of Machine i and is $1/\tau_i$.

Let δ_i^r be the average surplus loss due to failures for Machine i. Dividing the shaded area of Fig.10 by the average time that Machine i is up once and down once, $(1/r_i + 1/p_i)$, leads to

$$\begin{aligned}\delta_i^r &= \frac{\frac{1}{2}(\frac{1}{r_i})^2 d + t_{ci}(\frac{d}{r_i}) + \frac{1}{2}t_{ci}^2 d - \frac{1}{2}t_{ci}^2 U_i}{\frac{1}{r_i} + \frac{1}{p_i}} \\ &= \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{1}{r_i})^2 \quad (i = 1, 2) \end{aligned} \tag{42}$$

Let δ_i^s be the average surplus loss due to starvation. As we did for δ_i^r, we determine δ_i^s by dividing the total surplus loss due to starvation in an average Machine i up-down interval by the length of the interval, $1/r_i + 1/p_i$. To do that, we use a heuristic approximation by replacing the machine down time $1/r_i$ with the starvation time in the interval, f_i^s/p_i, in (42). Then

$$\delta_i^s = \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{f_i^s}{p_i})^2 \quad (i = 1, 2) \tag{43}$$

Similarly, δ_i^b, the average surplus loss due to blockage, is given by

$$\delta_i^b = \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{f_i^b}{p_i})^2 \quad (i = 1, 2) \tag{44}$$

Therefore the average surplus loss is approximately given by

$$\begin{aligned}\Delta_i &= \delta_i^r + \delta_i^s + \delta_i^b \\ &= \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})\{(\frac{1}{r_i})^2 + (\frac{f_i^s}{p_i})^2 + (\frac{f_i^b}{p_i})^2\} \quad (i = 1, 2). \end{aligned} \tag{45}$$

According to the performance specification (c) of Section 3.3, we would like to minimize the absolute value of surplus x_2. Since we do not have control over the variance of x_2, the best we can do is to keep the average of x_2 close to zero. Therefore, we choose the hedging point (z_1, z_2) such that

$$\bar{x}_2 = 0. \tag{46}$$

From (25), (41), and (46), the hedging point should satisfy

$$\begin{aligned} z_2 &= \Delta_2; \\ z_1 &= z^b + \Delta_2. \end{aligned} \tag{47}$$

Up to this point, we have constructed the real-time scheduler for the two-machine, one-part-type system. In Section (3.6), the feedback controller is established as a linear programming problem with three unknown parameters, namely, the buffer size, B, and the components of the hedging point, z_1 and z_2. In Section (3.8), the optimal buffer size is determined by solving a non-linear optimization problem. The components of the hedging point are determined above in this section.

3.10 The algorithm and the hierarchical policy

In this section, we summarize the steps of the algorithm of the production scheduler and describe the hierarchical structure for implementation.

Step 1: Collect the input data set, which consists of the failure rate p_i, the repair rate r_i, and the processing time, τ_i for Machine i (i=1,2), and the demand.

Step 2: Calculate the buffer hedging level z^b and hedging space z^s, and the starvation and blockage fractions for each machine by solving the the nonlinear program (39). Then, calculate the buffer size, B, by summing the buffer hedging level and hedging space.

Step 3: calculate the components, z_1 and z_2, of the hedging point according to (47).

Step 4: Using the feedback information of surplus x_i and machine state α_i (i=1,2), calculate the production rates, u_i (i=1,2), in real time by solving the linear program (23).

Step 5: The loading times for each machine are determined by a heuristic policy called staircase strategy [23]. That is, whenever the actual cumulative production is less than the integral of the production rate, load a part into the machine.

Step 6: If the demand or any one of the machine parameters changes, go to Step 2.

This production scheduling algorithm can be divided into a three-level hierarchy [18]. At the top level of the hierarchy, we calculate the buffer size and the hedging point given the demand and the machine parameters (as we state in Step 1, 2, and 3 of the

algorithm). At the middle level, we calculate the production rates in real time, while the machine states and the production surplus are fedback from the shop-floor (Step 4 of the algorithm). At the bottom level of the hierarchy, we determine the loading times for each machine using the staircase strategy (Step 5 of the algorithm). Fig.11 illustrates the hierarchical structure.

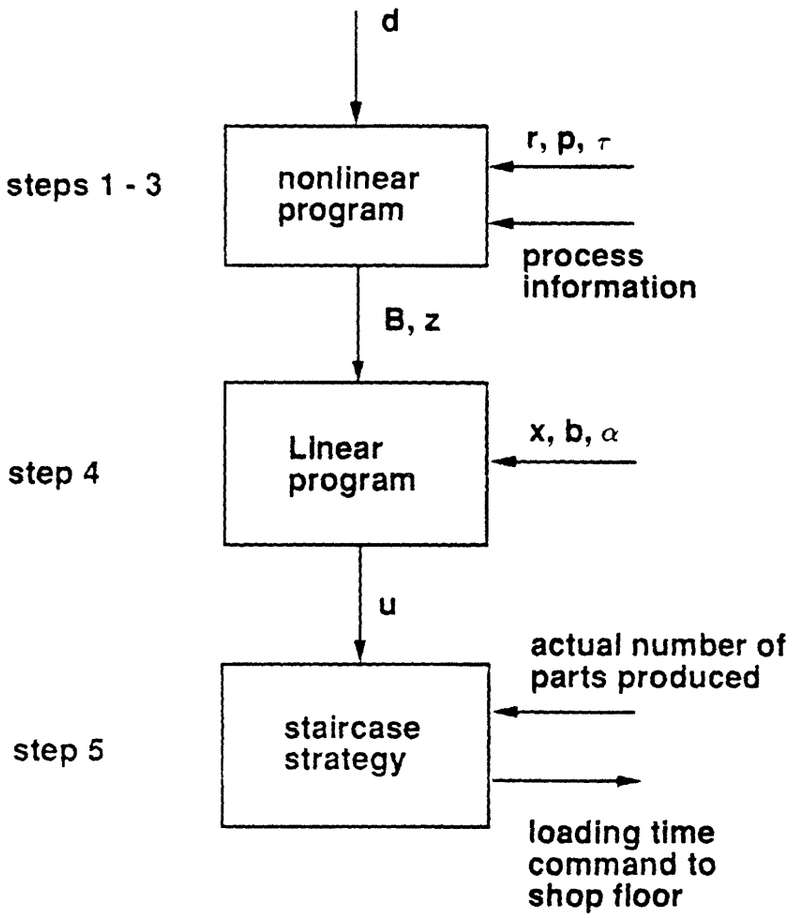

Figure 11: The hierarchical policy

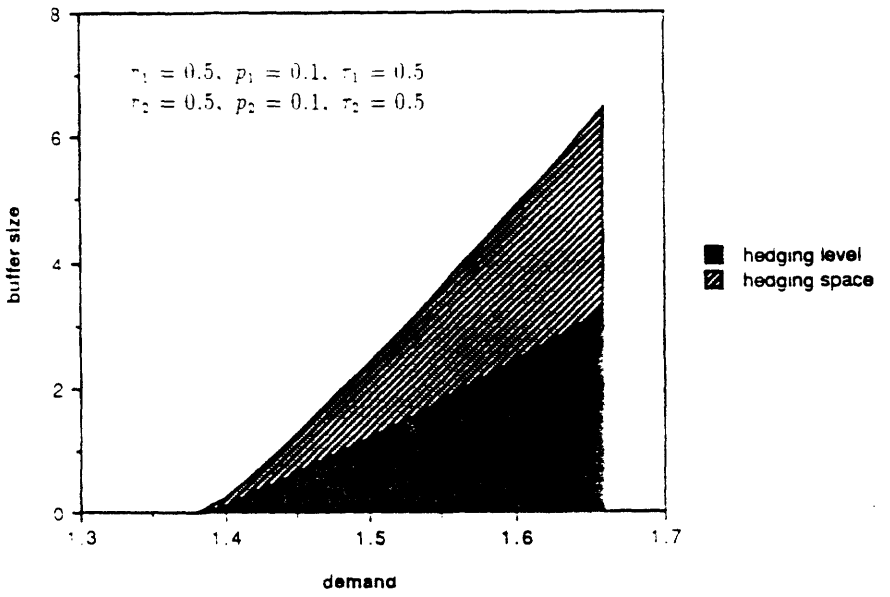

Figure 12: Buffer size vs demand

3.11 Example

To verify the algorithm, we have done simulations for many different cases. All the simulations showed us very promising results. In this section, we demonstrate a simulation example of the production control algorithm. First, we look at the control parameter calculation at the top level of the hierarchy. Then we simulate a production system with a constant demand.

3.11.1 Buffer size vs demand and machine parameters

The nonlinear program (39) is a well behaved problem. For the top level calculation in the algorithm, we used a commercially available software package [35].

As we mentioned earlier, the buffer level and size are functions of the demand and machine parameters. Fig.12 depicts the results of the top level calculation for different demand values. When the demand is small, the system has extra capacity which can be used to reduce the buffer level and size. The bigger the demand is, the bigger the buffer level and size are.

Fig.13 illustrates the results of the top level calculation for different values of r_1. The bigger r_1 is, the more reliable Machine 1 is, and so the smaller the buffer level and size are.

Fig.14 illustrates the results of the top level calculation for different values of p_1. The bigger p_1 is, the smaller the isolated efficiency of Machine 1 is, and the bigger

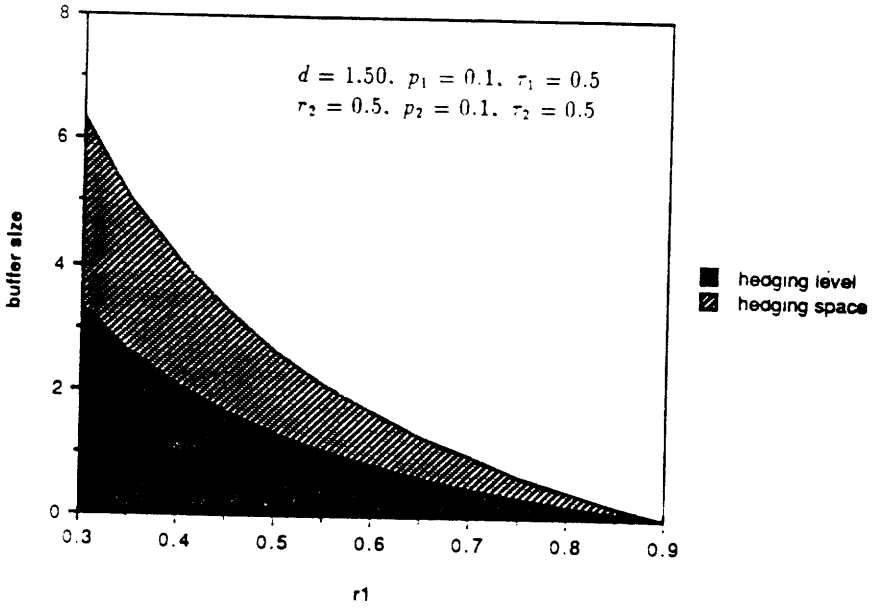

Figure 13: Buffer size vs r_1

the buffer level and size are.

3.11.2 Simulation results

For the simulation, we chose the machine parameters as follows:

$$r_1 = 0.5, \quad p_1 = 0.1, \quad \tau_1 = 0.5$$
$$r_2 = 0.5, \quad p_2 = 0.1, \quad \tau_2 = 0.5$$

where the unit of r, p, and τ is 1/day. The unit of parts is lot.

Given that the demand is equal to 1.6 (lots/day), the buffer size and hedging point are listed in Table.1. In the simulations, the buffer size is rounded up to an integer.

The simulation program that we use is called HIERCSIM which was developed by B. Darakananda [36]. Fig.15 illustrates the cumulative production results of the simulation. The straight line is the cumulative demand. The upper curve is the cumulative input of the raw parts at Machine 1. The lower curve is the cumulative output of the final products at Machine 2. The dashed lines are the middle level results which are the integrals of the flow rates. The staircase-like graphs are the bottom level results which are the actual count of cumulative production. It is almost impossible to tell the difference between the middle and bottom level results.

Fig.16 illustrates buffer level vs time. The buffer level consists of the parts in the buffer and the working piece in Machine 1. The dashed line is the middle level result.

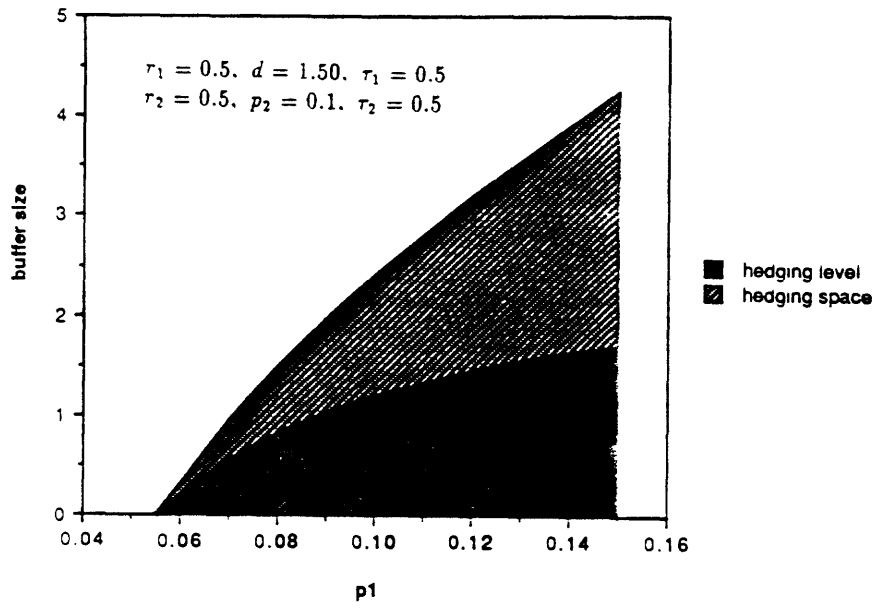

Figure 14: Buffer size vs p_1

i	f_i^s	f_i^b	z_i^s	z_i^b	B_i	z_i
1	0.0	0.04	2.53	2.53	5	3.9
2	0.04	0.0				1.4

Table 1: The buffer size and hedging point for the two machine and one part type example

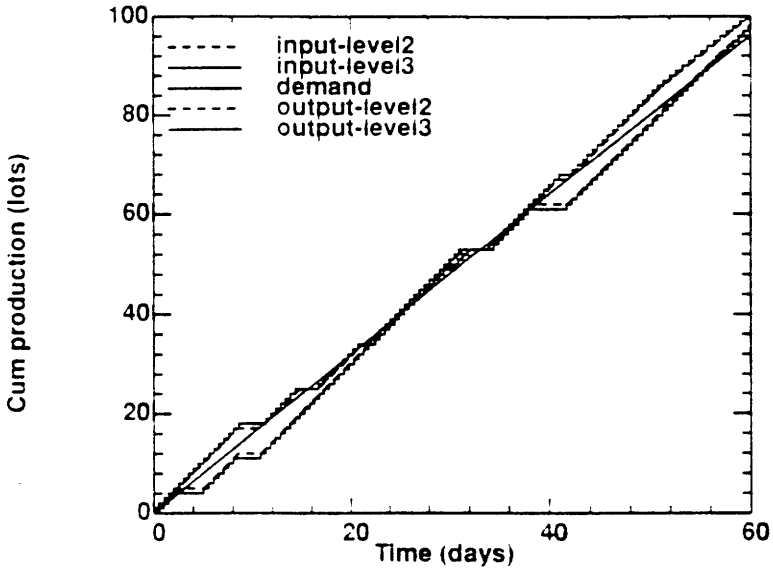

Figure 15: The simulation result of the cumulative production of a two-machine and one-part-type system

That is, it is the buffer level $b(t)$ which is governed by Eq.(5). The solid line is the bottom level result which is the actual count of the parts in the buffer. The actual count and $b(t)$ rarely differ by more than 1.

3.12 Summary

In this section, a real-time feedback control algorithm has been developed for the scheduling of two-machine, one-part-type system. The simulation results verify that it works well.

In the following section, we extend the algorithm to N-machine, one-part-type systems.

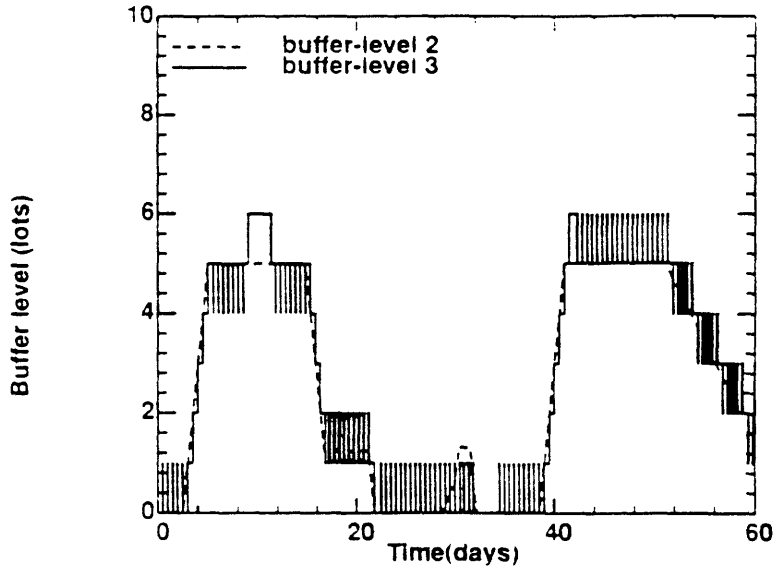

Figure 16: The simulation result of buffer level

4 N-machine, one-part-type system without reentry

In this section, we study the N-machine, one-part-type system. As illustrated in Fig.17, the system consists of N machines and $N-1$ buffers. For Machine i (i=1,2,...,N), the failure rate is p_i and the repair rate is r_i. The mean time to fail $MTTF_i = 1/p_i$ and the mean time to repair $MTTR_i = 1/r_i$. One part type is produced. Each part needs an operation with processing time τ_i on Machine i. The parts travel in a fixed sequence: Machine 1, Machine 2, ..., Machine N. The buffers are located between machines. We assume that Machine 1 is never starved and Machine N is never blocked.

In this case, a machine in the middle of the production line can be either starved or blocked. The relations among machines are more complex than the previous case, since a machine failure can starve or block more than one machine. The technique developed in the previous section is extended to deal with the serial production line.

4.1 Dynamic optimization

Given the system as described above, an initial surplus state vector $x(t_0)$, and machine state vector $\alpha(t_0)$, we wish to specify a feedback control strategy $u(x, \alpha)$ for production

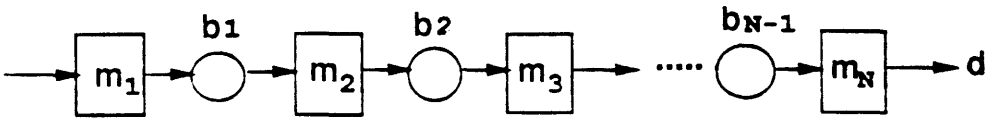

Figure 17: N-machine, one-part-type system

during $t_0 \leq t \leq T$ that satisfies

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \tag{48}$$

subject to:

$$\begin{array}{ll} \tau_i u_i \leq \alpha_i & (i = 1, 2, \ldots, N) \\ u_i \geq 0 & (i = 1, 2, \ldots, N) \end{array}$$

where the system dynamics are

$$\begin{array}{ll} \dot{x}_i = u_i - d & (i = 1, 2, \ldots, N) \\ \dot{b}_i = u_i - u_{i+1} & (i = 1, 2, \ldots, N-1) \\ B_i \geq b_i \geq 0 & (i = 1, 2, \ldots, N-1) \end{array}$$

where B_i is the buffer size which is to be determined. Assume that the initial buffer level $b_i(t_0)$ satisfies

$$b_i(t_0) = x_i(t_0) - x_{i+1}(t_0), \qquad (i = 1, 2, \ldots, N-1)$$

Then by the definition (5) and (13), we have

$$b_i = x_i - x_{i+1} \qquad (i = 1, 2, \ldots, N-1). \tag{49}$$

The constraints are in form of (6) and (8). The function $g(x, b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive or too negative.

4.2 Feedback control law and the quadratic J function

With the same reasoning as for the two-machine, one-part-type system in Section 3, the optimal production rate, u, can be determined, if the optimal cost-to-go (or value

function) $J(x,\alpha,t)$ is known, by solving the following linear programming problem for u:

$$\min_{u}\{\frac{\partial J}{\partial x_1}u_1+\frac{\partial J}{\partial x_2}u_2+\ldots+\frac{\partial J}{\partial x_N}u_N\} \tag{50}$$

subject to:

$$\begin{array}{ll}\tau_i u_i\leq\alpha_i & (i=1,2,\ldots,N)\\ u_i\geq 0 & (i=1,2,\ldots,N)\end{array}$$

where

$$\begin{array}{ll}\dot{x}_i=u_i-d & (i=1,2,\ldots,N)\\ \dot{b}_i=u_i-u_{i+1} & (i=1,2,\ldots,N-1)\\ B_i\geq b_i\geq 0 & (i=1,2,\ldots,N-1)\end{array}$$

Let

$$\tilde{x}=\{x_1,\ldots,x_N,b_1,\ldots,b_{N-1}\}^t$$

$$=\{x_1,\ldots,x_N,x_1-x_2,\ldots,x_{N-1}-x_N\}^t$$

Since we do not know the optimal cost-to-go $J(x,\alpha,t)$, a quadratic approximation

$$J=\frac{1}{2}\tilde{x}^t A\tilde{x}+c^t\tilde{x}+D \tag{51}$$

is used, where A is a (2N-1)×(2N-1) positive semi-definite and symmetric matrix, C is a (2N-1)×1 vector, and D is a scalar.

Let

$$L_1(x)=\frac{\partial J}{\partial x_1}=(1\ \ 0\ldots 0\ \ 1\ \ 0\ldots 0)A\begin{pmatrix}x_1\\x_2\\\vdots\\x_N\\x_1-x_2\\\vdots\\x_{N-1}-x_N\end{pmatrix}+C^t\begin{pmatrix}1\\0\\\vdots\\0\\1\\0\\\vdots\\0\end{pmatrix}$$

$$L_2(x)=\frac{\partial J}{\partial x_2}=(0\ \ 1\ldots 0\ \ 0\ \ -1\ \ \ldots 0)A\begin{pmatrix}x_1\\x_2\\\vdots\\x_N\\x_1-x_2\\\vdots\\x_{N-1}-x_N\end{pmatrix}+C^t\begin{pmatrix}0\\1\\\vdots\\0\\0\\-1\\\vdots\\0\end{pmatrix}$$

$$\vdots$$

$$L_N(x) = \frac{\partial J}{\partial x_N} = (0 \;\; 0 \ldots 1 \;\; 0 \;\; 0 \;\; \ldots -1)A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \\ x_1 - x_2 \\ \vdots \\ x_{N-1} - x_N \end{pmatrix} + C^t \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix} \qquad (52)$$

Note that $L_i(x)$ is a linear function of $x = \{x_1, \ldots, x_N\}^t$, (i=1, ..., N).

Plugging (52) into (50), the linear programming problem becomes

$$\min_u \{\sum_{i=1}^{N} L_i(x_1, \ldots, x_N) u_i\} \qquad (53)$$

subject to:

$$\begin{aligned} \tau_i u_i \leq \alpha_i \quad & (i = 1, 2, \ldots, N) \\ u_i \geq 0 \quad & (i = 1, 2, \ldots, N) \end{aligned}$$

where

$$\begin{aligned} \dot{x}_i &= u_i - d \quad && (i = 1, 2, \ldots, N) \\ \dot{b}_i &= u_i - u_{i+1} \quad && (i = 1, 2, \ldots, N-1) \\ B_i \geq b_i &\geq 0 \quad && (i = 1, 2, \ldots, N-1) \end{aligned}$$

As we observed in Section 3.2, the coefficient boundaries divide the x-space into mutually exclusive regions. Because the coefficients are linear functions of x, the boundaries are straight line segments.

4.3 System behavior and performance specification

To solve the production control problem, we need to find the parameters of the quadratic J function. We specify the desirable system behavior and performance. If the linear program (53) gives us satisfactory system behavior and performance, we say that it is an approximation of the dynamic optimization problem (48), and the quadratic J function is an approximation of the optimal value function. The following are specifications on system behavior and performance:

(1) When Machine 1 fails, keep Machine 2 producing without changing the production plan until Buffer 1 is empty.

(2) When Machine i (i=2,...,N-1) fails, keep Machine i-1 producing without changing the production plan until Buffer i-1 is full and keep Machine i+1 producing without

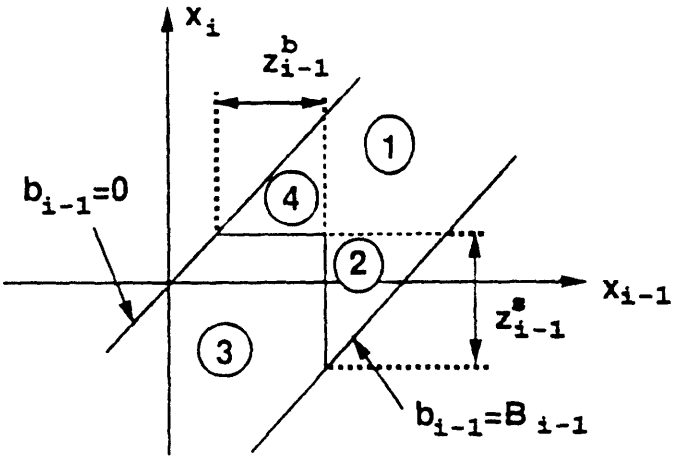

Figure 18: The desirable boundary shape in (x_{i-1},x_i) space

changing the production plan until the Buffer i is empty.
(3) When Machine n fails, keep Machine N-1 producing without changing the production plan until Buffer N-1 is full.
(4) Keep the absolute value of x_N as small as possible. That is, keep the production close to demand.
(5) Keep the buffer sizes and average buffer levels as small as possible. That is, keep WIP low and throughput time short.

The behavior requirements (1), (2), and (3) demand that all coefficient boundaries in x-space are parallel with axes. The performance requirements (4) and (5) will be used to determine the buffer sizes and the hedging point.

4.4 The conditional constraints

The solutions on the boundaries are different from those in the interior of x-space. The properties are modeled as conditional constraints to the linear program.

Define $(z_1, z_2, \ldots, z_N)$ to be the hedging point which is the desirable operating state of the system. The conditional constraints are

$$\begin{array}{llll} \text{if } x_i = z_i, & \text{then} & u_i = d & (i = 1,2,\ldots,N) \\ \text{if } b_i = 0, & \text{then} & u_i \geq u_{i+1} & (i = 1,2,\ldots,N-1) \\ \text{if } b_i = B_i, & \text{then} & u_i \leq u_{i+1} & (i = 1,2,\ldots,N-1) \end{array} \qquad (54)$$

which imply that when the production surplus x_i reaches its component of the hedging point, z_i, at Machine i, the flow rate should be equal to the demand. This is so that chattering on the attractive boundary can never occur [20]. When Buffer i is empty, Machine i+1 cannot be faster than the upstream machine. When Buffer i is full, Machine i cannot be faster than the downstream machine.

4.5 The linear program

To ensure that the coefficient boundaries in x-space are parallel with axes, and go through the hedging point, the linear program (53) becomes

$$\min_u \{a_1(x_1 - z_1)u_1 - a_2(x_2 - z_2)u_2 + \ldots + a_N(x_N - z_N)u_N\} \qquad (55)$$

subject to:

$$\begin{array}{lll} r_i u_i \leq \alpha_i & & (i = 1, 2, \ldots, N) \\ u_i \geq 0 & & (i = 1, 2, \ldots, N) \\ \text{if } x_i = z_i, & \text{then } u_i = d & (i = 1, 2, \ldots, N) \\ \text{if } b_i = 0, & \text{then } u_i \geq u_{i+1} & (i = 1, 2, \ldots, N-1) \\ \text{if } b_i = B_i, & \text{then } u_i \leq u_{i+1} & (i = 1, 2, \ldots, N-1) \end{array}$$

where

$$\begin{array}{ll} \dot{x}_i = u_i - d & (i = 1, 2, \ldots, N) \\ \dot{b}_i = u_i - u_{i+1} & (i = 1, 2, \ldots, N-1) \\ B_i \geq b_i \geq 0 & (i = 1, 2, \ldots, N-1) \end{array}$$

In the linear program, the hedging point $(z_1, \ldots, z_N)$ and the buffer sizes $(B_1, \ldots, B_{N-1})$ are still unknown. We show how they may be found in Section 4.7 and 4.8.

Comparing (55) and (53), we have

$$L_i(x_1, \ldots, x_N) = \frac{\partial J}{\partial x_i} = a_i(x_i - z_i) \qquad (i = 1, 2, \ldots, N) \qquad (56)$$

The choice of the quadratic J function is not unique. (56) leads to a family of convex functions, whose level sets are ellipsoids centered at the hedging point. For simplicity, we choose $a_i = 1$, (i=1, 2, ..., N).

4.6 Starvation and blockage

Define z_i^b to be the hedging level of Buffer i (see Fig.18). It is the number of parts in Buffer i when the system reaches the hedging point, which satisfies

$$z_i^b = z_i - z_{i+1} \quad (i = 1, 2, \ldots, N-1). \qquad (57)$$

Define z_i^s to be the hedging space of Buffer i. It is the room left for more parts in Buffer i when the system reaches the hedging point, which satisfies

$$z_i^s = B_i - z_i^b \quad (i = 1, 2, \ldots, N-1). \tag{58}$$

The starvation fraction of Machine 1: Since we assumed that Machine 1 is never starved, we have

$$f_1^s = 0. \tag{59}$$

The starvation fraction of Machine i (i=2,....N): We assume that the demand d is a member of the long term capacity set (12). Then the system has enough capacity to recover from machine failures. That is, when the system leaves the hedging point due to Machine i-1 going down, it is very likely that the system will come back to the hedging point after Machine i-1 is repaired.

As we did in Section 3.7, we assume that the amount of material in Buffer i-1 is z_{i-1}^b at the instant that Machine i-1 goes down. Consider the average length of a period in which Machine i-1 is up once and down once, $1/r_{i-1} + 1/p_{i-1}$. During such a period, Machine i-1 is starved for amount of time f_{i-1}^s/p_{i-1} (since that a machine cannot be starved when it is down). The average amount of time that Machine i-1 is down or starved during this period is $(1/r_{i-1} + f_{i-1}^s/p_{i-1})$. When Machine i-1 is down or starved, Machine i can be down, or blocked, or starved, or producing (see Fig. 19).

Let β_i be the average amount of time that Machine i is down or blocked when Machine i-1 is down or starved during a Machine i-1 up-down cycle. In the working time frame, the fraction of time that Machine i-1 is down or starved is

$$\frac{\frac{1}{r_{i-1}} + \frac{f_{i-1}^s}{p_{i-1}}}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}}$$

The fraction of time that Machine i is down or blocked is

$$\frac{\frac{1}{r_i} + \frac{f_i^b}{p_i}}{\frac{1}{r_i} + \frac{1}{p_i}}$$

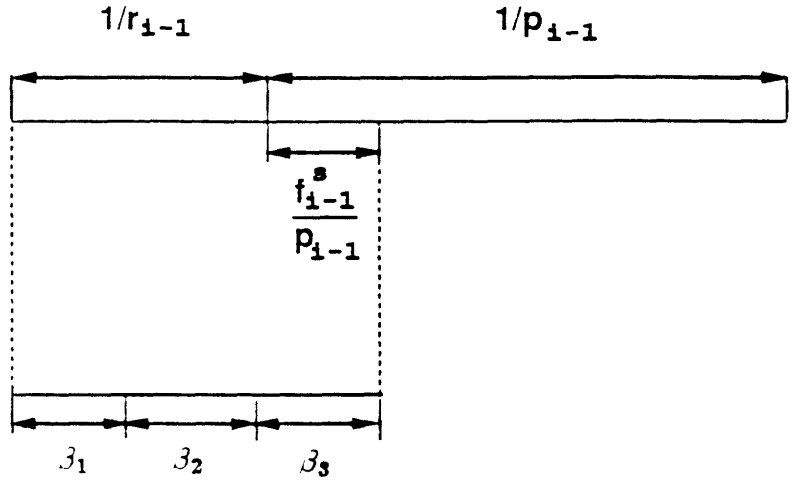

Figure 19: The average cycle time of Machine i-1 breakdown

Consequently,

$$\mathcal{B}_1 = (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}) \left(\frac{\frac{1}{r_{i-1}} + \frac{f^s_{i-1}}{p_{i-1}}}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}} \right) \left(\frac{\frac{1}{r_i} + \frac{f^b_i}{p_i}}{\frac{1}{r_i} + \frac{1}{p_i}} \right)$$

$$= (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} f^s_{i-1})(\frac{r_i p_i}{r_i + p_i})(\frac{1}{r_i} + \frac{1}{p_i} f^b_i).$$

Let $\bar{u}_i$ be the average production flow rate of Machine i when it is producing. It is determined by

$$\frac{1}{p_i}(1 - f^s_i - f^b_i)\bar{u}_i = (\frac{1}{r_i} + \frac{1}{p_i})d \qquad (60)$$

or

$$\bar{u}_i = \frac{(r_i + p_i)d}{r_i(1 - f^s_i - f^b_i)}$$

where $(1/p_i)(1 - f^s_i - f^b_i)$ is the amount of time that Machine i is producing during an average Machine i up-down interval. Equation (60) says that the cumulative production at Machine i equals the cumulative demand during an interval of length $(1/r_i + 1/p_i)$. Let $\mathcal{B}_2$ be the average amount of time that Machine i produces when Machine i-1 is down or starved during an interval of length $1/r_{i-1} + 1/p_{i-1}$. When Machine i-1 is down or starved, the production at Machine i is maintained by the material in Buffer i-1. Since we assumed that the amount of material in Buffer i-1 is z^b_{i-1} at the instant that Machine i-1 goes down, the average amount of time that

production at Machine i can last, approximately, is

$$\mathcal{B}_2 = \frac{z_{i-1}^b}{\bar{u}_i}.$$

Let $\mathcal{B}_3$ be the average amount of time that Machine i is starved when Machine i-1 is down or starved during an interval of length $1/r_{i-1} + f_{i-1}^s/p_{i-1}$.

In the working time frame, the fraction of time that Machine i is starved is

$$\frac{\frac{f_i^s}{p_i}}{\frac{1}{r_i}+\frac{1}{p_i}}$$

or

$$(\frac{r_i p_i}{r_i + p_i})\frac{1}{p_i} f_i^s$$

The amount of time that Machine i is starved during an Machine i-1 up-down interval is

$$(\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i p_i}{r_i + p_i})\frac{1}{p_i} f_i^s. \tag{61}$$

Since Machine i cannot be starved when Machine i-1 is up, $\mathcal{B}_3$ is the same as (61).

$$\begin{aligned}\mathcal{B}_3 &= (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i p_i}{r_i + p_i})\frac{1}{p_i} f_i^s \\ &= f_i^s(\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i}{r_i + p_i}).\end{aligned}$$

The $\mathcal{B}$'s satisfy

$$\mathcal{B}_1 + \mathcal{B}_2 + \mathcal{B}_3 = \frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} f_{i-1}^s. \tag{62}$$

Plugging the $\mathcal{B}$'s into (62), and manipulating, leads to

$$\frac{1}{d} z_i^b - \frac{1}{p_i} f_i^s + \frac{r_i + p_i}{r_i p_i} f_{i+1}^s + \frac{1}{r_i} f_{i+1}^b - \frac{1}{d} z_i^b f_{i+1}^s - \frac{1}{d} z_i^b f_{i+1}^b + \frac{1}{p_i} f_i^s f_{i+1}^b = \frac{1}{r_i} \tag{63}$$

or

$$f_i^s = \frac{1 - f_i^b}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} - \frac{z_{i-1}^b}{d}}(\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} f_{i-1}^s - \frac{z_{i-1}^b}{d}) \quad (i = 2, \ldots, N). \tag{64}$$

The blockage fraction of machine i (i=1,2,....N-1): Assume that the average spare space in Buffer i is z_i^s at the instant that Machine i−1 goes down. By similar reasoning as for f_i^s, the blockage fraction of Machine i is governed by

$$\frac{1}{d} z_i^s + \frac{1}{r_{i+1}} f_i^s + \frac{r_{i+1} + p_{i+1}}{r_{i+1} p_{i+1}} f_i^b - \frac{1}{p_{i+1}} f_{i+1}^b - \frac{1}{d} z_i^s f_i^s - \frac{1}{d} z_i^s f_i^b + \frac{1}{p_{i+1}} f_i^s f_{i+1}^b = \frac{1}{r_{i+1}} \tag{65}$$

or

$$f_i^b = \frac{1 - f_i^s}{\frac{1}{r_{i+1}} + \frac{1}{p_{i+1}} - \frac{z_i^s}{d}}\left(\frac{1}{r_{i+1}} + \frac{1}{p_{i+1}} f_{i+1}^b - \frac{z_i^s}{d}\right), \quad (i = 1, 2, \ldots, N-1). \tag{66}$$

The blockage fraction of Machine N: Since we assumed that Machine N is **never** blocked, the blockage fraction of Machine N is

$$f_N^b = 0. \tag{67}$$

To ensure that the system has enough capacity to achieve the demand, the **starvation** and blockage fractions must satisfy

$$f_i^b + f_i^s \leq 1 - \frac{d}{d_{maxi}}, \quad (i = 1, 2, \ldots, N) \tag{68}$$

where

$$d_{maxi} = \frac{r_i}{r_i + p_i} U_i \quad (i = 1, 2, \ldots, N).$$

and U_i is the maximum service rate of Machine i. In this case, $U_i = 1/\tau_i$ (i=1,2,...,**N**).

4.7 Buffer hedging levels and buffer sizes

By putting (59), (63), (65), (67), and (68) together, we form an optimization **problem** to minimize the buffer hedging levels and spaces.

$$\min\{z_1^b + \ldots + z_{N-1}^b + z_1^s + \ldots + z_{N-1}^s\} \tag{69}$$

subject to:

$$\frac{1}{d} z_i^b - \frac{1}{p_i} f_i^s + \frac{r_i + p_i}{r_i p_i} f_{i+1}^s + \frac{1}{r_i} f_{i+1}^b - \frac{1}{d} z_i^b f_{i+1}^s - \frac{1}{d} z_i^b f_{i+1}^b + \frac{1}{p_i} f_i^s f_{i+1}^b = \frac{1}{r_i}$$
$$(i = 1, 2, \ldots, N-1)$$

$$\frac{1}{d} z_i^s + \frac{1}{r_{i+1}} f_i^s + \frac{r_{i+1} + p_{i+1}}{r_{i+1} p_{i+1}} f_i^b - \frac{1}{p_{i+1}} f_{i+1}^b - \frac{1}{d} z_i^s f_i^s - \frac{1}{d} z_i^s f_i^b + \frac{1}{p_{i+1}} f_i^s f_{i+1}^b = \frac{1}{r_{i+1}}$$
$$(i = 1, 2, \ldots, N-1)$$

$$\begin{aligned} & f_1^s = 0, \quad f_n^b = 0 \\ & f_i^s + f_i^b \leq 1 - \tfrac{d}{d_{maxi}} \quad (i = 1, 2, \ldots, N) \\ & f_i^s \geq 0, \quad f_i^b \geq 0 \quad (i = 1, 2, \ldots, N) \\ & z_i^b \geq 0, \quad z_i^s \geq 0 \quad (i = 1, 2, \ldots, N-1) \end{aligned}$$

The optimal buffer sizes are given by

$$B_i = z_i^b + z_i^s \quad (i = 1, 2, \ldots, N-1)$$

4.8 The hedging point

Since all machines are unreliable and can be starved or blocked, there is difference between the hedging point $(z_1, z_2, \ldots, z_N)$ and the average surplus $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_N)$, which is the time average over the planning horizon of $(x_1, x_2, \ldots, x_N)$. The relation is governed by

$$z_i = \bar{x}_i + \Delta_i \quad (i = 1, 2, \ldots, N) \tag{70}$$

where $\Delta_i (i = 1, 2, \ldots, N)$ is the surplus loss at Machine i. The surplus loss is approximately given by (see Section 3.9)

$$\Delta_i = \frac{r_i p_i}{r_i + p_i} \frac{d}{2} \left(\frac{U_i}{U_i - d}\right) \left\{\left(\frac{1}{r_i}\right)^2 + \left(\frac{f_i^s}{p_i}\right)^2 + \left(\frac{f_i^b}{p_i}\right)^2\right\} \quad (i = 1, 2, \ldots, N). \tag{71}$$

According to the performance specification (4) of Section 4.3, we would like to minimize the absolute value of surplus x_N. With the same reasoning as we did for the two-machine, one-part-type system, we choose the hedging point $(z_1, z_2, \ldots, z_N)$ such that

$$\bar{x}_N = 0. \tag{72}$$

From (57), (70), and (72), the hedging point must satisfy

$$\begin{aligned} z_N &= \Delta_N; \\ z_i &= \sum_{k=i}^{N-1} z_k^b + \Delta_N, \qquad (i = 1, 2, \ldots, N-1). \end{aligned} \tag{73}$$

4.9 The algorithm

We have extended the real-time feedback control algorithm to N-machine, one-part-type production lines. The steps of the algorithm are summarized in the following:

Step 1: Collect the input data set, which consists of the failure rates p_i, the repair rates r_i, and the processing time τ_i for Machine i $(i = 1, 2, \ldots, N)$, and the demand, d, which should be a member of the long term capacity set (12).

Step 2: Calculate the buffer hedging levels, z_i^b (i=1,2,.... N), and hedging spaces, z_i^s (i=1,2,....N), and the starvation and blockage fractions for each machine by solving the nonlinear program (69). Then, calculate the buffer size for each machine by summing the buffer hedging level and hedging space.

Step 3: Calculate the components of the hedging point, $(z_1, z_2, \ldots, z_N)$, according to (73).

Step 4: Using the feed-back information of surplus x, and machine state α_i (i=1,2,...,N), calculate the production rates, u_i (i=1,2,...,N), in real time by solving the linear program (55).

Step 5: The loading times for each machine are determined by the heuristic staircase strategy. That is, whenever the actual cumulative production is less than the integral of the production rate, load a part into the machine.

Step 6: If the demand or any of the machine parameters changes, go to Step 2.

i	f_i^s	f_i^b	z_i^s	z_i^b	B_i	z_i
1	0.0	0.44	0.0	1.4	2	3.96
2	0.0	0.30	1.46	0.0	2	2.56
3	0.18	0.13	0.79	1.36	3	2.56
4	0.29	0.18	0.0	0.0	1	1.2
5	0.35	0.0				1.2

Table 2: The buffer sizes and hedging point for a five-machine, one-part-type system (d=0.7)

4.10 Example

For the simulation, we still use the three-level hierarchical policy described in Section 3.10. At the top level, the buffer sizes and hedging point are calculated by using a commercially available software package [35]. HIERCSIM [36] is used for the next two level simulation. The system consists of five machines and four buffers. The parameters are chosen as follows:

$$\begin{array}{lll} r_1 = 0.5, & p_1 = 0.3, & \tau_1 = 0.5 \\ r_2 = 0.2, & p_2 = 0.05, & \tau_2 = 0.3 \\ r_3 = 0.3, & p_3 = 0.2, & \tau_3 = 0.6 \\ r_4 = 1.2, & p_4 = 0.1, & \tau_4 = 0.4 \\ r_5 = 0.3, & p_5 = 0.1, & \tau_5 = 0.7 \end{array}$$

Given that the demand is 0.7, the buffer sizes and hedging point are calculated at the top level by solving (69) and (73), and listed in Table.2.

Fig.20 illustrates the simulation results of the cumulative production. The straight line is the cumulative demand. The upper curve is the input of the raw parts at Machine 1. The lower curve is the output of the final products at Machine 5. The dashed lines are the results at the middle level by solving (55).

Fig.21 shows the history of the level of Buffer 3 which lies between Machine 3 and Machine 4. The dashed lines are the second level result which is determined by $b_3(t) = x_3(t) - x_4(t)$. The solid lines are the actual count of the parts in the buffer.

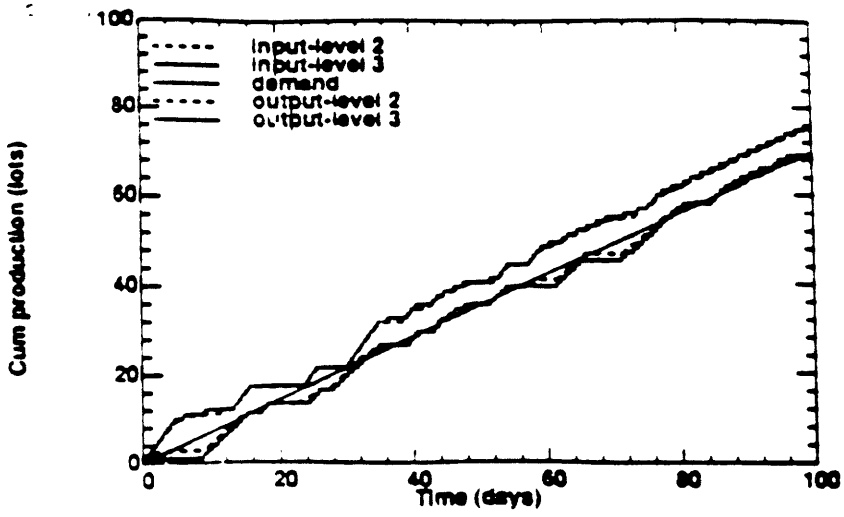

Figure 20: The simulation result of cumulative production of the five-machine and one-part-type system

To see the effects of buffer levels and sizes, we increase the demand to 0.85 without changing the buffer sizes and the hedging point (Table 2). The simulation result in Fig. 22 shows that the production fell behind the demand. That is, with the buffer levels and sizes in Table 2, the system is starved or blocked too much to achieve the demand, 0.85.

Given that the demand is 0.85, the desirable buffer sizes and hedging point are calculated and listed in Table 3. With the appropriate buffer sizes and hedging point (Table.3), the actual production follows the demand closely (see Fig. 23).

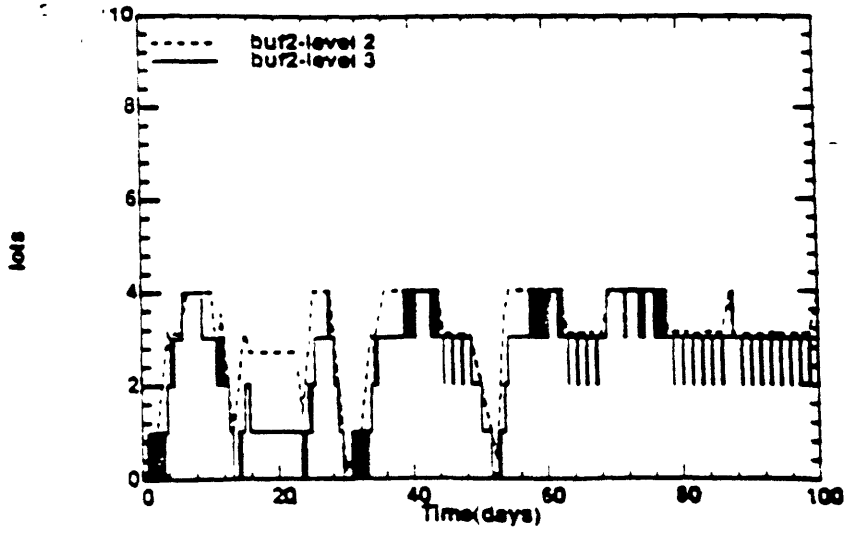

Figure 21: The history of buffer level

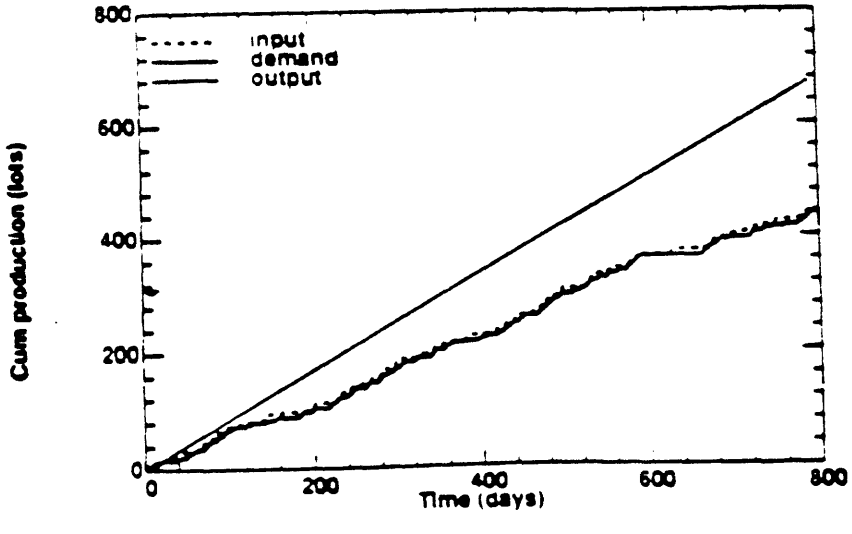

Figure 22: The effects of infeasible buffer levels and sizes

i	f_i^s	f_i^b	z_i^s	z_i^b	B_i	z_i
1	0.0	0.32	0.0	1.7	2	6.84
2	0.0	0.15	2.08	1.25	4	5.13
3	0.15	0.0	2.54	2.68	6	3.89
4	0.14	0.22	0.0	0.0	1	1.2
5	0.21	0.0				1.2

Table 3: The buffer sizes and hedging point for a five-machine, one-part-type system (d=0.85)

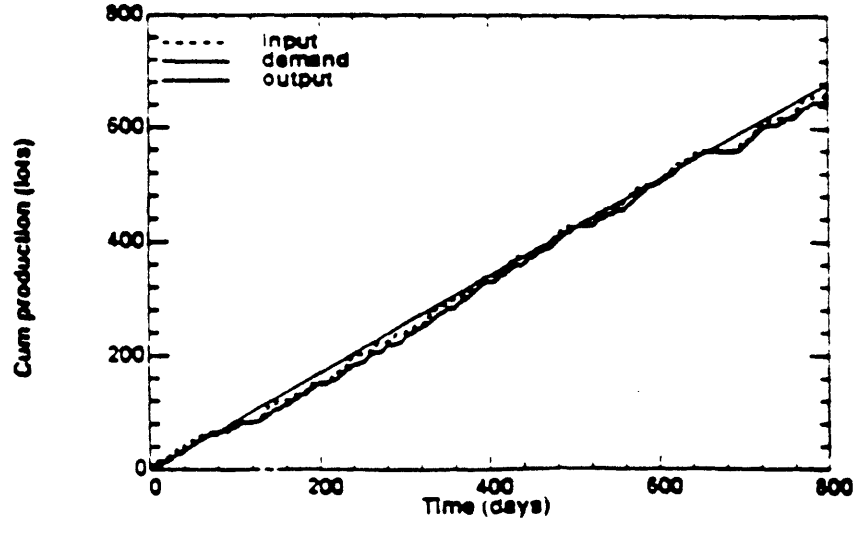

Figure 23: The effects of desirable buffer levels and sizes

5 Summary

A real-time feedback control algorithm is developed for scheduling single-part-type production lines in this paper. The WIP inventory is allocated dynamically according to the production demand and machine parameters. The simulation results verify that the algorithm works well.

The method developed in this paper is extended to multiple- part-type and reentrant production systems in [31] and [32]. The algorithm is also modified in [33] such that the formulation for buffer hedging levels and hedging spaces becomes linear. In many cases, the linear formulation gives us equally good results without the trouble of solving the non-linear programming problem.

6 Acknowledgments

The authors would like to thank B. Darakananda for his help about the simulations. The research reported in this paper has been supported by the Defense Advanced Research Projects Agency under contracts N00014-85-k-0213 and MDA-972-88-K-0008.

References

[1] S. C. Graves, "A Review of Production Scheduling," *Operations Research*, vol. 29, no. 4, pp. 646–675, 1981.

[2] P. Afentakis, B. Gavish, and U. Karmarkar, "Computationally Efficient Optimal Solutions to the Lot-sizing Problem in Multi-stage Assembly Systems," *Management Science*, vol. 30, no. 2, pp. 222–239, 1984.

[3] B. J. Lageweg, J. K. Lenstra, , A. H. G., and R. Kan, "Job-Shop Scheduling by Implicit Enumeration," *Management Science*, vol. 24, no. 4, pp. 441–450, 1977.

[4] B. J. Lageweg, J. K. Lenstra, A. H. G., and R. Kan, "A General Bounding Scheme for the Permutation Flow-Shop Problem," *Operations Research*, vol. 26, no. 1, pp. 53–67, 1978.

[5] E. F. P. Newson, "Multi-Item Lot Size Scheduling by Heuristic, Part I: With Fixed Resources," *Management Science*, vol. 21, no. 10, pp. 1186–1193, 1975.

[6] E. F. P. Newson, "Multi-Item Lot Size Scheduling by Heuristic, Part II: With Variable Resources," *Management Science*, vol. 21, no. 10, pp. 1194–1203, 1975.

[7] C. H. Papadimitriou and P. C. Kannelakis, "Flowshop Scheduling with Limited Temporary Storage," *Journal of the ACM*, vol. 27, no. 3, 1980.

[8] H. M. Wagner and T. M. Whitin, "Dynamic Version of the Economic Lot Size Model." *Management Science*, vol. 5, no. 1, pp. 89–96, 1958.

[9] H. Chen, M. Harrison, A. Mandelbaum, A. V. Ackere, and L. Wein, "Empirical Evaluation of A Queueing Network Model for Semicondctuor Wafer Fabrication," *Operations Research*, vol. 36, no. 2, 1988.

[10] C. R. Glassey and M. G. C. Resende, "Close-Loop Job Release for VLSI Circuit Manufacturing," Tech. Rep. ORC#:87-8a, University of California at Berkeley, 1986.

[11] L. M. Wein, "Scheduling Semiconductor Wafer Fabrication," tech. rep., Stanford University, 1987.

[12] G. R. Bitran, E. A. Haas, and A. C. Hax, "Hierarchical Production Planning: A Single-Stage System," *Operations Research*, vol. 29, no. 4, pp. 717–743, 1981.

[13] M. A. H. Dempster, L. J. M. L. Fisher, B. J. Lageweg, J. K. Lenstra, A. H. G., and R. Kan, "Analytical Evaluation of Hierarchical Planning Systems," *Operations Research*, vol. 29, no. 4, pp. 707–716, 1981.

[14] S. B. Gershwin, "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems," *Proceedings of the IEEE, Special Issue on Dynamics of Discrete Event Systems*, vol. 77, no. 1, pp. 195–209, 1989.

[15] S. C. Graves, "Using Lagrangean Relaxation Techniques to Solve Hierarchical Production Planning Problems," *Management Science*, vol. 28, no. 3, pp. 260–275, 1982.

[16] A. C. Hax and H. C. Meal, "Hierarchical Integration of Production Planning and Scheduling," *North Holland/TIMS, Studies in Management Sciences*, vol. 1, Logistics, 1975.

[17] R. Rishel, "Dynamic Programming and Minimum Principles for Systems with Jump Markov Disterburces," *SIAM Journal on Control*, vol. 13, no. 2, 1975.

[18] J. Kimemia and S. B. Gershwin, "An Algorithm for the Computer Control of a Flexible Manufacturing System," *IIE Transactions*, vol. 15, no. 4, pp. 353–362, 1983.

[19] J. Tsitsiklis, "Convexity and Characterization of Optimal In a Dynamic Routing Problem," Tech. Rep. LIDS-R-1178, MIT, 1982.

[20] S. B. Gershwin, R. Akella, and Y. F. Choong, "Short-Term Production Scheduling of an Automated Manufacturing Facility," *IBM Journal of Research and Development*, vol. 29, no. 4, pp. 392–400, 1985.

[21] R. Akella and P. R. Kumar, "Optimal Control of Production Rate in a Failure Prone Manufacturing System," *IEEE Transaction on Automatic Control*, vol. AC-31, no. 2, pp. 116–126, 1986.

[22] G. Van Ryzin, "Control of Manufacturing Systems With Delay," Master's thesis, MIT, 1987.

[23] R. Akella, Y. F. Choong, and S. B. Gershwin, "Performance of Hierarchical Production Scheduling Policy," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. CHMT-7, no. 3, 1984.

[24] O. Z. Maimon and Y. F. Choong, "Dynamic Routing in Reentrant Flexible Manufacturing System," *Robotic and Computer-Aided Manufacturing*, vol. 3, pp. 295–300, 1987.

[25] O. Z. Maimon and S. B. Gershwin, "Dynamic Scheduling and Routing for Flexible Manufacturing Systems that have Unreliable Machines," *Operations Research*, vol. 36, no. 2, pp. 279–292, 1988.

[26] A. Sharifnia, "Production Control of a Manufacturing System with Multiple Machine States," *IEEE Transactions on Automatic Control*, vol. AC-33, no. 7, pp. 620–625, 1988.

[27] M. N. Eleftheriu, *On The Analysis of Hedging Point Policies of Multi-Stage Production Manufacturing Systems*. PhD thesis, Rensselaer Polytechnic Institute, 1989.

[28] R. Conway, W. Maxwell, J. O. McClain, and L. J. Thomas, "The Role of Work-In-Process Inventory In Serial Production Lines," *Operations Research*, vol. 36, no. 2, 1988.

[29] D. Y. Burman, F. J. Gurrola-Gal, A. Nozari, S. Sathaye, and J. P. Sitarik, "Performance Analysis Techniques for IC Manufacturing Lines," *AT&T Technical Journal*, vol. 65, no. 4, 1986.

[30] A. H. Zeghmi, *Inventory Buffers For a Production Line With Controlable Production Rates*. PhD thesis, MIT, 1985.

[31] X. Bai and S. B. Gershwin, "Scheduling Manufacturing Systems With Work-In-Process Inventory Control: Multiple-Part-Type Systems," tech. rep., MIT, Operations Research Center, 1990.

[32] X. Bai and S. B. Gershwin, "Scheduling Manufacturing Systems With Work-In-Process Inventory Control: Reentrant Systems," tech. rep., MIT, Operations Research Center, 1990.

[33] X. Bai and S. B. Gershwin, "Scheduling Manufacturing Systems With Work-In-Process Inventory Control: Linear Formulation," tech. rep., MIT, Operations Research Center, 1990.

[34] X. Bai and S. B. Gershwin, "A Manufacturing Scheduler's Perspective on Semiconductor Fabrication," tech. rep., MIT, Laboratory for Manufacturing and Productivity, 1989.

[35] A. Brook, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*. The Scientific Press, 1988.

[36] B. Darakananda, "Simulation of Manufacturing Process Under a Hierarchical Control Structure," Master's thesis, MIT, 1989.