SCHEDULING MANUFACTURING SYSTEMS WITH WORK-IN-PROCESS INVENTORY CONTROL

by

Sherman Xiewei Bai

Bachelor of Science in Mechanical Engineering
Luoyang Institute of Technology, China, January, 1981
Master of Science in Engineering Mechanics
Southwestern Jiaotong University, China, September, 1984
Master of Science in Mechanical Engineering
Massachusetts Institute of Technology, September, 1991

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 1991

Signature of Author: _____

Operations Research Center
August, 1991

Certified by: _____

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

Accepted by: _____

Thomas L. Magnanti
George Eastman Professor of Management Science
Co-Director of Operation Research Center

Scheduling Manufacturing Systems With Work-In-Process Inventory Control

by

Sherman Xiewei Bai

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

A real-time feedback control algorithm is developed for scheduling manufacturing systems 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 (WIP) inventory as low as possible. In the production scheduling algorithm, the level of work-in-process inventory is involved in three phases of the decision making procedure. For the long-term capacity planning, the buffer sizes and average buffer levels are allocated so as to have enough system capacity to achieve the demand. Consequently, we determine the total average work-in-process inventory and its allocation. When a demand or machine parameter changes, we recompute the WIP distribution. The scheduling system recalculates the production rates in real-time whenever a machine fails or is starved or blocked. For selecting the actual loading times, starvation and blockage are also a concern of the decision making. When the system reaches the steady-state, the production control algorithm generates the same policy as in a KANBAN system. That is, the production rates are equal to the demand. Moreover, if the system drifts away from the steady-state due to random events such as machine failures and starvation or blockage, real-time policy changes are made such that the system recovers as soon as possible.

To begin with, we study a two-machine, one-part-type system, to get insight into the buffer effects and production control policies. The result from the simple case is extended, step by step, to more and more complicated and realistic systems. As a real world application, we study a wafer fabrication facility, the Integrated Circuit Laboratory of MIT. The feedback control policy has been used for the simulation of

the MIT Twin-Well CMOS process production control. Simulation results are presented.

Thesis committee: Dr. Stanley B. Gershwin, Chairman and Supervisor,
Senior Research Scientist,
Department of Mechanical Engineering, MIT.
Dr. Michael C. Caramanis,
Professor of Operations Research,
Department of Manufacturing Engineering,
Boston University.
Dr. Lawrence M. Wein,
Professor of Management Science,
The Sloan School of Management, MIT.

Acknowledgments

I would like to thank my thesis supervisor, Dr. Stanley B. Gershwin, for his inspiration, guidance, and constant support during my study at MIT. His contribution to my education goes far beyond his guidance of this thesis research. I would like to thank the members of my Thesis Committee, Professor Michael C. Caramanis and Professor Lawrence M. Wein, for their interest in my work and thoughtful comments and suggestions. I would also like to thank the members of my Doctoral Degree Committee, Professor Amedeo R. Odoni and Professor Robert M. Freund, for their encouragement and help. I thank all the participants of the Computer-Aided-Fabrication (CAF) Project of MIT for the stimulating discussions during the weekly project meetings. The friendly and stimulating atmosphere in the Operations Research Center has been a constant source of support. I would like to thank all the members of ORC for their friendship and help.

I am deeply indebted to my wife, Lisa, for her constant support and understanding, not only during the course of this research, but also in prior and succeeding years. Most of all, I am grateful to my parents for their support and encouragement. Without them, I would not have been here.

The research reported in this thesis has been supported by the Defense Advanced Research Projects Agency under contracts N00014-85-k-0213 and MDA-972-88-K-0008.

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 makes the scheduling of manufacturing systems difficult. For example, in a semiconductor fabrication factory, dozens 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 lateness.

Lateness is the time difference between actual production and demand. If the actual production is ahead of the demand (negative lateness or earliness), final product inventory accumulates. If the actual production is behind the demand (positive lateness), 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 lateness 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 thesis, 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 approximate linear 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 thesis, 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 allocated 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 [18]. Many of the works before the early 80's are based on combinatorial optimization/ integer programming or mixed integer methods ([1], [24], [25], [30], [31], [32], and [37]). Some other works are based on queuing network models ([10], [17], and [38]).

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 ([7], [13], [15], [19], and [20]). 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 [33], Kimemia and Gershwin [23] investigated the optimal flow controller's structure and determined that it is a hedging point feedback control policy. Tsitsiklis [35] proved the convexity of the value function that satisfies the Hamilton-Jacobi-Bellman equations and determines the optimal controller. Gershwin, Akella, and Choong [16] proposed a heuristic approximation of the value function. Akella and Kumar [3] solved analytically the Hamilton-Jacobi-Bellman equation to obtain the optimal value function for a simple one-part-type, one-machine system. Van Ryzin [36] 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 [2], [28], [29], [34], and [14].)

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. [11] studied the effects of WIP in serial production lines. Burman et al. [9] investigated the relation between the WIP level and the system performance of integrated circuit

manufacturing lines. Zeghmi studied inventory buffers in a production line [39]. 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 Outline of the thesis

A real-time feedback control algorithm is developed for scheduling manufacturing systems. Three important classes of activities are considered. They are operations, machine failures, and starvation or blockage. The algorithm also responds to changes of demand and machine parameters. The major contribution of this thesis is that we explicitely introduce buffer sizes, average buffer levels, and starvation and blockage fraction as control parameters for long term capacity planning of manufacturing systems. Initially, the production control problem is formulated as a dynamic program. By using the Bellman's equation, the dynamic program is transformed to a stochastic linear program, which represents a feedback control law. Whenever a failure or repair occurs, or the production surplus reaches a boundary, the production rates are recalculated according to the feedback information on the system state . To estimate the unknown parameters in the feedback control linear program, namely, the hedging point and buffer sizes, a frequency-duration method is developed to formulate an approximation of the relationship among buffer levels, buffer spaces, starvation, blockage, machine parameters, and demand. Under the long term capacity constraints, a nonlinear program is set up to minimize the buffer level and space. Consequently, we determine the Work-In-Process inventory distribution in the system. Whenever a demand or machine parameter changes, we recompute the WIP distribution by solving the nonlinear program.

In Section 2, we study a real wafer fabrication faciclity, the MIT Integrated Circuit Laboratory, to find the important phenomena which should be taken into account for production scheduling. Section 3 describes the systems which are under study. The system assumptions are described. Notation and terminology are introduced there. The activities, constraints, and objectives are discussed. The simplest case of the manufacturing system model, two-machine, one-part-type tandem production lines, is studied in Section 4. We investigate the buffer effects and the relation between WIP and starvation or blockage. A hedging policy is derived from the dynamic program. The results of the simple case are extended to N-machine, one-part-type

systems in Section 5. Each part travels in a fixed sequence: Machine 1, Buffer 1, ..., Machine N. A machine in the middle of the production line can be either starved or blocked. The production control algorithm for single-part-type systems is then extended to multiple-part-type tandem production lines. The two-machine, two-part-type systems are studied in Section 6 and N-machine, M-part-type systems in Section 7. An approximate linear system method is developed to allocate capacity for each part type at each machine such that the parameters of the feedback controller are estimated by solving a nonlinear program for each single-part-type sub-system. We study single-part-type reentrant systems in Section 8, and multiple-part-type reentrant systems in Section 9. The approximate linear system method is extended to allocate capacity for each operation at each machine for reentrant systems. Based on the algorithms developed in the preceding sections, the bounds on the WIP inventory are established in Section 10.

As an application, the real-time scheduling algorithm is used to simulate the semiconductor fabrication production control at the integrated circuit laboratory of MIT. In Section 11, we discuss the simulation of the MIT Twin-Well COMS process production control. Also, a multi-process example is presented there. Some implementation issues are briefly discussed in Section 12. Concluding remarks are in Section 13.

2 A wafer fabrication facility

In this section, we describe the important phenomena in semiconductor fabrication which should be considered for scheduling. We focus our attention on collecting and understanding events (both those under the manager's control and those that are not), and describing the related concepts, such as fabrication processes, operation sets, production machines, support equipment, operation workers, support technicians, activities, objectives, etc. Most of our observations are taken from the MIT Integrated Circuit Laboratory.

2.1 Overview of semiconductor manufacturing

The overall semiconductor manufacturing procedure can be roughly divided into six subprocedures: circuit design and mask preparation, wafer preparation, wafer fabrication, probe test and sort, assembly, test and classify. Wafer fabrication, probe test, sort, and assembly take place in a *clean room*, a room in which the the atmosphere is purified so that the particulate count and humidity are kept to within very narrow limits. Many of the support machines described below are used for the maintenance of atmospheric quality. Figure 1 illustrates the manufacturing process flow for a semiconductor firm. The solid arrows indicate the material flow while the dashed arrows represent the information flow. In terms of the product structures, the six subprocedures are described in the following:

Circuit design and mask preparation: According to marketing information and technology development, new circuits are laid out with the aid of computers. A mask is a glass plate with a hard opaque surface material such as chromium. An image is created in a mask via a pattern generator and associated processes which remove material using a directed electron beam in a high vacuum.

Wafer preparation: This process begins with quartz which is refined into electronics grade silicon. The silicon is grown into cylindrical crystals four to six inches in diameter. Some newer systems use eight inch wafers and there is experimental work with twelve inch wafers. During growth, controlled amounts of dopant atoms are incorporated to the crystal. The cylinders are sliced into wafers which are then buffed and polished. These wafers are then ready for fabrication of circuits.

Wafer fabrication: A wafer fabrication procedure is performed in a *wafer fab* factory, which contains a number of machines and a workforce. Corresponding to dif-

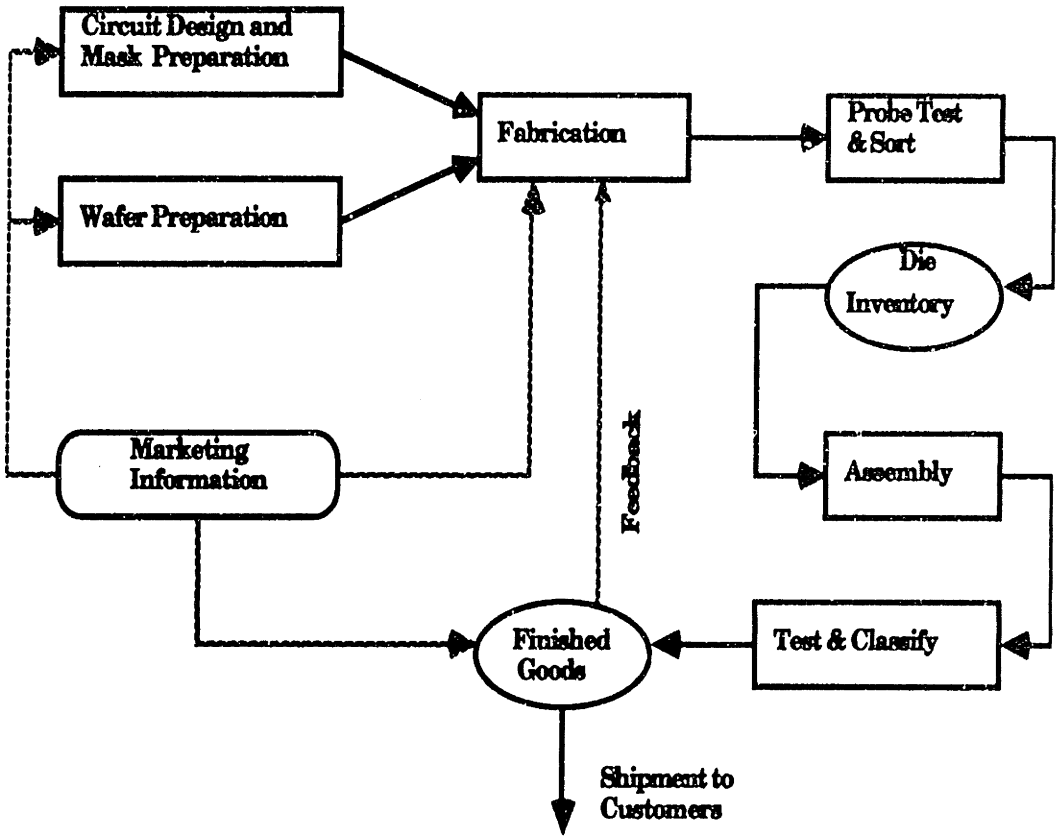

Figure 1: Semiconductor manufacturing procedure

ferent final products, a number of *fabrication processes* are run in a wafer fab. Each of the fabrication processes is a sequence of *processing steps* the wafers pass through during the manufacturing. Common processes include CMOS, NMOS, and Bipolar. Less common processes serve to make monolithic accelerometers, flow transducers, microphones, etc. Each of the processes creates useful three-dimensional structures, such as transistors, capacitors, resistors, and transducers, on the wafers. Each potential integrated circuit device is called a *die*, which consists of a number of those structures.

Wafer probe and sort: Each die on a wafer is inspected by using a wafer probe. Rejects are marked (sorted) so as to be discarded in the assembly procedure. The inventory of probed wafers is called *die inventory*. The wafer probe process consists of one or only a few steps. In some firms, these steps are thought of simply as the final steps of the fabrication processes [26].

Assembly: In this process, wafers are sawed and rejected die are discarded. Good die are electrically connected and sealed in packages of various types [26].

Test and classify: Two test processes, raw test and final test, are involved here. In the raw test (or class test) process, packaged devices are subjected to a series of tests to determine device performance. As a result of this test, the devices are categorized into bins based on the measured performance of one or more attributes such as device speed, power consumption, tolerance of voltage variance, etc. Final tests are performed before delivering products to customers, according to the orders [26].

In the following subsections, we focus on the wafer fabrication procedure. We discuss fabrication processes, operation sets, machines, and human resources involved in wafer fabrication. Activities, constraints, and objectives are also listed.

2.2 Semiconductor wafer fabrication

In this section, we focus on the wafer fabrication procedure. We discuss fabrication processes, operation sets, machines, and human resources involved in wafer fabrication. Activities, constraints, and objectives are also listed.

2.2.1 Fabrication processes and operation sets

In a wafer fabrication factory, wafers are grouped in *lots*. The number of wafers in a lot is usually a constant. They are grouped this way because many machines are

designed to work on many wafers at the same time; because changing operations on some machines can be expensive; and because this makes it easier to trace the path of a wafer through the system for the purpose of determining causes of poor yield. Usually a wafer fab factory produces more than one product. For each product type, an operation sequence is performed to create the required structures on the wafers. We refer to the predefined sequence of operations as the *fabrication process* (or fab process). Since hundreds of operations are involved in a fab process, the operations are divided into groups, called *processing steps* (or unit processes). Each processing step has an associated *operation set* (or opset) which consists of several operations in sequence and information used for the operations such as machine name, processing time, and handling time. An opset also specifies some parameters like furnace recipe number and photomask ID.

Figure 2 depicts a simple fab process for two-mask polysilicon gate capacitor, which consists of fifteen processing steps. Each block in the graph represents a processing step. Each processing step has a associated operation set. The first processing step is field oxidation, and the associated opset is dfield5k.set, and so on. All the operations at the present step must be completed before wafers go to the next step.

Table 1 illustrates an example of opset, named dfield5k.set. It consists of three operations, RCA clean, diffusion, and inspection, in sequence. The machines used for the operations are: RCA wet station, furnace B1, and nanospec, respectively. The wafers undergoing this opset visit the RCA wet station first, then furnace B1, followed by inspection at nanospec. All three operations must be completed before the wafers are ready for the next opset. The required operation times and handling times are listed in the table. The operation time is the amount of time needed to finish the operation. The handling time is the amount of time during which the worker performs the operation (such as the loading and unloading of the diffusion operation). The specified parameters provide further information to support each operation. For example, recipe# 240 contains the information about the temperature set-up and gas inputs for the furnace, and so on.

2.2.2 Inspection

Most processing steps are ended by an inspection operation. The purpose of inspection is to control the product quality and to test machine performance. Decisions are made according to the inspection results. For example, good wafers which pass

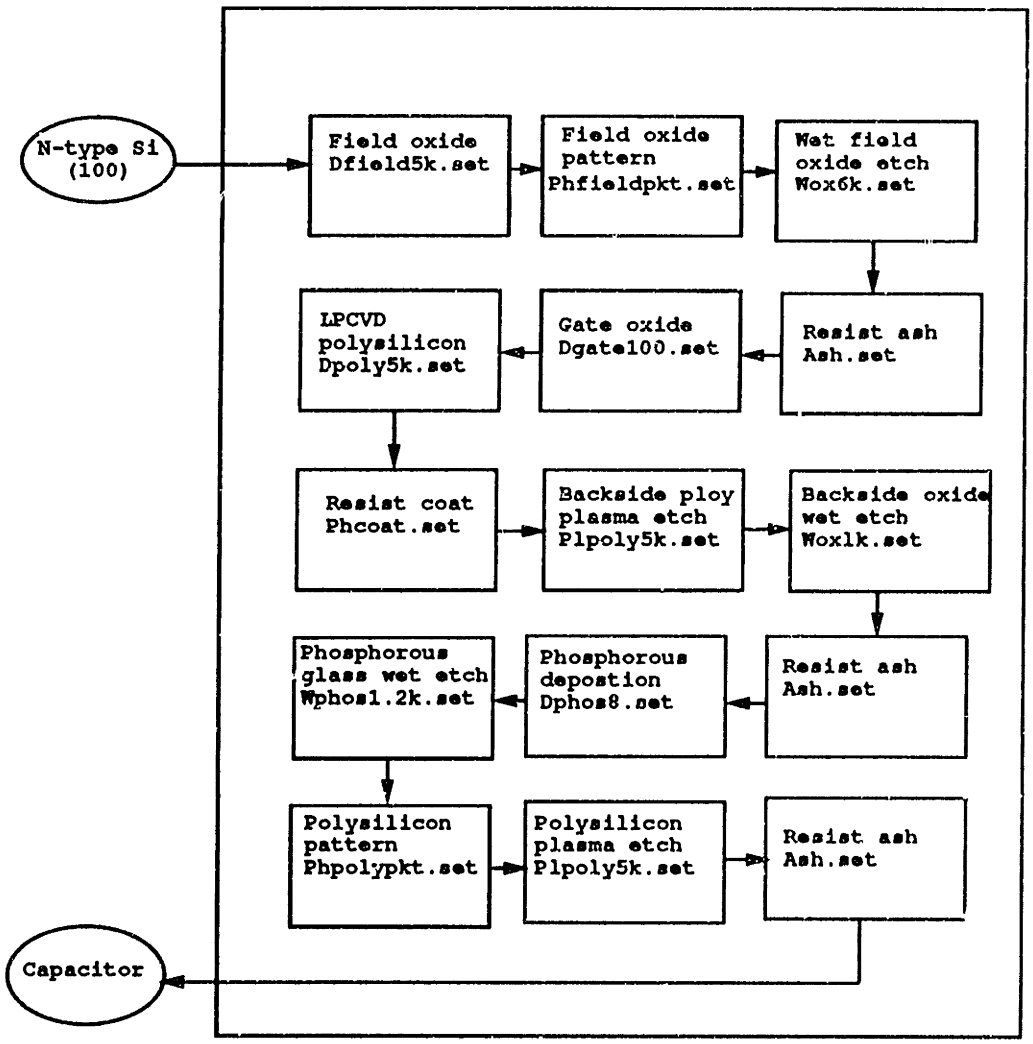

Figure 2: Two mask poly gate capacitor process

dfield5k.set					
No.	operation	machine	parameter	op time	hdl time
1	RCA clean	RCA wet station		2 hrs 0 min	2 hrs 0 min
2	diffusion	furnace B1	recipe: 240 film spec avg: 5100 range: 500	7 hrs 30 min	0 hrs 30 min + 0 hrs 30 min
3	inspection	nanospec	thickness top:___ center:___ left:___ right:___ bottom:___	0 hrs 15 min	0 hrs 15 min
			total time	10 hrs 0 min	3 hrs 30 min

Table 1: An example of opset: dfield5k.set

inspection are sent to a downstream buffer to queue for the next processing step, and bad wafers which fail inspection are either sent for rework or scrapped to the trash can. Most rework occurs at the photo step.

Not all of the production wafers go to inspection machines. Usually, only control (pilot) wafers or sample wafers are sent for inspection. When a wafer fails inspection, the cause of the failure is carefully investigated, and then support technicians are notified to maintain or repair machines.

Frequently, rework requires off-route operations (i.e., extra operations) such as an extra strip in an etch area sink. In this case, one or more machines would be visited by the rework wafers on the way back to the upstream buffers. Additionally, rework could also cause *lot splitting*. That is, a portion of a lot may be reworked while the remainder waits for the rework sub-lot to catch up or, alternatively, the two sub-lots continue through the process independently.

2.2.3 Machines

In this subsection we describe the machines involved in semiconductor fabrication. In terms of function of the machines, we divide them into two basic groups, i.e., *support equipment* and *production machines*. Machines in different wafer fab factories may not be exactly the same. The discussion here is based on the Integrated Circuit

Laboratory of MIT.

Support equipment

Support equipment in a wafer fab is never visited by the wafers. The operation states of production machines depend on the status of these machines. If one of the support machines is down or undergoing maintenance, one or more production machines will be down. In general, performing maintenance on each of the support machines causes a shutdown of the clean room if the system is fully utilized. The IC Laboratory of MIT is run only twelve hours a day and five days a week. Consequently, maintenance is usually done during the off time. We refer to the whole set of support equipment as the *house system.* Following is a list of support equipment in the IC Laboratory of MIT.

Clean air flow: The laminar flow of filtered air is used to maintain a dust-free environment in clean rooms. This flow is provided by fans mounted on the roof, by a maze of ducts, and by filters located above the clean rooms.

Emergency water: This is ordinary cold city water. It is used in emergencies to wash a person splashed by chemicals.

Process vacuum: This is used to pick up wafers and hold wafers in place during processing.

Cleaning vacuum: These are used to vacuum the rooms clean or clean up spills. The centrally located pumps collect waste, and the tanks must be emptied when full. The hoses are attached to the plug points in the wall from which pipes lead to the machines.

Compressed air: This can be used to operate pneumatic equipment. A dessicant at the main pumps dries the air.

Process chilled water (pcw): This is water used to control the temperature of equipment. It is recycled and reheated or recooled as required.

Solvent tank: A variety of chemicals are used and dumped. They are poured down the solvent waste sinks where they drain into a special underground storage tank and then are taken to long term storage.

City waste: Other chemicals are neutralized to ph of 7 and dumped into the city sewer.

Fume exhaust: This is a system of ducts and fans that perform the work required to exhaust toxic fumes from fume hoods.

Power: Several different voltages, currents and phases are available. All of the machines require electricity to run. A backup power supply exists, so power is essentially always available.

Deionized (DI) water: This can be considered as a storage tank with fixed volume and a production system with a maximum replenishment rate.

Humidity controllers and temperature controllers: Excessive moisture can damage wafers by accelerating undesirable chemical reactions on chemically active surfaces. Therefore the humidity in the clean rooms is controlled. The temperature in the clean room is controlled to within $\pm$ 1 degree because wafer dimensions depend on temperature and some equipment are highly temperature sensitive.

Tank farm: Semiconductor fabrication requires clean dry gases. At MIT, three tanks of Argon, Nitrogen and Oxygen supply the building with these gases.

Local gases: These are small tanks of gases placed in cabinets near the equipment which use them.

Local gas vents: To ensure that the leakage of a gas cylinder does not poison anyone, the air from the gas cabinets is exhausted.

Safety alarms: Fire and gas leaks are reported by safety alarms. The clean room is equipped with fire extinguisher, fire pumps, hydrogen monitors, and toxic gas monitors.

Production machines

Wafers visit the production machines and occupy them for certain periods of time. These production machines impose capacity constraints on the production rates. Following is a list of production machines in the MIT ICL.

HMDS vacuum bake vapor prime and image reversal system (Model 3/10): Wafers are sprayed with a dehydrating chemical, HMDS, at 150° C. A dedicated commercial oven is used for this operation, which requires house vacuum, power, and a dry and particle-free environment.

Photoresist coater & developer (GCA 1006 Wafertrack): After HMDS, wafers are loaded on the GCA wafer coating track for photoresist coating. The pre-exposure bake is done in the in-line contact oven module. After the exposure, the exposed wafers are developed on the GCA developing track. Post-development hard baking is done in a in-line oven.

Wafer stepper system (GCA 4800 DSW): This equipment does the exposure. The pattern transfer from an appropriate mask is carried out in a GCA 4800 , 10X direct

step-on wafer system equipped with a 10-78-45 g-line lens. All the relevant information about stepping a given mask pattern on a wafer are stored in a job specification file in a dedicated PDP-11. Operation times are longer for wafers with smaller feature sizes, due to the greater precision required for alignment.

Asher: In all baseline steps, resist is removed in a photoresist stripper (Drytek Model Megstrip 6).

Dry etching: This is done in a plasma. Due to an applied voltage the ions in the plasma bombard the silicon target perpendicularly to the surface. The surface is etched away where it is not covered by resist. There are three plasma etchers in the MIT ICL. Silicon-nitride etching and polysilicon etching are done in etcher-1 (LAM 480). Etcher-2 (LAM 594) is used for oxide etching. Metal etching is done in etcher-3 (LAM 690).

Wet etching (wet chemical process station): In addition to the dry etching steps, stripping of oxide and silicon nitride is done using wet chemistries. The wafer is placed in a bath, and the etch eats away at the parts of the layer not covered by resist.

RCA cleaning: Before wafers are loaded into furnaces, they are cleaned in a wet station using various chemicals.

Oxidation furnaces: The MIT ICL is equipped with BTU oxidation furnaces. These machines are used to expose wafers to hot gases at a variety of pressures for oxidation or diffusion. Furnaces consist of quartz tubes, gas controllers, temperature controllers, a suspended loading system, and a dedicated PDP-11 computer.

Chemical vapor deposition (CVD): Layers can be deposited on wafers, in furnaces, from gases by a process called chemical vapor deposition (CVD). If the process occurs at low pressure, it is called low pressure chemical vapor deposition (LPCVD). There are four LPCVD tubes in MIT ICL.

Sputtering system (CVC 601): AlSi is the baseline first level metal. It is deposited in a CVC sputtering system in the dc megnetron sputtering mode.

Ion implanter: In MIT ICL, ion implantation is done in an ETON medium current machine (model NV 3206). This machine injects ions into wafers.

Microscope: This is used for inspection in opsets like photo and etching.

Surface profiler (DEKTAK IIA): A surface profiler is essentially a phonograph needle that measures the height of the bump that it crossses.

Ellipsometer (GSC L116BL-26A): This measures the reflection of a laser beam off the measured thin film layer.

Junction sectioner (PIC 2015D): This machine carves a groove in the wafer, stains the wafer, and then a microscope is used to identify the depth of the dopant.

Automatic four point probe: This measures the resistivity of the incoming silicon wafers and of the doped layers formed or grown on it.

CV-plotter (MDC CSM-16): This machine measures the space charge capacitance as a function of reverse bias voltage on a junction.

Film thickness measurement system (Nanospec/AFT 010-0180): This equipment measures the thickness of thin films on wafers.

2.2.4 Human resources

In a wafer fab, a workforce is needed to run the fab processes. Technicians do operations and maintain equipment. Process engineers are responsible for the execution of designed processes and for monitoring machine performance and device characteristics. According to their responsibilities, we group them as *operation workers* and *support technicians.*

Operation workers are the group of people who are in charge of operations. They either carry wafers from one machine to another or are assigned to a certain machine and spend periods of time there to perform operations. As resources, they impose capacity constraints on production rates. That is, no person can be more than 100% busy performing operations. The period of time required by an operation worker to do an operation on a machine is called *handling time.* Depending on experience, different people need different handling times to do the same operation. In the wafer fab industry, operation workers are usually assigned to sector or a group of machines. They might only know how to turn the machines on or off, load or unload wafers, and press buttons to start or end operations. In a research laboratory, operation workers carry wafers around to perform operations. They are usually more knowledgeable and more actively involved in process design and product development. In the MIT IC laboratory, about 60 graduate students are involved in wafer fabrication, and most of them are in the operation worker category. Using students as workers causes scheduling complexity because of their complex personal schedules.

Support technicians are those who do not perform operations, and so they do not impose capacity constraints on production rates directly. But they do impose constraints on the availability of both of production machines and support equipment.

2.3 Activities

In this section, we discuss the activities that occur in a wafer fab factory. In terms of degree of control, we categorize activities as *controllable activities*, *uncontrollable and predictable activities*, and *uncontrollable and unpredictable activities.*

Controllable activities

This kind of activity can be arranged by a decision-maker or a manager of a wafer fab factory. Usually we can only decide when to start an activity and cannot change the time that an activity requires.

Production operations: The major activities in a IC fab are production operations, such as wafer processing, wafer inspection, and so on. These activities are well studied and accurately timed.

Set-ups: To convert a production machine from one operation to another may require cleaning the machine, changing chemistries, or performing adjustments. All such activities are called *set-up changes.*

Multiple routings: As we mentioned earlier, for every operation, there are qualified machines suitable for performing the operation. Sometimes, multiple generations of the same machine type are available for the same operation. Certain operations have their choice of the generations, while others are restricted to certain sub-classes.

Preventative maintenance: Some regular procedures must be performed to maintain both production and support machines.

Specified events for human resources: Both operation workers and support technicians are subject to some specified events. For example, new worker training, technician training, group meetings, overtime, and vacations are all activities that scheduling should be concerned about.

Environment control: We need to do some tests regularly to ensure that yield does not suffer from variability in consumables, particle counts, humidity, machine performance, and other critical parameters.

Overtime: Overtime is often used to make up for tardiness.

Uncontrollable and predictable activities

This kind of activity cannot be arranged by decision-makers or managers, but they know when they will happen and how long they will take.

Holidays: All human resources take holidays off. We must account for these events in advance.

Special time schedules: Some of the operation workers and support technicians are subject to special time schedules. For example, the maintenance technician from the manufacturer of a equipment may be available only during certain period of time.

Uncontrollable and unpredictable activities

For these activities, we do not know either when they will happen or how long they will take. The only thing we can do is to respond as quickly as possible. But we may have statistical data on uncontrollable and unpredictable activities for long term planning.

Machine failure and repair: Both production and support machines are subject to random failures and need random repair times.

Random absence of human resources: Both operation workers and support technicians are subject to random absences due to illness, accident, etc.

Defective wafers: At various points in the fab process, entire wafers are discarded, either because the wafers failed inspection or because they are broken.

Rework: Sometimes, one or more operations can be redone when a wafer fails inspection. Sometimes, rework requires extra off-process operations on the way back to the upstream buffers.

Engineering holds on lots: Sometimes, a certain process will be stopped until some experimental results are obtained or engineering decisions are made.

Demand change: The production demand is a function of the customer orders and production yield and usually varies randomly.

2.4 Constraints

Semiconductor fabrication require machines, people, wafers, time, and so on. Each of the requirements imposes constraints on the scheduling. Here we assume that circuit design and mask preparation, wafer preparation, and process design have been done before we implement fab processes.

Production machine capacity: Production machines process a certain number of wafers at a time. It takes a certain amount of time to perform an operation on a machine.

Support machine availability: Support machines do not impose capacity constraints on production rates directly. But if one support machine is down, one or more production machines may be down.

Down-time delay: For some support machines, if they go down, one or more production machines will be down after a time delay. For example, if the clean air system is down, the number of dust particles will increase in the clean room. When the number reaches a certain value, all the machines which need the laminar flow of the clean air will be prevented from working.

Operation worker capacity: Each person takes a certain amount of time to complete an operation, and each person is limited to operations that he or she knows how to perform. Workers are often trained to operate more than one piece of equipment (cross training), but each worker has limits. There are also a limited number of people available.

Support technician capacity: Each technician takes a certain amount of time to maintain or repair a machine. The support technician availability affects the repair times of both production and support machines, and therefore system capacity.

Operation sequence: In a fab process, operations have to be performed one after another following a pre-defined sequence.

Limited load size: The number of wafers a machine can process simultaneously may be variable with an upper limit. For instance, the diffusion tubes in the MIT ICL can process 100 wafers in a single operation.

Limited waiting time: At some points in a fab process, wafers cannot wait for a long time in a buffer, because exposing the wafer surface in the air will decrease yield. For example, after RCA clean, diffusion operations must be done as soon as possible without letting the wafers wait in a buffer for a long time.

Shifts: Some wafer fab factories are operated one or two shifts per day that total to less than 24 hours. Wafers must not be in certain states at the end of the final shift of the day. Therefore, some operations cannot be started late in a shift, because they will not be completed by the end of the day.

Pilot wafer runs: A pilot wafer is often run through a series of processing steps before running the whole lot. By grouping compatible lots, the scheduler can increase capacity by having lots share pilot wafer runs and setup changes.

2.5 Summary

In this section, we have discussed many of the events associated with wafer fabrication. We have attemtped to describe the issues which are important for production scheduling. The purpose has been to list and understand the events so that scheduler can be designed to account for these phenomena.

In following sections, we construct mathematical models for scheduling manufacturing systems. We focus our attention on a set of important phenomina such as machine random failure, starvation or blockage, and operation. Because of the limitation of available solution techniques, some important issues such as set-up changes and rework are not included in the mathematical models we will develop.

3 The model of a manufacturing system

In this section, we introduce a general model of a manufacturing system. In Section 3, we construct a real-time scheduler for the simplest case of this model. In later sections, we construct controllers for more and more complex systems, until we are able to control a system just as general as the one presented here.

3.1 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.

3.1.1 Clock time

It is the time measured by a clock. Define T_c to be the set of clock time, which satisfies

$$T_c = \{t \in \mathbf{R} \cup \{-\infty, +\infty\}\}. \tag{1}$$

Define C_c to be the complement of the clock time, which is an empty set:

$$C_c = \emptyset.$$

3.1.2 Working time

Working time is a subset of clock 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

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

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

$$T_w + C_w = T_c.$$

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).

3.1.3 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

$$T_{oi} = \{t \in T_w | \text{Resource i is operational}\}. \tag{3}$$

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

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

3.1.4 The relationship among the time frames

Fig. 3 illustrates the relationship among the three time frames defined above. The clock time, T_c, is the universal set, which consists of the working time, T_w, and its complement, C_w. The operational time, T_{oi}, and its complement, C_{oi}, are complementary subsets of working time.

Since we do not consider issues like holidays and overtime in this thesis, 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.

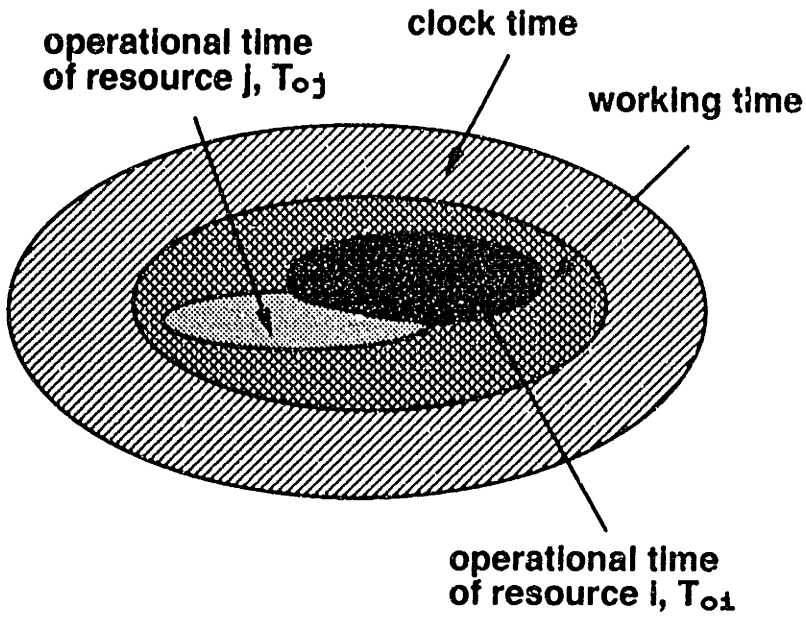

Figure 3: The relationship among the frames of time

3.2 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 [5].

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 Ryzin [36] studied the phenomenon

of delay in machines as well as in buffers.

3.3 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, machines are divided into two groups: *operation machines* and *support equipment* [5]. 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.

3.3.1 Machines

The manufacturing system under study includes a total of N machines. All machines are subject to random failures and need random repair times. For each machine, the time to fail is the time interval from a repair to next failure. The time to repair is the time interval from a failure to the instant when the machine is repaired.

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

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

Failures and repairs on different machines are assumed to be independent. Given that Machine i is operational, the probability of a failure in a small 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

$$p[\alpha_i(t + \delta t) = 1 | \alpha_i(t) = 0] = r_i \delta t,$$

$$p[\alpha_i(t+\delta t)=0|\alpha_i(t)=1]=p_i\delta t, \tag{4}$$

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

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 within the working time frame. The length of such an interval is $1/r_i + 1/p_i$.

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 event-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).

3.3.2 Buffers

There are M part types in the system. For part type k ($k = 1,\ldots,M$), a total of L_k operations are required. We assume that there are buffers between every two consecutive operations. Therefore, there are $L_k - 1$ buffers to store the Type k parts. The total number of buffers in the system is then equal to $\sum_{k=1}^{M} L_k - M$. All buffers in the system are homogeneous. That is, each buffer holds only identical parts.

Let Buffer (j,k) be the buffer between the j^{th} and $j+1^{th}(j=1,2,\ldots,L_k-1)$ operation for the Type k parts ($k=1,2,\ldots,M$). We use b_{jk} to represent the *buffer level*, i.e., the number of parts in Buffer (j,k). We represent material as continuous flow. Therefore, b_{jk} is a real number, i.e., it is not restricted to be integer-valued.

The buffer level b_{jk} 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_{jk} to be the *production rate* of the j^{th} operation of Type k parts, which is the frequency that the j^{th} operation is performed on the Type k parts.

The dynamics of the buffer level are governed by

$$\dot{b}_{jk} = u_{jk} - u_{j+1k}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M). \tag{5}$$

Define B_{jk} to be the size of Buffer (j, k). The *buffer size* B_{jk} 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 the physical buffer sizes, 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_{jk}. The buffer level b_{jk} satisfies

$$B_{jk} \geq b_{jk} \geq 0, \tag{6}$$

$$(j = 1, 2, \ldots, L_k - 1; k = 1, 2, \ldots, M).$$

Define s_{jk} to be the *empty space* in Buffer (j, k), which satisfies

$$b_{jk} + s_{jk} = B_{jk},$$

$$(j = 1, 2, \ldots, L_k - 1; \ k = 1, 2, \ldots, M).$$

Here, we would like to emphasize that the empty space s_{jk} is equally important as the buffer level b_{jk} for production control. The *empty space* s_{jk} determines how long the production can last without blocking the adjacent upstream machine when a machine is down.

3.4 Activities

An activity is a pair of events associated with a resource [15]. 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. The time to fail is the length of the time period from a repair to the next failure, which is measured in the operational time frame, T_{oi}. The time to repair is the 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 failures and repairs 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 idle (time-dependent failures in the working time frame).

Starvation and blockage: A machine is *starved* when it is idle because there are no parts in any of its upstream buffers. A machine is *blocked* when it is idle because all of its downstream buffers are 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. *A machine can be starved or blocked only when it is operational.*

An operation is blocked when the associated machine is operational and the downstream buffer is full and the adjacent downstream machine is down. An operation is starved when the associated machine is operational and the upstream is empty and different from those of machines. When an operation is starved or blocked, the associated machine may work on the other operations which require the same machine.

If a machine performs only one operation, operation starvation or blockage is the same as machine starvation or blockage. For each part type, we assume that the first operation is never starved and the last operation is never blocked. Define f_{jk}^{b} to be the *blockage fraction.* It is the fraction of operational time, T_{oi}, during which the j^{th} operation of Type k parts is blocked. Define f_{jk}^{s} to be the *starvation fraction.* It is the fraction of operational time, T_{oi}, during which the j^{th} operation of Type k parts is starved.

In general, the starvation and blockage fractions are functions of machine parameters, buffer levels, buffer sizes, and production demands.

3.5 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 and the permissible delay time between consecutive operations 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 θ_{ijk} to be the *operation index* which is 1 if Machine i performs the j^{th} operation on Type k parts, and 0 otherwise. Since we do not consider multiple route case in this paper, the operation index should satisfy

$$\sum_{i=1}^{N} \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M). \tag{7}$$

Define τ_{jk} to be the processing time of the j^{th} operation of Type k parts. The current or instantaneous capacity is then defined by

$$\sum_{\{j,k|\theta_{ijk}=1\}} \tau_{jk} u_{jk} \leq \alpha_i, \qquad (i = 1, 2, \ldots, N); \tag{8}$$

where $\tau_{jk}u_{jk}$ is the fraction of time during which Machine i performs the j^{th} operation on Type k parts if $\theta_{ijk} = 1$. The capacity constraints simply say that no machine can

be more than 100% busy to perform operations. As a machine fails or is repaired, i.e., as the machine state changes, the set of feasible instantaneous production rates changes.

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

$$\Omega(\alpha) = \{u \mid \sum_{\{j,k|\theta_{ijk}=1\}} \tau_{jk} u_{jk} \leq \alpha_i, \quad \text{for all } i \text{ and } \quad u \geq 0\}. \tag{9}$$

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.

3.6 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 (8), we have

$$\sum_{\{j,k|\theta_{ijk}=1\}} \{\frac{1}{T} \int_0^T \tau_{jk} u_{jk} dt\} \leq \frac{1}{T} \int_0^T \alpha_i dt, \quad (i = 1, 2, \ldots, N). \tag{10}$$

If the system is ergodic and in steady state, then

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T \alpha_i dt = \frac{r_i}{r_i + p_i}, \quad (i = 1, 2, \ldots, N). \tag{11}$$

Let

$$\bar{u}_{jk} = \lim_{T \to \infty} \frac{1}{T} \int_0^T u_{jk} dt. \tag{12}$$

When the system is in steady state, the long term average production rates are the

same for all operations on the same part type. Therefore,

$$\bar{u}_k = \bar{u}_{jk}, \qquad (k = 1, \ldots, M). \tag{13}$$

Plugging (13) into (10), the long term capacity is given by

$$\sum_{\{j,k|\theta_{ijk}=1\}} \tau_{jk}\bar{u}_k \leq \frac{r_i}{r_i + p_i}, \quad (i = 1, 2, \ldots, N). \tag{14}$$

The long term capacity set is

$$\bar{\Omega} = \{\bar{u}_k, \ (k = 1, \ldots, M) \ \mid \sum_{\{j,k|\theta_{ijk}=1\}} \tau_{jk}\bar{u}_k \leq \frac{r_i}{r_i + p_i}, \quad \text{for all } i \text{ and } \bar{u}_k \geq 0\}. \tag{15}$$

Define d_k to be the *production demand* for part type k 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 if and only if it is a member of the long term capacity constraint set (15).

3.7 Objectives

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

Lateness 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_{jk} to be the *production surplus* of the j^{th} operation of Type k parts, which satisfies

$$\dot{x}_{jk} = u_{jk} - d_k, \qquad (j = 1, \ldots, L_k; k = 1, \ldots, M). \tag{16}$$

Define the production surplus vector

$$x(t) = \{x_{jk}; (j = 1, \ldots, L_k; k = 1, \ldots, M)\}.$$

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 of part type k, if the surplus $x_{L_k k}$ 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. 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 $x_{L_k k}$ 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 lateness. Lateness is the time difference between the due date and the actual shipment from the system. Since we represent material as continuous flow, lateness is a continuous variable. Fig.4 illustrates the relations among demand, production, surplus, and lateness, at the final stage of a production process. Positive surplus always indicates negative lateness (actual shipment is ahead of due date). Negative surplus always indicates positive lateness (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 lateness depends on the future production (see Fig.4). This is the major reason that we choose the production surplus as the feedback variable instead of lateness.

The objective of minimizing the lateness and final product inventory is equivalent to minimizing the absolute value of the surplus of the final operation for each part type, $x_{L_k k}$ $(k = 1, 2, \ldots, M)$. 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

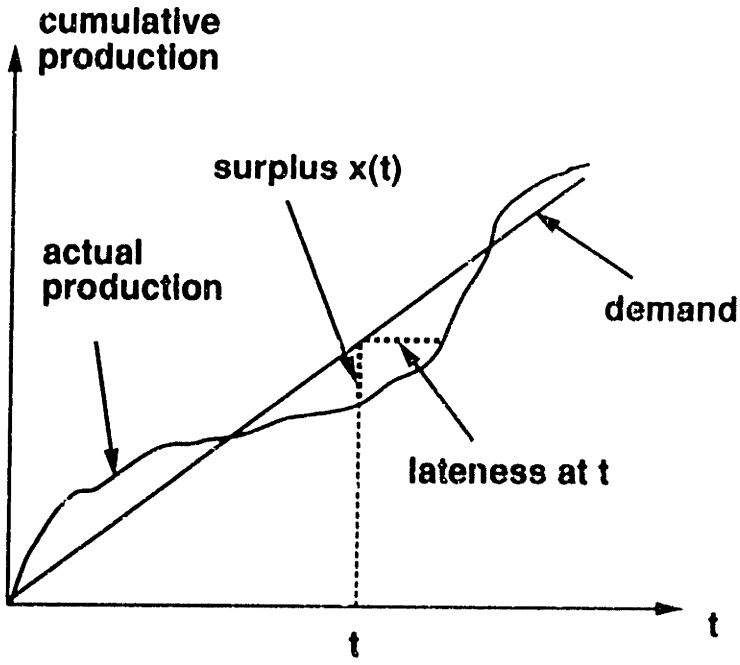

Figure 4: Demand, production, surplus, and lateness

work-in-process inventory will increase starvation and blockage, and therefore reduce production rates.

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 the production control as 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.

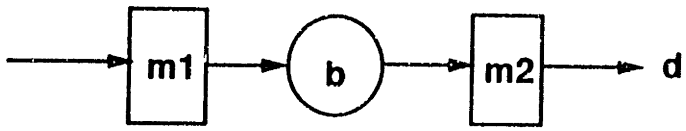

Figure 5: Two-machine, one-part-type system

4 Two machine, one part type systems

In this section, we study the simplest case, a two-machine, one-part-type system. The results in this section are extended to more complex and realistic systems in the subsequent sections.

As illustrated in Fig. 5, 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. 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.

4.1 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

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \tag{17}$$

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 and buffer constraints are

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

The function $g(x, b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive or too negative. It satisfies

$$\lim_{\|x\| \to \infty} g(x, b) = \infty,$$

$$\lim_{\|b\| \to \infty} g(x, b) = \infty,$$

and $g(0, 0) = 0$. The constraints are specified in the form of $u \in \Omega(\alpha)$, where $\Omega(\alpha)$ is given by (9). Assume that the initial buffer level $b(t_0)$ satisfies

$$b(t_0) = x_1(t_0) - x_2(t_0). \tag{18}$$

The buffer level, b, is a function of the surplus x, which can be determined by (5), (16), and (18),

$$b(t) = x_1(t) - x_2(t). \tag{19}$$

Therefore, by plugging (19) and (18) into (17), the value function J is not an explicit function of the buffer level $b(t)$.

There is no a technique available to solve this dynamic optimization problem analytically. A numerical solution was obtained by Van Ryzin [36] 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 to more complicated systems.

4.2 Feedback control law

If the optimal value function $J(x, \alpha, t_0)$ is known, the optimal production flow rate u can be determined by solving the linear programming problem [see Appendix A]:

$$\min_{u}\{\frac{\partial J}{\partial x_1}u_1 + \frac{\partial J}{\partial x_2}u_2\} \tag{20}$$

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 the linear program (20), we can make the following observations.

Observation 1: The right hand side of the inequalities in the constraint set are random parameters. Therefore, the shape and size of the capacity constraint set change randomly with the machine status $\alpha(t)$.

Observation 2: The linear program (20) represents a *real-time feedback control law* since the LP is determined by x and α. When the production surplus x and machine state α are fed back from the shop-floor, a new production rate, u, is generated by solving the linear program.

Observation 3: The objective function is linear in the production rate u. The constraint set is a convex polyhedron of u. Therefore, the optimal production policy $u(t)$ takes on values at the extreme points of the constraint set. The coefficients of u in the objective are functions of production surplus x. Therefore, for each machine state α, an optimal production policy divides the x-space into a set of regions in which the production rate is constant. Each region corresponds to an extreme point of the constraint set. However, the regions do not cover the whole x-space. If the

gradient of J does not exist, or is zero, an unique optimal solution may not exist. In the following text, we assume that the gradient of J exists. This assumption is not actually restrictive since we can replace the gradient with the subgradient of J. The subgradient always exists since the value function J is continuous and convex [35].

In the case that both machines are operational ($\alpha_1 = 1$ and $\alpha_2 = 1$), Fig. 7 illustrates the regions in x-space. The two straight lines are the *zero-buffer* and *full-buffer* boundaries since the buffer level is a linear function of x. The feasible region of x-space lies between the zero buffer boundary ($b = 0$) and the full buffer boundary ($b = B$). The other two curves are the *coefficient boundaries*, which are the sets of points in which one of the coefficients of the objective function in (20) is zero ($\partial J/\partial x_i = 0$). The feasible area of the x-space is divided into four mutually exclusive regions which correspond to the four extreme points of the constraint set. In each region, the production rate u is constant. The intersection of the coefficient boundaries is called the *hedging point*, which is the desirable operating state of the system. The feedback controller (20) always attempts to drive the system to the hedging point, and to keep it at there. Intuitively, this is because that the hedging point is a global minimum of the value function J. There might be more than one hedging point if the value function J is not strictly convex. But all hedging points must form a convex set since J is convex. In the subsequent sections, we only consider the cases which have an unique hedging point.

Observation 4: Both the coefficient boundaries are *attractive* (See [16] for a similar problem.) That is, when the system state reaches a boundary, it crosses the boundary back and forth while it moves along on a boundary towards the hedging point. This phenomena is referred to as *chattering* on a boundary which severely affects the efficiency of the production control algorithm. To avoid chattering, a set of *conditional constraints* is used to guide the system to move to the hedging point when production surplus x reaches a coefficient boundary. The conditional constraints are stated in Section 4.5. Consequently, the production rates are constant on boundaries in x-space.

Observation 5: Since the production rates are constant in every region as well as on every boundary in x-space, the linear program (20) yields a piecewise constant solution, $u(t)$. Therefore, the production rates do not have to be calculated at every

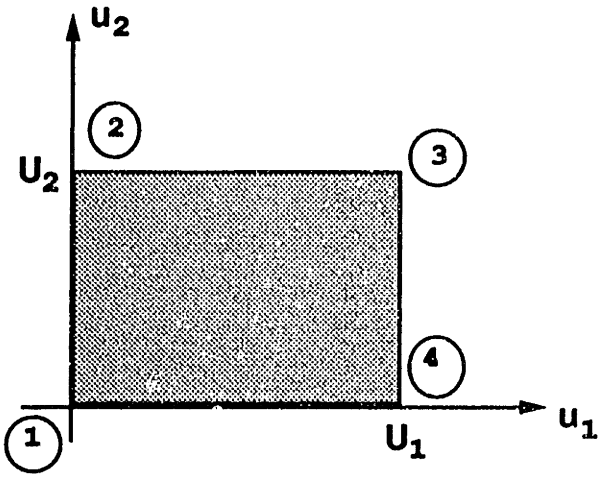

Figure 6: The capacity set when both machines are operational

time instant. They need only to be computed when machine state α changes or when production surplus $x(t)$ reaches a boundary.

We have discussed some properties of the optimal production policy by inspecting the necessary optimality condition which is in the form of linear programming formulation. However, we do not know the optimal shape and position of the coefficient boundaries in x-space. In the following subsections, we construct approximations for the boundary shape and position and buffer size such that the system behavior and performance are satisfactory.

4.3 System behavior specification

In this section, we specify a list of requirements regarding system behavior, which serves as the guide for the construction of approximations in the subsequent subsections. The desirable system behavior 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.

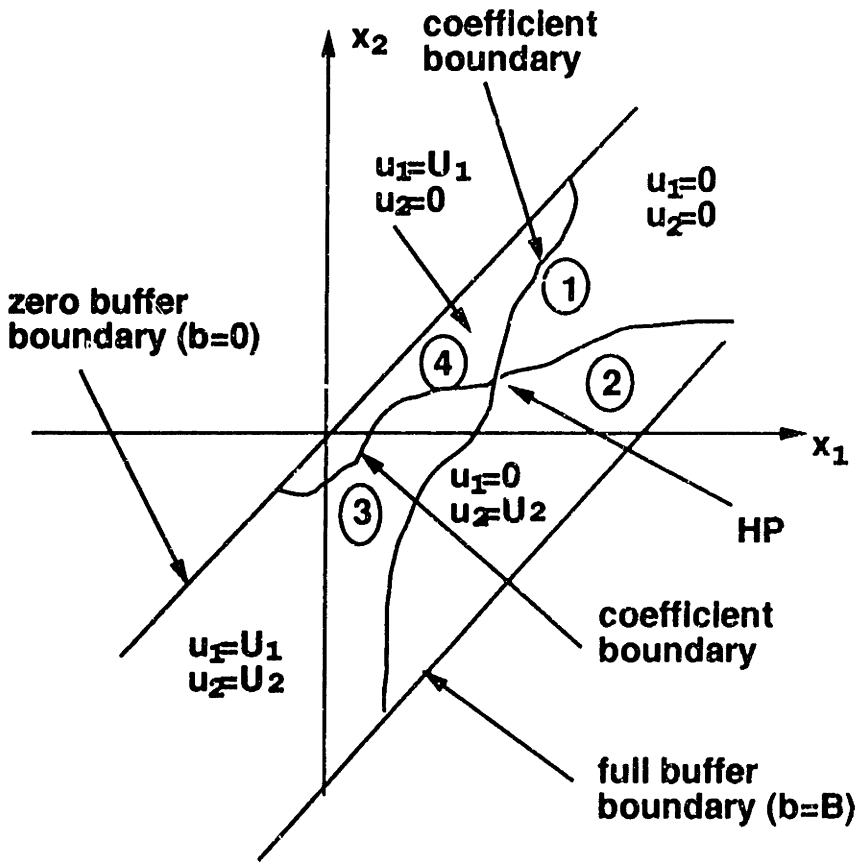

Figure 7: The linear program solution in x-space

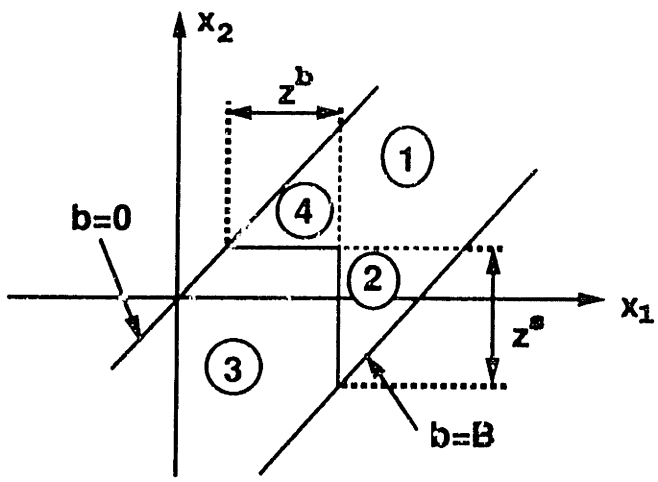

Figure 8: The desirable boundary shape in x-space

The behavior requirements are the considerations of spatial decomposition. For example, for a large factory, when a single machine fails, we do not want to change the production plan of the entire factory 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 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 (17). They are heuristic approximations. In latter sections, we will see that these specifications reduce the complexity of the production control problem.

4.4 The desirable boundary shape in x-space

From the discussion in the previous sections, we see that the production controller (20) is characterized by the shape and position of the coefficient and the buffer-size boundaries in x-space. In this section, we determine the proper boundary shape such that the system behaves as specified in Section 4.3. Consequently, we construct an approximate, $\tilde{J}$, of the optimal J function. For simplicity, we assume that the hedging

point does not change with machine state α. Suppose that the system has reached the hedging point and both machines are operational. The production rate decision is then $u_1 = u_2 = d$, so the system stays at the hedging point indefinitely. Then consider the following situation. Suppose that at time t after the system reaches the hedging point, Machine 1 fails ($\alpha_1 = 0$) and Machine 2 is still operational ($\alpha_2 = 1$). According to the behavior requirement (a) of Section 4.3 and the capacity constraints (9), the production decision should be $u_1 = 0$ and $u_2 = d$ 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 ($x_1 = x_2$). Before the system reaches the zero buffer boundary, the coefficient of u_2 remains to be zero ($\partial \check{J}/\partial x_2 = 0$). From the observation above, we conclude that the coefficient boundary defined by ($\partial \check{J}/\partial x_2 = 0$) must be straight and horizontal, and must go through the hedging point.

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

The desirable boundary shape in x-space is illustrated in Fig.8. 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.

4.5 The conditional constraints

To avoid chattering on the coefficient boundaries and to respect the buffer constraints, we need to impose a set of conditional constraints to the linear program.

Define (z_1, z_2) to be the *hedging point* which is the desirable value of (x_1, x_2), and is assumed to be independent of α. The components of the hedging point are unknown and will be determined in Section 4.7.4. The conditional constraints are

$$\begin{aligned}
&\text{if } x_1 = z_1, \quad B > b > 0, \quad \text{and} \quad \alpha_1 = 1, \quad \text{then } u_1 = d;\\
&\text{if } x_2 = z_2, \quad B > b > 0, \quad \text{and} \quad \alpha_2 = 1, \quad \text{then } u_2 = d;\\
&\text{if } b = 0, \qquad\qquad\qquad\qquad\qquad\quad \text{then } u_1 \geq u_2;\\
&\text{if } b = B, \qquad\qquad\qquad\qquad\qquad\quad \text{then } u_1 \leq u_2.
\end{aligned}$$

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. 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.

4.6 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 (20) becomes

$$\min_u\{a_1(x,\alpha,t)(x_1-z_1)u_1 + a_2(x,\alpha,t)(x_2-z_2)u_2\} \tag{21}$$

subject to:

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

$$\begin{aligned} &\text{if } x_1 = z_1, \quad B > b > 0, \quad \text{and } \alpha_1 = 1, \quad \text{then } u_1 = d; \\ &\text{if } x_2 = z_2, \quad B > b > 0, \quad \text{and } \alpha_2 = 1, \quad \text{then } u_2 = d; \\ &\text{if } b = 0, \quad \text{then } u_1 \geq u_2; \\ &\text{if } b = B, \quad \text{then } u_1 \leq u_2; \end{aligned}$$

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 $a_i(x,\alpha,t)$ is a positive function over the feasible region in x-space. Different forms of $a_i(x,\alpha,t)$ correspond to different value functions. Comparing (21) and (20), we have the approximate gradient

$$\frac{\partial \tilde{J}}{\partial x_1} = a_1(x,\alpha,t)(x_1 - z_1);$$

$$\frac{\partial \tilde{J}}{\partial x_2} = a_2(x, \alpha, t)(x_2 - z_2).$$

The vertical boundary in the x-space corresponds to $\partial \tilde{J}/\partial x_1 = 0$ (or $x_1 = z_1$). The horizontal boundary corresponds to $\partial \tilde{J}/\partial x_2 = 0$ (or $x_2 = z_2$). By inspecting the feedback controller (21), the choice of $a_i(x, \alpha, t)$ does not affect the production control policy $u(t)$ in this case. That means that all convex functions which give us the same vertical and horizontal boundaries in x-space are equally good choices of value functions for this scenario. For simplicity, we choose $a_1(x, \alpha, t) = a_2(x, \alpha, t) = 1$. This choice of a_i corresponds to a family of quadratic J functions whose level set, $s_\beta = \{x \in \mathbf{R}^2 \mid J(x, \alpha) \leq \beta\}$, is an ellipse centered at the hedging point (z_1, z_2).

Note that the coefficient of u_1 is not a function of x_2. Therefore, if the buffer is neither empty nor full, (21) 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.

When the system reaches the hedging point, the scheduler generates the same policy as in a KANBAN system. That is, the production rates are equal to the demand. Moreover, if the system drifts away from the hedging point due to random events such as machine failures or starvation or blockage, new policies are generated such that the system recovers as soon as possible.

In the linear program, there are three parameters we need to determine. They are the components of the hedging point (z_1, z_2) and the buffer size B. In the following subsections, we formulate approximations for those unknowns.

4.7 Control parameter estimation

In this section, we estimate the unknown parameters of the feedback control linear program (21). In doing so, we develop a frequency-duration method to formulate an approximate relationship among starvation, blockage, buffer hedging level, buffer hedging space, machine parameters, and demand. A pair of starvation and blockage constraints are also formulated. A nonlinear optimization problem is set up to determine the buffer size. The hedging point is determined such that the average final product inventory is minimized.

4.7.1 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. 8). It is the buffer level when the system reaches the hedging point, which satisfies

$$z^b = z_1 - z_2. \tag{22}$$

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

$$z^s = B - z^b. \tag{23}$$

Fig.9 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 also 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.

Let f_i^b and f_i^s be the blockage and starvation fraction of i^{th} operation at Machine i ($i = 1, 2$), which are defined in Section 2.4. In the following, we estimate the starvation and blockage fractions in terms of buffer hedging level and space, machine parameters, and demand. Since we are considering the time average, we treat both machines as if they have deterministic failure-repair cycles with length $(1/r_i + 1/p_i)$. But the starting time of the cycles are random.

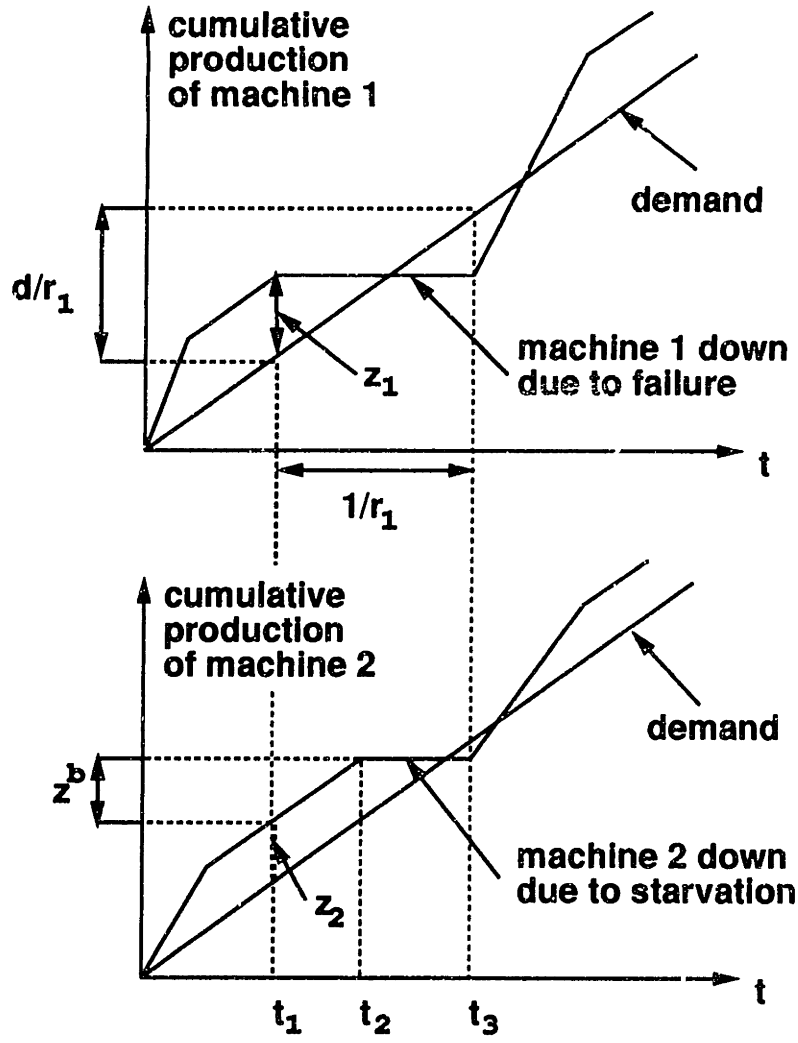

Figure 9: A sample trajectory of the cumulative production

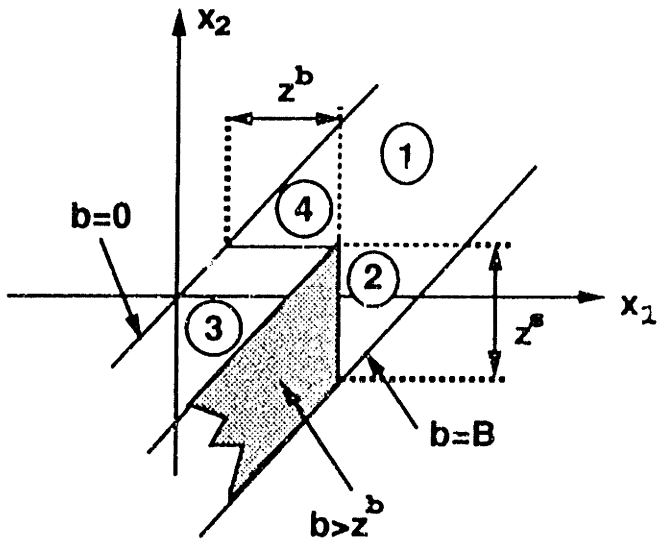

Figure 10: The region in x-space where $b \geq z^b$

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

$$f_1^s = 0. \tag{24}$$

The starvation fraction of Machine 2: Assume that the demand is a member of the long term capacity set $\bar{\Omega}$ (15). 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. Therefore, at the instant that Machine 1 goes down, the system state x is very likely to be in the shaded area of Fig.10, which is the feasible area of the x-space that satisfies

$$b \geq z^b.$$

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.11).

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 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

$$\frac{1}{p_2}(1 - f_2^s)\bar{u}_2 = (\frac{1}{r_2} + \frac{1}{p_2})d; \tag{25}$$

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 (25) 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

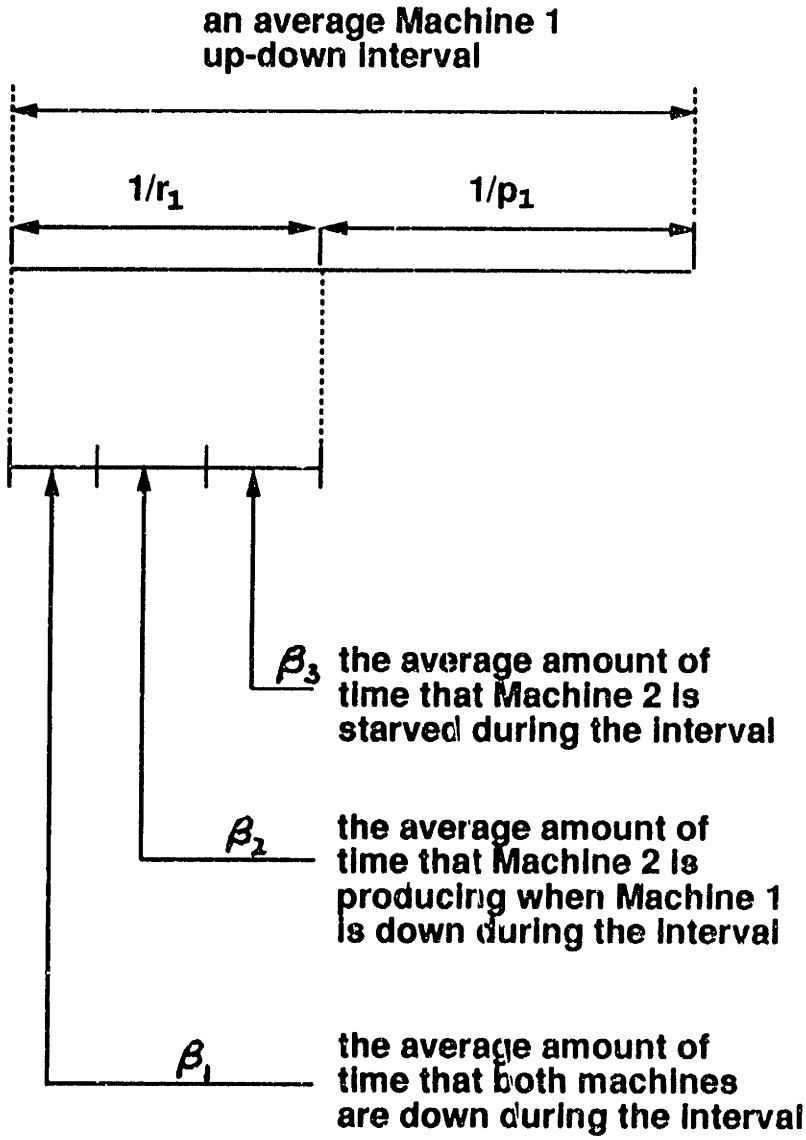

Figure 11: The average cycle time of Machine 1 breakdown

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

$$\beta_1 + \beta_2 + \beta_3 = \frac{1}{r_1}. \tag{26}$$

After manipulation, this leads to

$$\frac{1}{d}z^b + \frac{r_1+p_1}{r_1p_1}f_2^s - \frac{1}{d}z^b f_2^s = \frac{1}{r_1}, \tag{27}$$

or

$$f_2^s = \frac{1}{\frac{1}{r_1}+\frac{1}{p_1}-\frac{z^b}{d}}(\frac{1}{r_1}-\frac{z^b}{d}). \tag{28}$$

Equation (27) 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.12 depicts the relationship described by Equation (27) in the space of decision

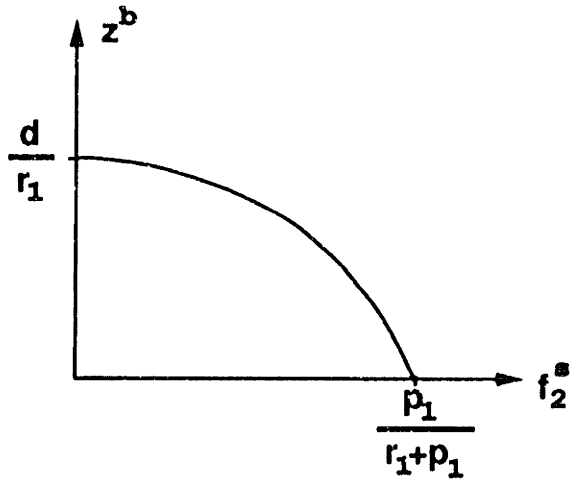

Figure 12: The relationship among z^b, f_2^s, d, r_1, and p_1

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 (27) is bounded from above by d/r_1. This differs from the real situation because of the approximation of not considering the variances of the failure and repair time.

Observation 3: The buffer hedging level z^b is a concave 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

$$\frac{1}{d}z^s + \frac{r_2 + p_2}{r_2 p_2} f_1^b - \frac{1}{d} z^s f_1^b = \frac{1}{r_2}; \tag{29}$$

or

$$f_1^b = \frac{1}{\frac{1}{r_2} + \frac{1}{p_2} - \frac{z^s}{d}} \left(\frac{1}{r_2} - \frac{z^s}{d}\right). \tag{30}$$

Equation (29) describes the relationship among the blockage fraction of Machine 1, the buffer hedging space, the demand, and machine parameters.

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

$$f_2^b = 0. \tag{31}$$

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

$$\frac{\frac{1}{p_1}(1 - f_1^b)}{\frac{1}{r_1} + \frac{1}{p_1}} U_1 \geq d, \tag{32}$$

$$\frac{\frac{1}{p_2}(1-f_2^s)}{\frac{1}{r_2}+\frac{1}{p_2}}U_2 \geq d, \tag{33}$$

where U_i is the maximum service rate of Machine i. In this case, $U_i = 1/\tau_i$ (i=1,2). We assume that the demand d is a member of the constraint set (15). Conditions (32) and (33) ensure that both machines have enough capacity to achieve the demand given the blockage and starvation fractions. Let D_i be the *isolated capacity* of Machine i which is given by

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

After a rearrangement, the starvation and blockage constraints, (32) and (33), become

$$f_1^b \leq 1 - \frac{d}{D_1}, \tag{34}$$

$$f_2^s \leq 1 - \frac{d}{D_2}. \tag{35}$$

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

$$0 \leq d \leq \min\{D_1, D_2\}.$$

4.7.2 The buffer hedging level and space

In the previous subsection, 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 minimize both buffer hedging level and space.

As we discussed in Section 2, 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 (27), (29), (34), and (35) together as follows

$$\min\{z^b + z^s\} \tag{36}$$

subject to:

$$\frac{1}{d}z^b + \frac{r_1+p_1}{r_1p_1}f_2^s - \frac{1}{d}z^b f_2^s = \frac{1}{r_1};$$

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

$$f_1^b \leq 1 - \frac{d}{D_1};$$

$$f_2^s \leq 1 - \frac{d}{D_2};$$

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

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

4.7.3 The buffer size and average buffer level

Solving (36) is equivalent to minimizing both the buffer size and average buffer level. The buffer size is defined in Section 2.2.2 as

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

The average buffer level is different from the buffer hedging level. Let $\bar{b}$ be the average buffer level, which can be obtained by taking time average of (19),

$$\bar{b} = \bar{x}_1 - \bar{x}_2. \tag{37}$$

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 the next subsection.

4.7.4 The hedging point and surplus loss

In the preceding subsection, we determined the buffer size needed in the feedback controller (21). In this section, we are going to determine the hedging point.

Since both machines are unreliable and can be starved or blocked, there is a

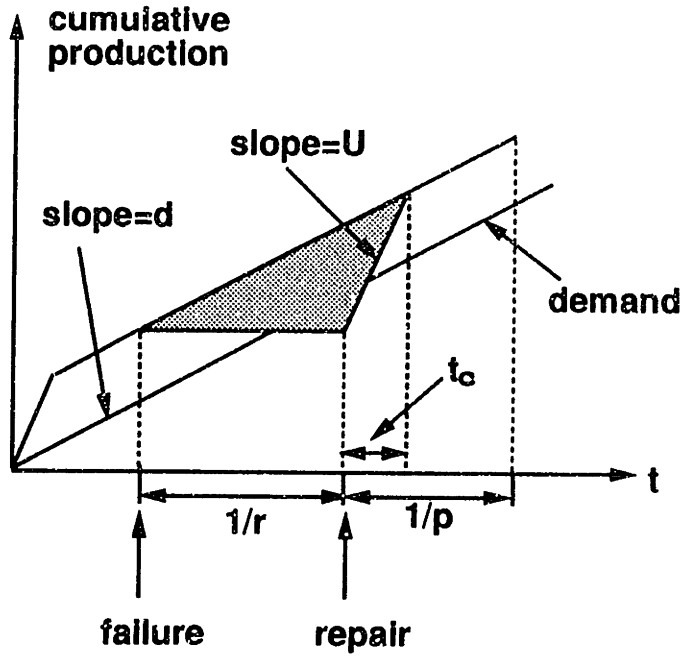

Figure 13: The surplus loss due to failure

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

$$z_i = \bar{x}_i + \Delta_i, \quad (i = 1, 2); \tag{38}$$

where Δ_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.13 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

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

in which U_i is the maximum service rate of Machine i and in this case is equal to $1/\tau_i$.

Let δ_i^r be the average surplus loss due to failures for Machine i. Dividing the shaded area of Fig.13 by the average time that Machine i is up once and down once, $(1/r_i + 1/p_i)$, leads to

$$\begin{aligned} \delta_i^r &= \frac{\frac{1}{2}(\frac{1}{r_i})^2 d + t_{ci}(\frac{d}{r_i}) + \frac{1}{2}t_{ci}^2 d - \frac{1}{2}t_{ci}^2 U_i}{\frac{1}{r_i} + \frac{1}{p_i}} \\ &= \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{1}{r_i})^2, \quad (i = 1, 2). \end{aligned} \tag{39}$$

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 (39). Then

$$\delta_i^s = \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{f_i^s}{p_i})^2, \quad (i = 1, 2). \tag{40}$$

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

$$\delta_i^b = \frac{r_i p_i}{r_i + p_i}\frac{d}{2}(\frac{U_i}{U_i - d})(\frac{f_i^b}{p_i})^2, \quad (i = 1, 2). \tag{41}$$

Therefore the average surplus loss is approximately given by

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

In order to deliver the products on time, we would like to minimize the absolute value of surplus x_2. Therefore, we choose the hedging point (z_1, z_2) such that

$$\bar{x}_2 = 0. \tag{43}$$

From (22), (38), and (43), the hedging point should satisfy

$$\begin{aligned} z_2 &= \Delta_2; \\ z_1 &= z^b + \Delta_2. \end{aligned} \tag{44}$$

Up to this point, we have constructed the real-time scheduler for the two-machine, one-part-type system. In Section (4.6), 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. The buffer size is determined in Section 4.7.3. The components of the hedging point are determined above in this section.

4.8 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 nonlinear program (36). 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 (44).

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 (21).

Step 5: The loading times for each machine are determined by a heuristic policy such as the *staircase strategy* [2]. 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 [23]. 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.14 illustrates the hierarchical structure.

4.9 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 describe 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.

4.9.1 Buffer size vs demand and machine parameters

The nonlinear program (36) is a well behaved problem. For the top level calculation in the algorithm, we used a commercially available software package [8].

As we mentioned earlier, the buffer size is function of the demand and machine parameters. Fig.15 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 size. The bigger the demand is, the bigger the buffer size is.

Fig.16 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 size is.

Fig.17 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

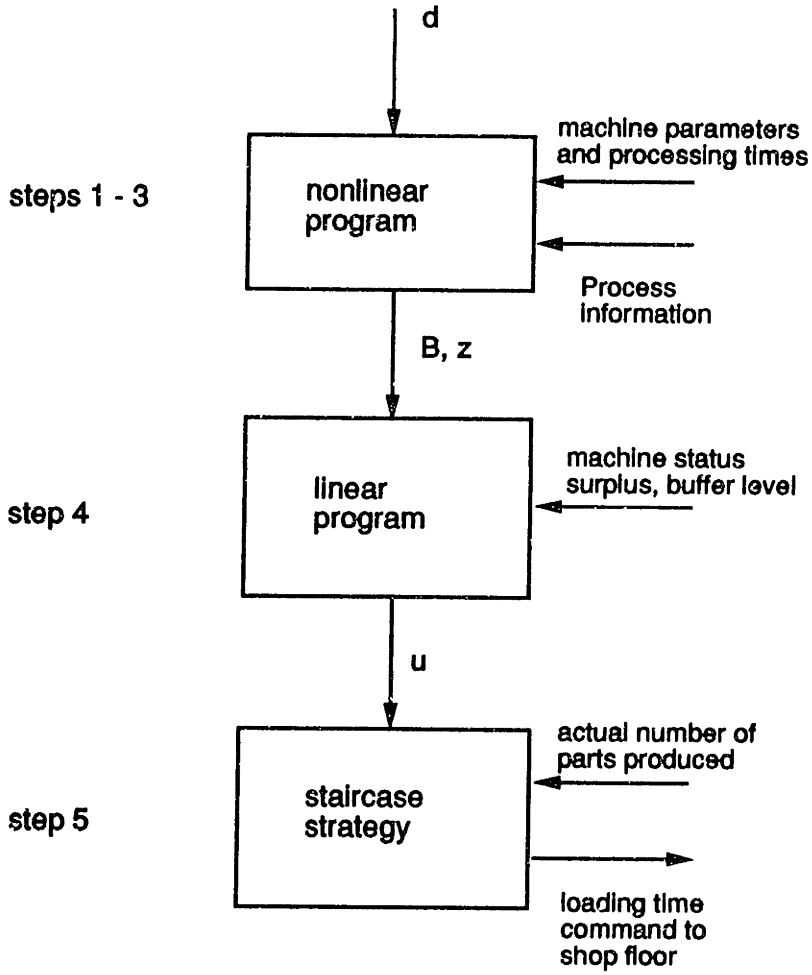

Figure 14: The hierarchical policy

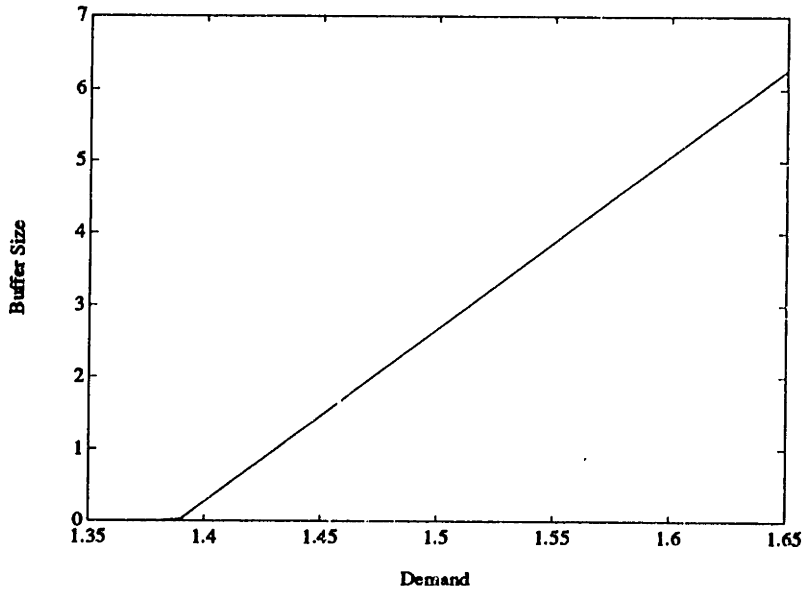

Figure 15: Buffer size vs demand

the buffer size is.

4.9.2 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 and p is 1/day. The unit of τ is a day. The unit of parts is a lot.

Given that the demand is equal to 1.6 (lots/day), the buffer size and hedging point are listed in Table.2. 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 [12]. Fig.18 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.19 illustrates buffer level vs time. The buffer level consists of the parts in the

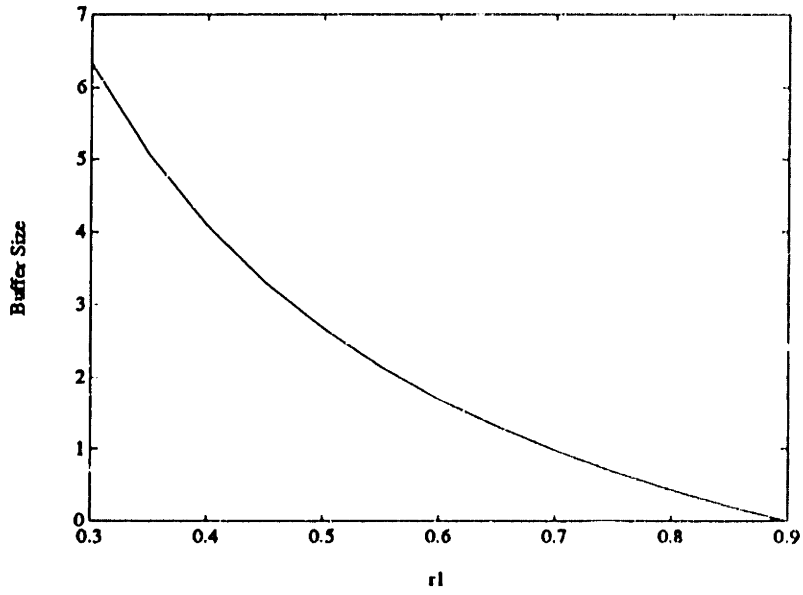

Figure 16: Buffer size vs r_1

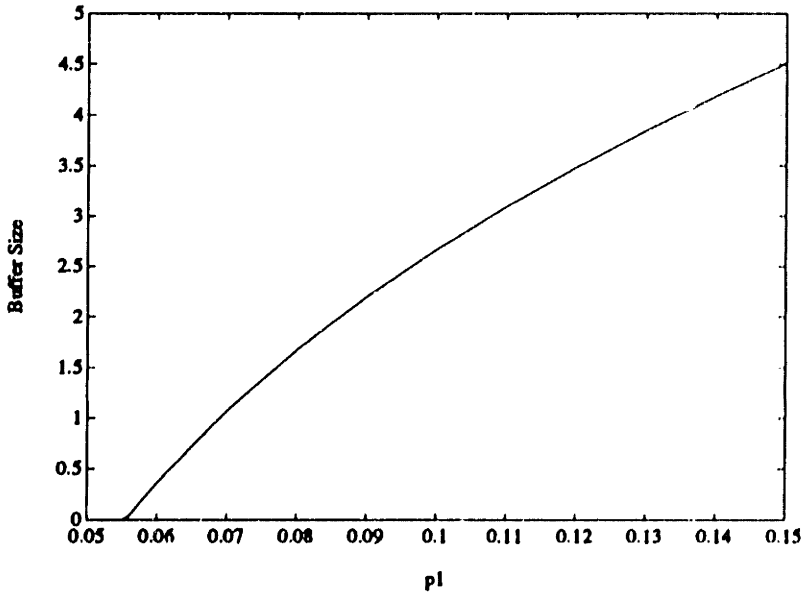

Figure 17: Buffer size vs p_1

i	f_i^a	f_i^b	z_i^a	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 2: The buffer size and hedging point for the two machine and one part type example

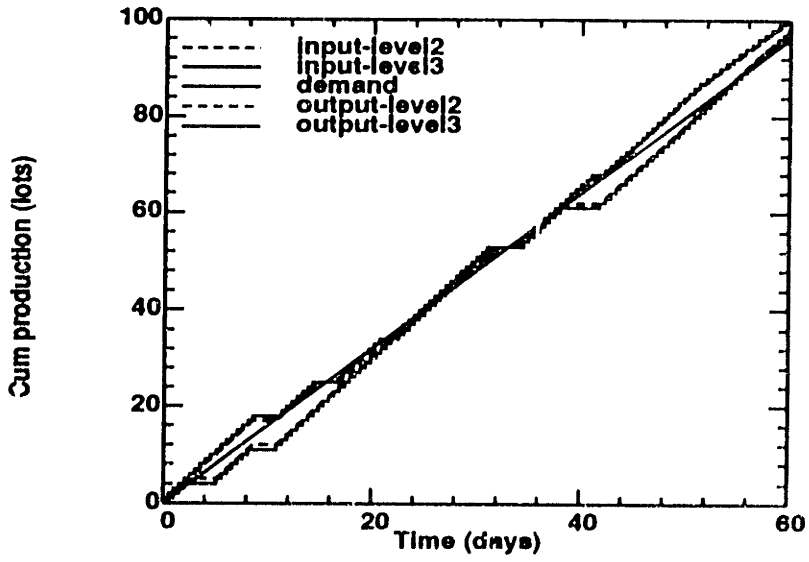

Figure 18: The simulation result of the cumulative production of a two-machine and one-part-type system

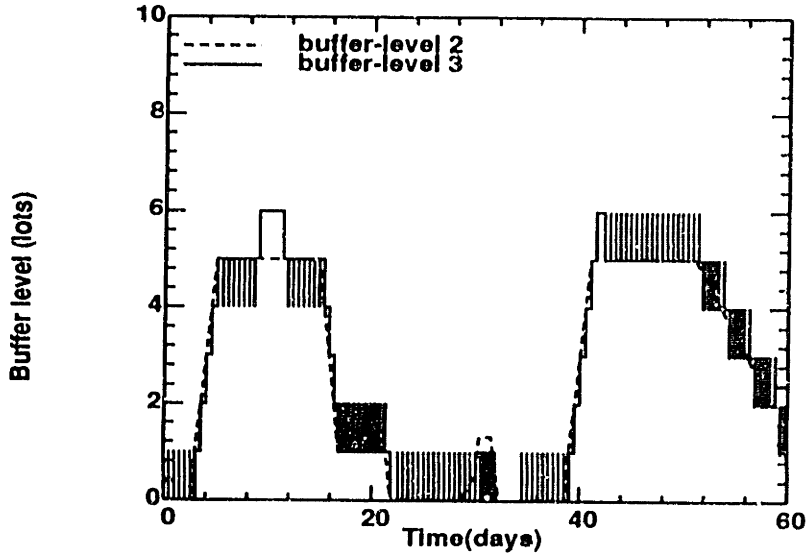

Figure 19: Buffer level as a function of time t

buffer and the working piece in Machine 1. The dashed line is the middle level result. 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.

4.10 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.

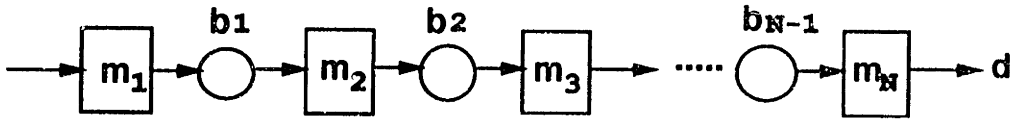

Figure 20: N-machine, one-part-type system

5 N-machine, one-part-type systems

In this section, we study the N-machine, one-part-type systems. As illustrated in Fig.20, the system consists of N machines and $N-1$ buffers. For Machine i ($i = 1, 2, \ldots, N$), the failure rate is p_i and the repair rate is 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, Buffer 1, Machine 2, Buffer 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.

5.1 Dynamic optimization

Given the system as described above, the production control is formulated as a dynamic programming problem,

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \tag{45}$$

subject to:

$$\tau_i u_i \leq \alpha_i, \quad (i = 1, 2, \ldots, N);$$
$$u_i \geq 0, \qquad (i = 1, 2, \ldots, N);$$

where the system dynamics and buffer constraints are

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

where B_i is the buffer size which is to be determined. The constraints are in form of (6) and (9). 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 $b_i(t_0)$ 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 (16), we have

$$b_i(t) = x_i(t) - x_{i+1}(t), \qquad (i = 1, 2, \ldots, N-1). \tag{46}$$

IF (46) is plugged into the penalty function $g(x, b)$, the value function J is not an explicit function of buffer level $b(t)$. As we did in the previous case, the optimal value function J is assumed to be differentiable with respect to x and t in this section.

5.2 Feedback control law

If the optimal value function $J(x, \alpha, t)$ is known, a necessary optimality condition for the production control policy u is following linear programming problem [See Appendix A]:

$$\min_u \{\sum_{i=1}^{N} \frac{\partial J}{\partial x_i} u_i\} \tag{47}$$

subject to:

$$\tau_i u_i \leq \alpha_i, \quad (i = 1, 2, \ldots, N);$$
$$u_i \geq 0, \qquad (i = 1, 2, \ldots, N);$$

where

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

As we observed in Section 3.2, the coefficient boundaries defined by $\frac{\partial J}{\partial x_i} = 0$ divide the x-space into mutually exclusive regions. The scheduler drives the system towards the hedging point. For different position of the hedging point and boundary shape, the system behaves differently.

Define $z = (z_1, z_2, \ldots, z_N)$ to be the hedging point which is again assumed to be independent of α. In the feedback control linear program, we do not know the optimal value function J and buffer sizes B_i $(i = 1, 2, \ldots, N-1)$. In subsequent subsections, we specify a list of system behavior requirements and extend the frequency-duration method of Section 4 to formulate approximations such that the system behaves as specified.

5.3 System behavior specifications

In a way similar to the classical control system design procedure, we specify the system behavior requirements in this section. Then, the unknown function and parameters are approximated for the feedback controller according to the specifications. The following are specifications on system behavior:

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

The behavior requirements impose maximum independence among machines for a given WIP distribution. This consideration is essential for system decomposition as well as for the implementation of the production control policy. Consequently we are

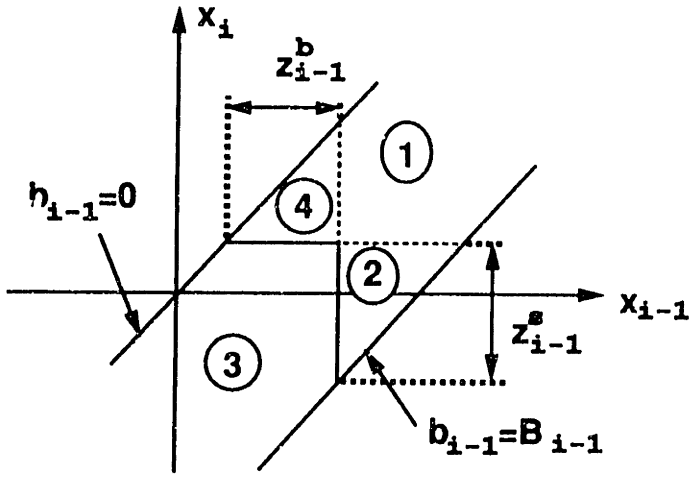

Figure 21: The desirable boundary shape in (x_{i-1},x_i) space

able to make decisions locally to some extent.

5.4 The boundary shape in x-space

To ensure the system behaves as specified, we need to determine the gradient of the value function in the controller. It is extremely difficult to relate the value function to system behavior directly. But we found that it is possible to determine the boundary shape in x-space according to the behavior specifications of previous section.

Let $\tilde{J}$ be an approximate of the optimal value function J, which yields satisfactory system behavior. It is assumed to be differentiable with respect to x and t. Suppose that the system has reached the hedging point, $x_i = z_i$ $(i = 1,2,\ldots,N)$. The production rate decision is $u_i = d$ $(i = 1,2,\ldots,N)$, so the system stays at the hedging point indefinitely. Then consider the following case. Suppose that at time t after the system has reached the hedging point, all machines but Machine k are down ($\alpha_k = 1, \alpha_i = 0$ for $i \neq k$). According to the behavior requirements in the preceding subsection and the capacity constraints to the feedback controller (47), the production decision should be

$$u_k = d, \tag{48}$$

and

$$u_i = 0, \quad (i \neq k); \tag{49}$$

until Machine k is either starved or blocked. Before the starvation or blockage occurs, x_k is equal to a constant z_k and the other production surpluses, $x_i (i \neq k)$, decrease at rate d, according to the system dynamics. This leads to the fact that the coefficient boundary defined by $\partial \tilde{J}/\partial x_k = 0$ must be perpendicular to the x_k axis in x-space. Otherwise, on the boundary, we have

$$x_k \neq \text{constant}$$

which contradicts Eq. (48). From the discussion above, in order to satisfy the system behavior requirements of Section 4.3, the coefficient boundaries in x-space must have the following properties:

(a) The coefficient boundary, a hyperplane defined by $\partial \tilde{J}/\partial x_i = 0$, must be perpendicular to the x_i axis in x-space. (b) All coefficient boundaries intersect each other at the hedging point.

Fig. 21 illustrates the boundary shape in (x_{i-1}, x_i) subspace. Suppose that both Machine $i-1$ and Machine i are in the middle of the production line. Since both Machine $i-1$ and Machine i can be either starved or blocked, the decision making rules and more complex than those in two-machine, one-part-type case. For example, suppose both Machine i and $i-1$ are operational and the system state is in region (3) of the (x_{i-1}, x_i) subspace (therefore, Buffer i is neither empty nor full). The decision rules depend on the adjacent machine and buffer states. There are four possible situations. If Buffer $i-2$ is not empty and Buffer i is not full, then the production rates should be $u_{i-1} = (1/\tau_{i-1})$ and $u_i = (1/\tau_i)$. If Buffer $i-2$ is empty and Buffer i is not full, we choose $u_{i-1} = \min(u_{i-2}, 1/\tau_{i-1})$ and $u_i = (1/\tau_i)$. If Buffer $i-2$ is not empty and Buffer i is full, we have $u_{i-1} = (1/\tau_{i-1})$ and $u_i = \min(1/\tau_i, u_{i+1})$. If Buffer $i-2$ is empty and Buffer i is full, the production policy is then $u_{i-1} = \min(u_{i-2}, 1/\tau_{i-1})$ and $u_i = \min(1/\tau_i, u_{i+1})$.

Regions (1), (2), and (4) in the (x_{i-1}, x_i) subspace (Fig.18) are the transient states. After the system reaches the hedging point, it will never go into those regions again. When the system leaves the hedging point because of machine failures, production surpluses decrease. When Machine i is operational, x_i can only increase if $x_i < z_i$.

5.5 The conditional constraints

To avoid chattering, we need a set of conditional constraints to guide the system moving along the boundaries in x-space. The conditional constraints are

$$\begin{array}{lll} \text{if } x_i = z_i, & & \\ B_i > b_i > 0, \text{ and } \alpha_i = 1, & \text{then} & u_i = d, \quad (i = 1, 2, \ldots, N); \\ \text{if } b_i = 0, & \text{then} & u_i \geq u_{i+1}, \quad (i = 1, 2, \ldots, N-1); \\ \text{if } b_i = B_i, & \text{then} & u_i \leq u_{i+1}, \quad (i = 1, 2, \ldots, N-1); \end{array} \tag{50}$$

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 [16]. 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.

5.6 The linear program for real-time control

To ensure that the coefficient boundaries in x-space are perpendicular to axes, and go through the hedging point, the linear program (47) becomes

$$\min_u \{\sum_{i=1}^{N} a_i(x, \alpha, t)(x_i - z_i)u_i\} \tag{51}$$

subject to:

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

where

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

in which $a_i(x, \alpha, t)$ is a positive function within the feasible region in x-space. By inspecting the feedback controller (47), the choice of $a_i(x, \alpha, t)$ does not affect the production control policy $u(t)$ in this case. For simplicity, we choose $a_i(x, \alpha, t) = 1$ for all i.

On the coefficient boundary defined by $\partial J/\partial x_i = 0$ the production surplus x_i is equal to a constant z_i. If Buffer $i-1$ is not empty and Buffer i is not full, the production control policy at Machine i is independent of the states of the other machines and buffers. coupling occurs only when the connected buffers are either empty or full as specified in the conditional constraints. In the linear program, the hedging point $(z_1, \ldots, z_N)$ and the buffer sizes, B_i $(i = 1, \ldots, N-1)$, are still unknown. We show how they may be found in the next subsection.

5.7 Control parameter estimation

In the preceding subsection we formulated a real-time production scheduler as a linear program. To estimate the unknown parameters of the controller, namely, the hedging point and buffer sizes, we extend the frequency-duration method of Section 4.7 to the N-machine, one-part-type case.

5.7.1 Starvation and blockage

Define z_i^b to be the *hedging level* of Buffer i (see Fig. 21). It is the number of parts in Buffer i when the system reaches the hedging point, which satisfies

$$z_i^b = z_i - z_{i+1}, \quad (i = 1, 2, \ldots, N-1). \tag{52}$$

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

$$z_i^s = B_i - z_i^b, \quad (i = 1, 2, \ldots, N-1). \tag{53}$$

To show the effects of machine failures in the middle of a production line, Fig.22 illustrates a sample cumulative production trajectory for a three-machine tandem system. Suppose that the system has reached the hedging point point before time t_1. Then, Machine 2 goes down at time t_1 and the other two machines are still operational. Machine 1 continues to produce and fills up the empty space of Buffer 1. Buffer 1 becomes full at time t_2 and Machine 1 is blocked until Machine 2 is repaired at time t_4. Meanwhile, Machine 3 keeps producing after Machine 2 goes down and consumes the material in Buffer 2. Buffer 2 becomes empty at time t_3 and Machine 3 is starved until t_4. In this example, a failure of Machine 2 causes starvation at Machine 3 as well as blockage at Machine 1. In this case, the length of the period of blockage $[t_2, t_4]$ at Machine 1 is a function of the demand, the empty space of Buffer 1, and the time to repair Machine 2. The length of the period of starvation $[t_3, t_4]$ at Machine 3 is a function of the demand, the buffer level of Buffer 2, and the time to repair Machine 2. For longer production lines, the starvation and blockage configuration could be a little more complicated than those in this scenario. In general, the starvation and blockage fractions are functions of buffer levels and empty spaces, machine parameter, and demand.

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

$$f_1^s = 0. \tag{54}$$

The starvation fraction of Machine i (i=2,...,N): We assume that the demand d is a member of the long term capacity set (15). 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

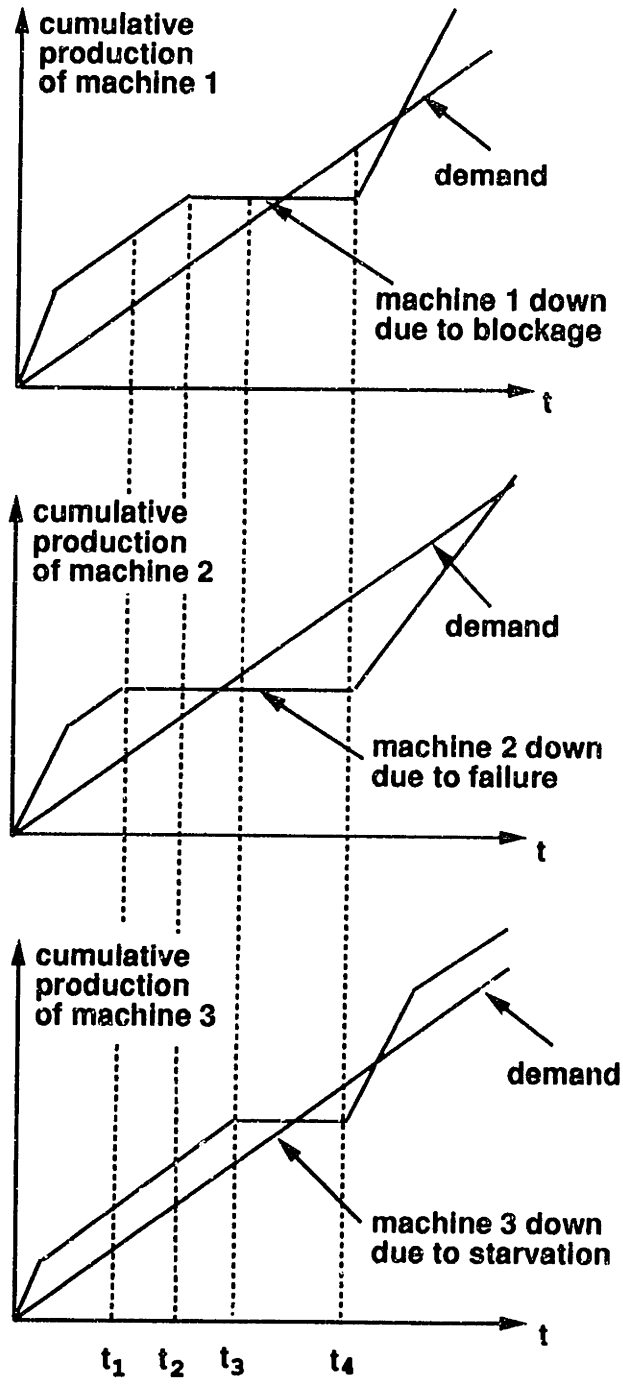

Figure 22: A sample trajectory of the cumulative productions

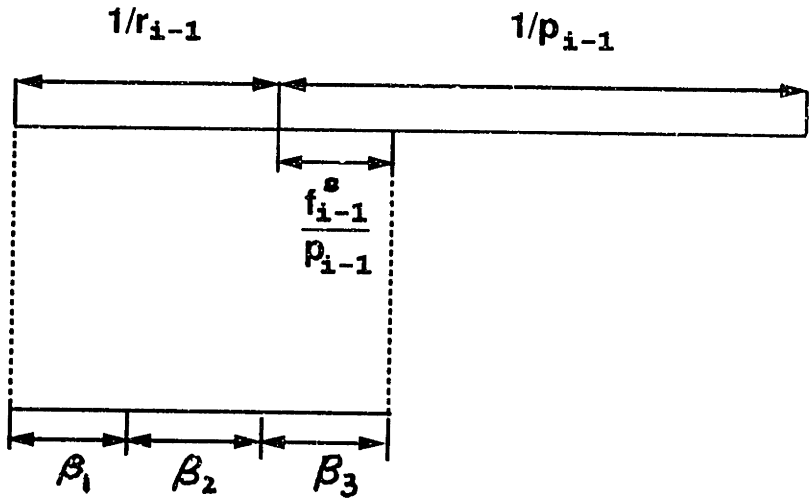

Figure 23: The average cycle time of Machine i-1 breakdown

Fig. 23).

Let β_1 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

$$\frac{\frac{1}{r_{i-1}} + \frac{f^s_{i-1}}{p_{i-1}}}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}}.$$

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

$$\frac{\frac{1}{r_i} + \frac{f^b_i}{p_i}}{\frac{1}{r_i} + \frac{1}{p_i}}.$$

Consequently,

$$\beta_1 = (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}) \left(\frac{\frac{1}{r_{i-1}} + \frac{f^s_{i-1}}{p_{i-1}}}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}}} \right) \left(\frac{\frac{1}{r_i} + \frac{f^b_i}{p_i}}{\frac{1}{r_i} + \frac{1}{p_i}} \right)$$

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

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

is determined by

$$\frac{1}{p_i}(1 - f_i^s - f_i^b)\bar{u}_i = (\frac{1}{r_i} + \frac{1}{p_i})d, \tag{55}$$

or

$$\bar{u}_i = \frac{(r_i + p_i)d}{r_i(1 - f_i^s - f_i^b)},$$

where $(1/p_i)(1 - f_i^s - f_i^b)$ is the amount of time that Machine i is producing during an average Machine i up-down interval. Equation (55) says that the cumulative production at Machine i equals the cumulative demand during an interval of length $(1/r_i + 1/p_i)$. Let β_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_{i-1}^b at the instant that Machine $i-1$ goes down, the average amount of time that production at Machine i can last, approximately, is

$$\beta_2 = \frac{z_{i-1}^b}{\bar{u}_i}.$$

Let β_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

$$(\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i p_i}{r_i + p_i})\frac{1}{p_i} f_i^s. \tag{56}$$

Since Machine i cannot be starved when Machine $i-1$ is up, β_3 is the same as (56):

$$\begin{aligned}\beta_3 &= (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i p_i}{r_i + p_i})\frac{1}{p_i} f_i^s \\ &= f_i^s(\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}})(\frac{r_i}{r_i + p_i}).\end{aligned}$$

The β's satisfy

$$\beta_1 + \beta_2 + \beta_3 = \frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} f_{i-1}^s. \tag{57}$$

Plugging the β's into (57), and manipulating, leads to

$$\frac{1}{d} z_{i-1}^b - \frac{1}{p_{i-1}} f_{i-1}^s + \frac{r_{i-1} + p_{i-1}}{r_{i-1} p_{i-1}} f_i^s + \frac{1}{r_{i-1}} f_i^b - \frac{1}{d} z_{i-1}^b f_i^s - \frac{1}{d} z_{i-1}^b f_i^b + \frac{1}{p_{i-1}} f_{i-1}^s f_i^b = \frac{1}{r_{i-1}},$$

or

$$f_i^s = \frac{1 - f_i^b}{\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} - \frac{z_{i-1}^b}{d}} (\frac{1}{r_{i-1}} + \frac{1}{p_{i-1}} f_{i-1}^s - \frac{z_{i-1}^b}{d}) \quad (i = 2, \ldots, N). \tag{58}$$

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

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

or

$$f_i^b = \frac{1 - f_i^s}{\frac{1}{r_{i+1}} + \frac{1}{p_{i+1}} - \frac{z_i^s}{d}} (\frac{1}{r_{i+1}} + \frac{1}{p_{i+1}} f_{i+1}^b - \frac{z_i^s}{d}), \quad (i = 1, 2, \ldots, N-1). \tag{59}$$

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

$$f_N^b = 0. \tag{60}$$

To ensure that the system has enough capacity to achieve the demand, the star-

vation and blockage fractions must satisfy

$$f_i^b + f_i^s \leq 1 - \frac{d}{D_i}, \quad (i = 1, 2, \ldots, N), \tag{61}$$

where

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

and U_i [illegible] service rate of Machine i. In this case, $U_i = 1/\tau_i$ $(i = 1, 2, \ldots, N)$.

5.7.2 The buffer hedging levels and spaces

By putting (54), (58), (59), (60), and (61) together, we form an optimization problem to minimize the sum of the buffer hedging levels and spaces.

$$\min\{z_1^b + \ldots + z_{N-1}^b + z_1^s + \ldots + z_{N-1}^s\} \tag{62}$$

subject to:

$$\frac{1}{d} z_{i-1}^b - \frac{1}{p_{i-1}} f_{i-1}^s + \frac{r_{i-1} + p_{i-1}}{r_{i-1} p_{i-1}} f_i^s + \frac{1}{r_{i-1}} f_i^b - \frac{1}{d} z_{i-1}^b f_i^s - \frac{1}{d} z_{i-1}^b f_i^b + \frac{1}{p_{i-1}} f_{i-1}^s f_i^b = \frac{1}{r_{i-1}},$$

$$(i = 2, \ldots, N);$$

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

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

$$f_1^s = 0, \quad f_n^b = 0;$$

$$f_i^s + f_i^b \leq 1 - \tfrac{d}{D_i}, \quad (i = 1, 2, \ldots, N);$$

$$f_i^s \geq 0, \quad f_i^b \geq 0, \quad (i = 1, 2, \ldots, N);$$

$$z_i^b \geq 0, \quad z_i^s \geq 0, \quad (i = 1, 2, \ldots, N-1).$$

5.7.3 The buffer sizes and average buffer levels

By definition, the buffer sizes are given by

$$B_i = z_i^b + z_i^s, \quad (i = 1, 2, \ldots, N-1). \tag{63}$$

As we discussed in Section 4.7.3, the average buffer levels are

$$\bar{b}_i = z_i^b + (\Delta_{i+1} - \Delta_i), \quad (i = 1, 2, \ldots, N-1), \tag{64}$$

where Δ_i is the surplus loss at Machine i, which is approximately given by (see Section 4.7.4)

$$\Delta_i = \frac{r_i p_i}{r_i + p_i} \frac{d}{2} \left(\frac{U_i}{U_i - d}\right) \left\{ \left(\frac{1}{r_i}\right)^2 + \left(\frac{f_i^s}{p_i}\right)^2 + \left(\frac{f_i^b}{p_i}\right)^2 \right\}, \quad (i = 1, 2, \ldots, N). \tag{65}$$

5.7.4 The hedging point

The relation between the hedging point $(z_1, \ldots, z_N)$ and the average surplus $(\bar{x}_1, \ldots, \bar{x}_N)$ is governed by

$$z_i = \bar{x}_i + \Delta_i, \quad (i = 1, 2, \ldots, N). \tag{66}$$

In order to reduce the final product inventory, we would like to minimize the absolute value of surplus x_N. A plausible criterion for the selection of the hedging point $(z_1, z_2, \ldots, z_N)$ is such that

$$\bar{x}_N = 0. \tag{67}$$

From (52), (66), and (67), the hedging point must satisfy

$$\begin{aligned} z_N &= \Delta_N; \\ z_i &= \sum_{k=i}^{N-1} z_k^b + \Delta_N, \quad (i = 1, 2, \ldots, N-1). \end{aligned} \tag{68}$$

5.8 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 (15).

Step 2: Calculate the buffer hedging levels, z_i^b $(i = 1, 2, \ldots, N)$, and hedging spaces, z_i^s $(i = 1, 2, \ldots, N)$, and the starvation and blockage fractions for each machine by

solving the nonlinear program (62). 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 (68).

Step 4: Using the feed-back information of surplus x_i and machine state α_i $(i = 1, 2, \ldots, N)$, calculate the production rates, u_i $(i = 1, 2, \ldots, N)$, in real time by solving the linear program (51).

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.

5.9 Example

For the simulation, we still use the three-level hierarchical policy described in Section 4.8. At the top level, the buffer sizes and hedging point are calculated by using a commercially available software package [8]. HIERCSIM [12] is used for the next two level simulation. The system consists of five machines and four buffers. The parameters are chosen as follows:

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

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

Given that the demand is $d = 0.7$ lots/day, the buffer sizes and hedging point are calculated at the top level by solving (62) and (68), and listed in Table 3.

Fig.24 illustrates the simulation results of the cumulative production. The straight

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

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 (51).

Fig. 25 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.

To see the effects of buffer levels and sizes, we increase the demand to 0.85 lots/day without changing the buffer sizes and the hedging point (Table 3). The simulation result in Fig. 26 shows that the production fell behind the demand. That is, with the buffer levels and sizes in Table 3, the system is starved or blocked too much to achieve the demand, 0.85.

Given that the demand is 0.85 lots/day, the desirable buffer sizes and hedging point are calculated and listed in Table 4. With the appropriate buffer sizes and hedging point (Table 4), the actual production follows the demand closely (see Fig. 27).

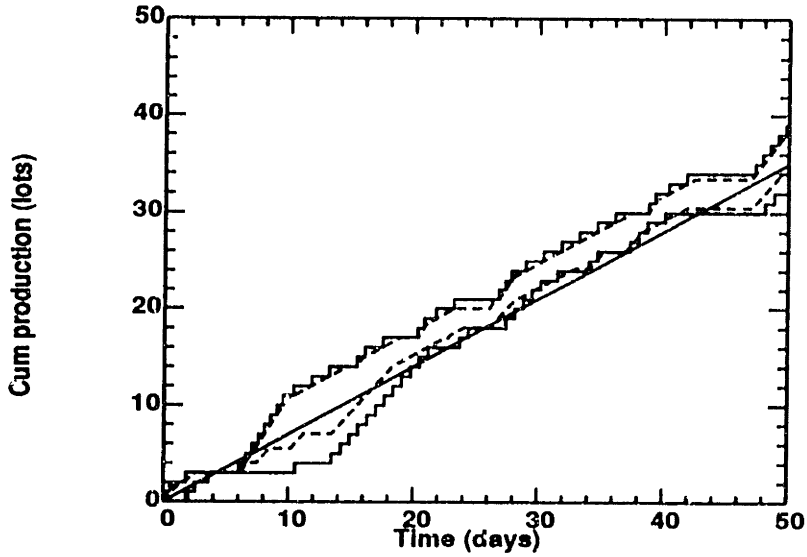

Figure 24: The simulation result of cumulative production of the five-machine and one-part-type system

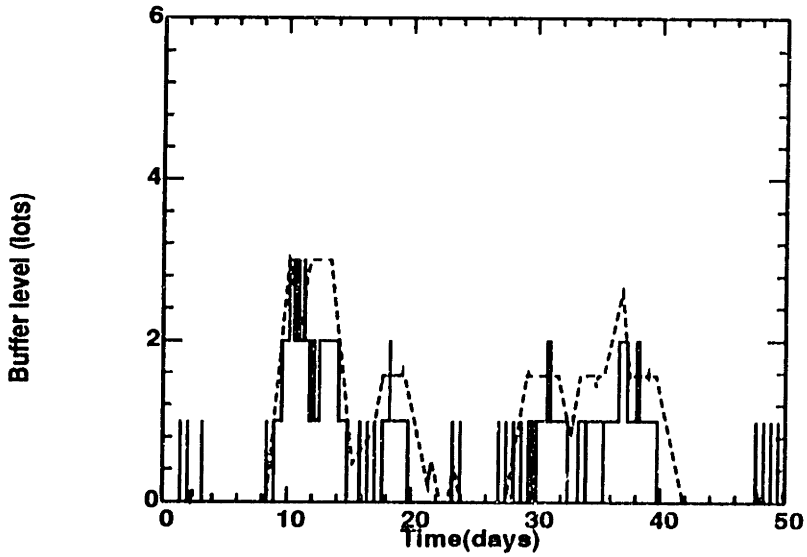

Figure 25: Buffer level as a function of time t

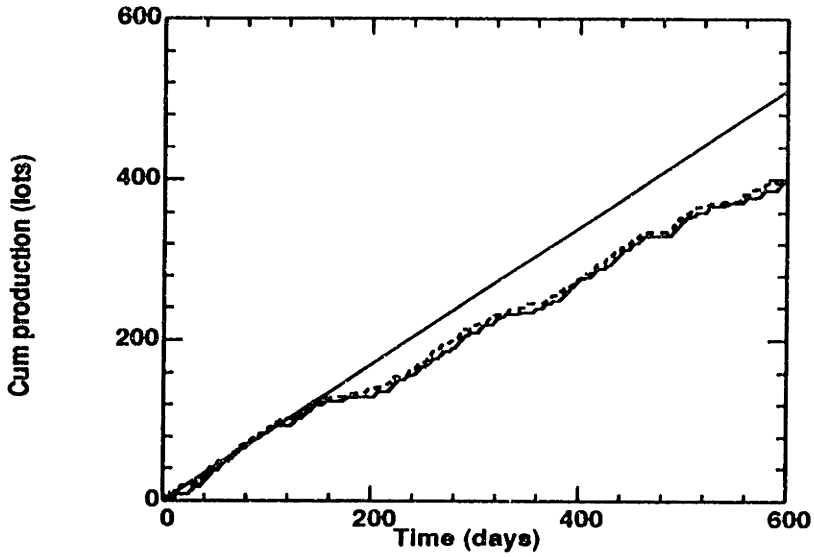

Figure 26: The effects of infeasible buffer levels and sizes

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

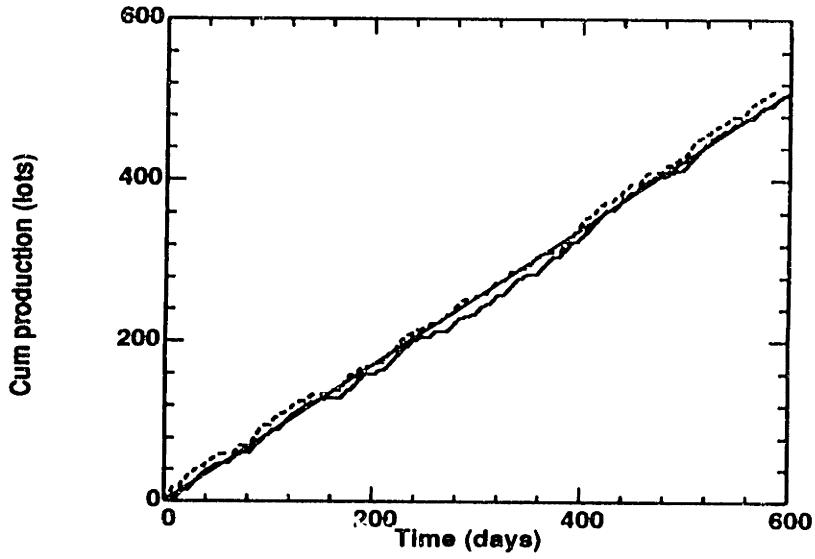

Figure 27: The effects of desirable buffer levels and sizes

6 Two-machine, two-part-type systems

In this section we study two-machine, two-part-type systems. As illustrated in Fig.28, the system consists of two machines and two buffers. For Machine i ($i = 1, 2$), the failure rate is p_i and the repair rate is r_i. Two part types are produced. Each Type j ($j = 1, 2$) part needs an operation with processing time τ_{1j} on Machine 1 and an operation with time τ_{2j} on Machine 2. Homogeneous buffers are located between the two machines. The demand for Type j parts is d_j (j=1,2). We assume that Machine 1 is never starved and Machine 2 is never blocked.

6.1 Dynamic optimization

Determining production flow control policy can be formulated as a dynamic optimization problem:

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \tag{69}$$

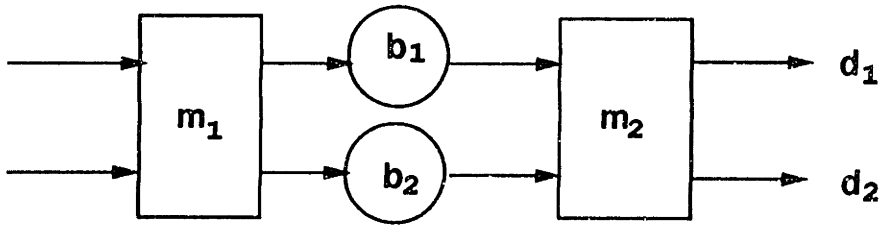

Figure 28: Two-machine, two-part-type system

subject to:

$$\tau_{11}u_{11} + \tau_{12}u_{12} \leq \alpha_1;$$
$$\tau_{21}u_{21} + \tau_{22}u_{22} \leq \alpha_2;$$

$$u_{11} \geq 0, \quad u_{12} \geq 0;$$
$$u_{21} \geq 0, \quad u_{22} \geq 0;$$

where the system dynamics and the buffer constraints are

$$\dot{x}_{ij} = u_{ij} - d_j, \qquad (i = 1,2; j = 1,2);$$
$$B_j \geq b_j \geq 0, \qquad (j = 1,2);$$

in which B_j $(j = 1,2)$ are the buffer sizes to be determined. The constraints are specified in the form of $u \in \Omega(\alpha)$, where $\Omega(\alpha)$ is given by (9). The function $g(x,b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive and too negative. Assume that the initial buffer levels satisfy

$$b_j(t_0) = x_{1j}(t_0) - x_{2j}(t_0), \quad (j = 1,2). \tag{70}$$

The buffer levels are functions of the surpluses x, which are determined by (5), (16), and (70):

$$b_j(t) = x_{1j}(t) - x_{2j}(t), \quad (j = 1,2). \tag{71}$$

Therefore, by plugging (70) and (71) into (69), the cost-to-go J is not an explicit function of the buffer levels.

6.2 Feedback control law

The optimal production rate u is determined, if the optimal value function $J(x, \alpha, t)$ is known, by solving the following linear programming problem [See Appendix A],

$$\min_u \{\frac{\partial J}{\partial x_{11}} u_{11} + \frac{\partial J}{\partial x_{12}} u_{12} + \frac{\partial J}{\partial x_{21}} u_{21} + \frac{\partial J}{\partial x_{22}} u_{22}\} \tag{72}$$

subject to:

$$\tau_{11} u_{11} + \tau_{12} u_{12} \leq \alpha_1;$$
$$\tau_{21} u_{21} + \tau_{22} u_{22} \leq \alpha_2;$$

$$u_{11} \geq 0, \quad u_{12} \geq 0;$$
$$u_{21} \geq 0, \quad u_{22} \geq 0;$$

$$\dot{x}_{ij} = u_{ij} - d_j, \qquad (i = 1, 2; j = 1, 2);$$
$$B_j \geq b_j \geq 0, \qquad (j = 1, 2).$$

The linear program (72) represents a feedback controller. When the machine state α and production surplus x are fed back from the production site, the production rate $u(t)$ is calculated by solving the LP. The objective function is linear in u. The constraint set is a convex polyhedron of the production rate. For each machine state α, the feedback controller divides the x-space into mutually exclusive regions. Each region corresponds to an extreme point of the constraint set.

Within each region of the x-space, the production rate $u(t)$ is a constant. Therefore, we do not have to calculate the production rates at every time instant. They need only be computed when α changes or when $x(t)$ reaches a boundary. The intersection of the coefficient boundaries defined by $\partial J / \partial x_{ij} = 0$ is called the *hedging point*, which is the desirable operating state of the system. The feedback controller always attempts to drive the system to the hedging point, and to keep it there. The hedging point is usually positive because when a failure occurs, the production surplus tends to decrease. A positive hedging point often keeps the average value of x closer to zero than if the hedging point is at the origin.

In the linear program, we do not know the boundary shape or position or the buffer sizes. They are approximated in the subsequent subsections.

6.3 System behavior specificatio

In order to construct approximate boundary shape in x-space, we specify desirable system behavior requirements as follows:

(1) When Machine 1 fails, keep Machine 2 producing without changing production plan until one of the buffers is empty.
(2) When Machine 2 fails, keep Machine 1 producing without changing production plan until one of the buffers is full.

The behavior requirements are essential for spatial decomposition and the implementation of the production control policy. Consequently, we are able to divide a manufacturing system into several smaller subsystems by putting buffers among them.

6.4 The boundary shape in x-space

In this section, we determine the boundary shape in x-space such that the system behaves as specified in the preceding subsection. Consequently, we construct an approximate, $\tilde{J}$, of the optimal value function J.

Suppose that the system has reached the hedging point. The production decision is then $u_{ij} = d_j$ $(i = 1, 2; j = 1, 2)$, so the system stays at the hedging point indefinitely. Let us consider that at time t after the system has reached the hedging point, Machine 1 fails and Machine 2 is still operational. Then, according to the behavior requirements in Section 6.3 and the capacity constraints of the linear program (72), the production decision should be

$$\begin{aligned} u_{1j} &= 0, \qquad (j = 1, 2); \\ u_{2j} &= d_j, \qquad (j = 1, 2); \end{aligned} \tag{73}$$

until one of the buffers is empty or Machine 1 is repaired. Before the starvation or repair occurs, Machine 2 consumes the material in the buffers. The production

surpluses x_{1j} $(j = 1, 2)$ decrease at rates d_j $(j = 1, 2)$ and x_{2j} $(j = 1, 2)$ are constant, according to the system dynamics. The system state moves along the boundary in x-space defined by $\partial \tilde{J}/\partial x_{2j} = 0$, $(j = 1, 2)$. If one of the buffers, say Buffer 1, becomes empty before Machine 1 is repaired, the production rates become $u_{11} = u_{12} = u_{21} = 0$ and $u_{22} = d_2$ until Buffer 2 becomes empty or Machine 1 is repaired. From then on and before the starvation or repair occurs, the production surpluses x_{11} and x_{21} decrease at rate d_1 and x_{12} decreases at rate d_2, and x_{22} is a constant. The system state moves along the boundary in x-space defined by $\partial \tilde{J}/\partial x_{22} = 0$ towards the zero buffer boundary defined by $b_2 = 0$. If we enumerate all possible configurations, it is possible to show that the coefficient boundary defined by $\partial \tilde{J}/\partial x_{ij} = 0$ is a hyperplane and perpendicular to the axis x_{ij} in x-space.

Since Type 1 parts share the machines with Type 2 parts, how fast we can produce Type 1 parts at a machine depends on the surplus state and production decision for Type 2 parts. The two part types are coupled together during the decision making procedure.

6.5 The conditional constraints

Because of the singularity of (72) on the coefficient boundaries and the buffer constraints ($B_j \geq b_j \geq 0$), a set of *conditional constraints* is imposed to the controller to avoid chattering on the boundaries in x-space.

Define $(z_{11}, z_{12}, z_{21}, z_{22})$ to be the hedging point which is assumed to be independent of α. The components of the hedging point are unknown and will be determined in Section 6.7.

The conditional constraints are

$$\begin{aligned}
&\text{if } x_{ij} = z_{ij}, \quad B_j > b_j > 0, \quad \text{and} \quad \alpha_i = 1, \quad \text{then} \quad u_{ij} = d_j; \\
&\text{if } b_j = 0, \quad \text{then} \quad u_{1j} \geq u_{2j}; \\
&\text{if } b_j = B_j, \quad \text{then} \quad u_{1j} \leq u_{2j}; \\
&\qquad\qquad\qquad\qquad\quad (i = 1, 2; j = 1, 2).
\end{aligned} \tag{74}$$

which indicate that when a production surplus reaches the hedging point, the corresponding production rate should be equal to the demand. When Buffer j is empty, the downstream machine cannot produce Type j parts faster than the upstream machine. When Buffer j is full, the upstream machine cannot produce Type j parts

faster than the downstream machine. In the control theory literature, the boundaries where $x = z$ are *singular arcs.* $B \geq b$ and $b \geq 0$ are *state variable inequality constraints.*

6.6 The feedback control linear program

To ensure that the coefficient boundaries in x-space are perpendicular to axes and go through the hedging point, the linear program (72) becomes

$$\min_{u} \sum_{i=1}^{2} \sum_{j=1}^{2} a_{ij}(x_{ij} - z_{ij})u_{ij} \tag{75}$$

subject to:

$$\tau_{11}u_{11} + \tau_{12}u_{12} \leq \alpha_1;$$
$$\tau_{21}u_{21} + \tau_{22}u_{22} \leq \alpha_2;$$

$$u_{11} \geq 0, \quad u_{12} \geq 0;$$
$$u_{21} \geq 0, \quad u_{22} \geq 0;$$

if $x_{ij} = z_{ij}$, $B_j > b_j > 0$, and $\alpha_i = 1$, then $u_{ij} = d_j$;
if $b_j = 0$, then $u_{1j} \geq u_{2j}$;
if $b_j = B_j$, then $u_{1j} \leq u_{2j}$;
$(i = 1, 2; j = 1, 2)$.

In the linear program, $a_{ij}(x, \alpha, t)$ is a positive function in the feasible region of the x-space. For simplicity, we choose $a_{ij} = 1$, $(i = 1, 2; j = 1, 2)$. The hedging point and the buffer sizes are still unknown. We show how they may be found in the next subsection.

6.7 Control parameter estimation

In this section, we estimate the unknown parameters of the feedback controller (75). We would like to extend the results of single-part-type systems in Section 4 to the two-part-type systems. In doing so, we separate the original system into two single-part-type approximate linear systems.

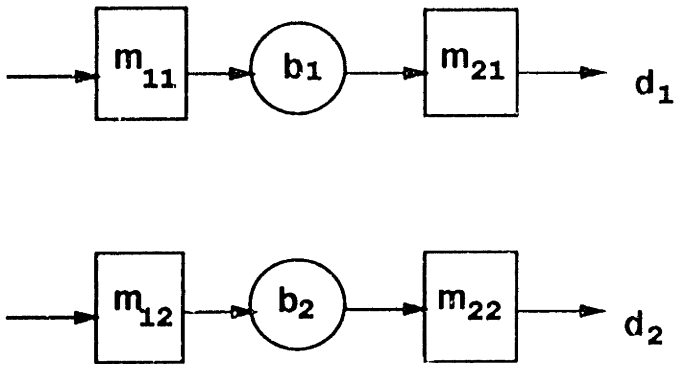

Figure 29: The approximate linear systems

6.7.1 Approximate linear systems and capacity allocation

To complete the description of the feedback control policy (75), we need to find the hedging point and buffer sizes. In order to allocate capacity for each part type, we conceptually slice the machines of the original system and separate them as two single-part-type approximate linear systems (Fig. 29). To ensure the approximate linear systems are as close as possible to the original system, we have to choose parameters and allocate capacity properly for each partial machine. Since we do not consider the interaction between two part types, the approximate linear system is a conservative approximate of the original system. The actual production could be slightly greater than the planned production based on the approximate linear system.

Define m_{ij} to be the partial machine which performs the i^{th} operation in approximate linear system j $(i = 1, 2; j = 1, 2)$. Each partial machine does only one operation. Approximate linear system j $(j = 1, 2)$ consists of two partial machines, m_{ij} $(i = 1, 2)$, and one buffer, b_j.

Define P_{ij} and R_{ij} be the failure and repair rate of Partial Machine m_{ij} (i=1,2; j=1,2). Define D_{ij} be the isolated capacity of Partial Machine m_{ij} (i=1,2; j=1,2), which is the maximal demand that can be achieved by m_{ij}. Note that the partial machine parameters R, P, and D are only used for the calculation of the hedging point and buffer sizes. The real-time production control is conducted by solving (75).

Partial Machines m_{i1} and m_{i2} must be up and down at the same time as Machine i of the original system is. Therefore, the partial machine parameters are determined as follows:

$$R_{i1} = R_{i2} = r_i, \quad (i = 1, 2);$$

$$P_{i1} = P_{i2} = p_i, \quad (i = 1, 2). \tag{76}$$

By taking the time average of the instantaneous capacity constraints of linear program (75), we have

$$\begin{aligned} \tau_{11}\bar{u}_{11} + \tau_{12}\bar{u}_{12} &\leq \frac{r_1}{r_1 + p_1}; \\ \tau_{21}\bar{u}_{21} + \tau_{22}\bar{u}_{22} &\leq \frac{r_2}{r_2 + p_2}; \end{aligned} \tag{77}$$

where $\bar{u}_{ij}$ is the time average of the production rate u_{ij}.

For a partial machine, the isolated capacity is the upper bound of the average production rate. Therefore, D_{ij} should satisfy

$$\begin{aligned} \tau_{11}D_{11} + \tau_{12}D_{12} &= \frac{r_1}{r_1 + p_1}; \\ \tau_{21}D_{21} + \tau_{22}D_{22} &= \frac{r_2}{r_2 + p_2}; \end{aligned} \tag{78}$$

Assume that a feasible demand set (d_1, d_2) is given. Let us consider a simple extreme case. Suppose that the demand for Type 1 parts is much higher than the demand for Part Type 2. Then, we should allocate most of the system capacity to produce Type 1 parts. We would do the reverse if the demand for Type 2 parts were much greater than that for Type 1 parts. Therefore, it is reasonable to choose the isolated capacities of the partial machines to be proportional to the demands,

$$\begin{Bmatrix} D_{11} \\ D_{12} \end{Bmatrix} = \frac{1}{\rho_1} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix};$$

$$\begin{Bmatrix} D_{21} \\ D_{22} \end{Bmatrix} = \frac{1}{\rho_2} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix}; \tag{79}$$

where ρ_i $(i = 1, 2)$ is the *capacity coefficient* of Machine i of the original system.

These parameters are chosen such that (78) is satisfied.

Plugging (79) into (78), and after manipulation, we obtain

$$\begin{aligned}\rho_1 &= \frac{1}{r_1}(r_1+p_1)(\tau_{11}d_1+\tau_{12}d_2);\\ \rho_2 &= \frac{1}{r_2}(r_2+p_2)(\tau_{21}d_1+\tau_{22}d_2).\end{aligned}$$

The demand set is feasible if and only if

$$\rho_i \leq 1, \quad (i=1,2).$$

This is a convenient way to check the feasibility of the demand.

6.7.2 The buffer hedging levels and spaces

Define z_j^b to be the *hedging level* of Buffer j $(j=1,2)$. It is the number of parts in Buffer j when the system reaches the hedging point. It is given by

$$z_j^b = z_{1j} - z_{2j}, \quad (j=1,2). \tag{80}$$

Define z_j^s to be the *hedging space* of Buffer j $(j=1,2)$. It is the room left for more parts in Buffer j when the system reaches the hedging point. it satisfies

$$z_j^s = B_j - z_j^b, \quad (j=1,2). \tag{81}$$

By applying the results of single-part-type systems of Section 4 to approximate linear system j $(j=1,2)$, the buffer hedging level and space are determined by the following nonlinear program

$$\min\{z_j^b + z_j^s\} \tag{82}$$

subject to:

$$\begin{aligned}\frac{1}{d_j}z_j^b + \frac{R_{1j}+P_{1j}}{R_{1j}P_{1j}}f_{2j}^s - \frac{1}{d_j}z_j^b f_{2j}^s &= \frac{1}{R_{1j}};\\ \frac{1}{d_j}z_j^s + \frac{R_{2j}+P_{2j}}{R_{2j}P_{2j}}f_{1j}^b - \frac{1}{d_j}z_j^s f_{1j}^b &= \frac{1}{R_{2j}};\end{aligned}$$

$$f_{1j}^b \leq 1 - \frac{d_j}{D_{1j}};$$
$$f_{2j}^s \leq 1 - \frac{d_j}{D_{2j}};$$
$$f_{1j}^b \geq 0, \quad f_{2j}^s \geq 0;$$
$$z_j^b \geq 0, \quad z_j^s \geq 0;$$

where the first two equality constraints formulate the relationship among the buffer hedging level and space, starvation and blockage, partial machine parameters, and the demand. The next two inequality constraints ensure that the sub-system has enough capacity to achieve the demand.

6.7.3 The buffer sizes and average buffer levels

The buffer sizes are given by

$$B_j = z_j^b + z_j^s, \quad (j = 1, 2). \tag{83}$$

The average buffer levels are

$$\bar{b}_j = z_j^b + (\Delta_{2j} - \Delta_{1j}), \tag{84}$$

where the *average surplus losses* are

$$\Delta_{ij} = \{\frac{R_{ij}P_{ij}}{R_{ij} + P_{ij}}\}\{\frac{d_j}{2}\}\{\frac{(R_{ij} + P_{ij})D_{ij}}{(R_{ij} + P_{ij})D_{ij} - R_{ij}d_j}\}\{(\frac{1}{R_{ij}})^2 + (\frac{f_{ij}^s}{P_{ij}})^2 + (\frac{f_{ij}^b}{P_{ij}})^2\},$$

$$(i = 1, 2; j = 1, 2).$$

6.7.4 The hedging point

The components of the hedging point are given by (see Section 4.7.4)

$$\begin{aligned} z_{2j} &= (\frac{d_j}{2})\{(\frac{1}{P_{2j}})^2 + (\frac{f_{2j}^s}{P_{2j}})^2\}\{\frac{P_{2j}D_{2j}}{(R_{2j} + P_{2j})D_{2j} - R_{2j}d_j}\}; \\ z_{1j} &= z_j^b + z_{2j}, \quad (j = 1, 2). \end{aligned} \tag{85}$$

We have now constructed a real-time scheduler for the two-machine, two-part-type system. It is an approximate solution to (69). In Section 6.6, the feedback controller

is established as a linear programming problem with unknown parameters, namely, the buffer sizes and the hedging point. In this section, the buffer sizes and hedging point are determined.

6.8 The algorithm

In this section, we summarize the steps of the production scheduling algorithm.

Step 1: Collect the input data, which consists of the failure rate p_i, the repair rate r_i, and the processing time, τ_{ij} for Part Type j ($j = 1, 2$) on Machine i ($i = 1, 2$), and the demands, d_j ($j = 1, 2$).

Step 2: Assign the repair and failure rates, R_{ij} and P_{ij}, and allocate isolated capacity D_{ij} for partial machine m_{ij} in the approximate system, according to (76) and (79).

Step 3: Calculate the buffer hedging level z_j^b and hedging space z_j^s ($j = 1, 2$), and the starvation and blockage fractions for each approximate linear system by solving the the nonlinear program (82). Then, calculate the buffer size, B_j, by summing the buffer hedging level and hedging space (83).

Step 4: Calculate the components, z_{ij} ($i = 1, 2; j = 1, 2$), of the hedging point according to (85).

Step 5: Using the feedback information of surplus $x_{ij}(t)$ ($i = 1, 2; j = 1, 2$) and machine state $\alpha_i(t)$ ($i = 1, 2$), calculate the time-varying production rates, $u_{ij}(t)$ ($i = 1, 2; j = 1, 2$), in real time by solving the linear program (75).

Step 6: The loading times for each part at each machine are determined in real time by a heuristic policy called *staircase strategy* [2]. That is, whenever the actual cumulative production is less than the integral of the production rate u_{ij}, load a Type j part onto Machine i. If there is more than one part type is eligible for loading, choose the one which is farthest behind.

Step 7: If any one of the demands or the machine parameters changes, go to Step 2.

The hierarchical structure described in Section 4.8 is suitable for this algorithm as well.

6.9 Simulation example

In this section, we demonstrate a simulation example of the production control algorithm. For the system described in the beginning of Section 6, we choose the machine parameters as follows:

$$r_1 = 0.5, \quad p_1 = 0.1;$$
$$r_2 = 0.5, \quad p_2 = 0.1;$$

$$\tau_{11} = 0.5, \quad \tau_{12} = 0.3;$$
$$\tau_{21} = 0.5, \quad \tau_{22} = 0.3;$$

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

Given that $d_1 = 1.1$ and $d_2 = 0.9$ lots/day, the buffer sizes and hedging point are calculated at the top level of the hierarchy. To solve the non-linear program, we used a commercially available software package [8]. The results are listed in the following

$$B_1 = 5 \qquad B_2 = 4;$$
$$z_{11} = 4.07, \quad z_{21} = 2.05;$$
$$z_{12} = 3.34, \quad z_{22} = 1.68;$$

where the buffer sizes are rounded up to integers. The simulation program that we use is called HIERCSIM which is developed by B. Darakananda [12]. Fig.30 and Fig.31 illustrate the cumulative productions of Part Type 1 and 2 respectively. The straight line is the cumulative demand for Type 1 parts. The upper curve is the cumulative input of the raw Type 1 parts at Machine 1. The lower curve is the cumulative output of the final Type 1 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 the cumulative production. It is almost impossible to tell the difference between the middle and bottom level results. Both part types are produced simultaneously.

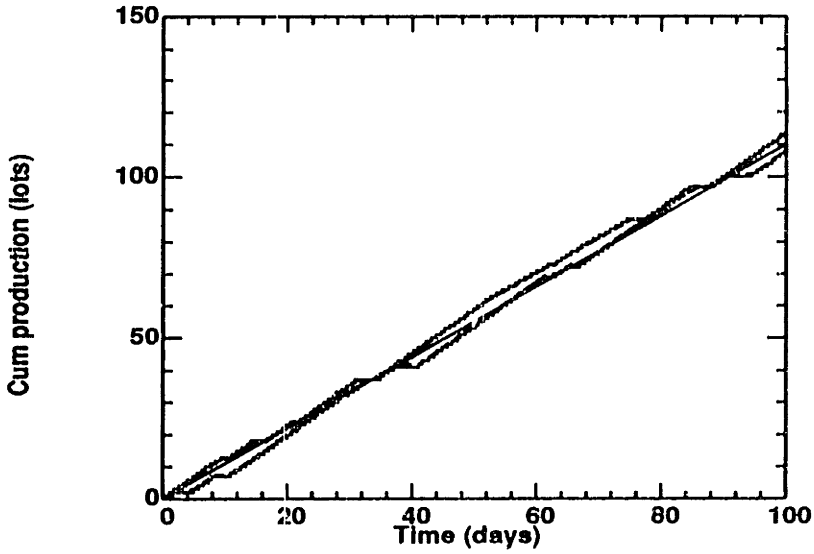

Figure 30: The cumulative production of part type 1

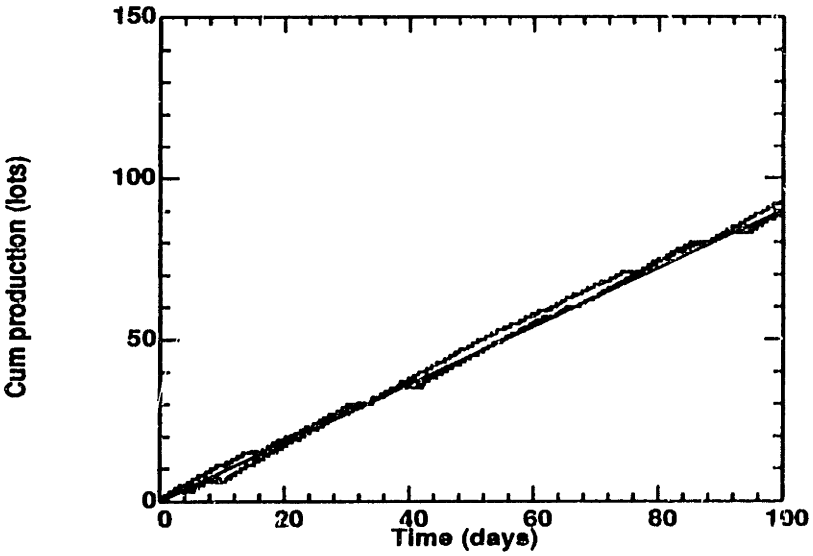

Figure 31: The cumulative production of part type 2

6.10 Summary

In this section, a real-time feedback control algorithm has been developed for the scheduling of two-machine, two-part-type systems. The simulation results verify that it works well. In the following section, we extend the algorithm to N-machine, M-part-type systems.

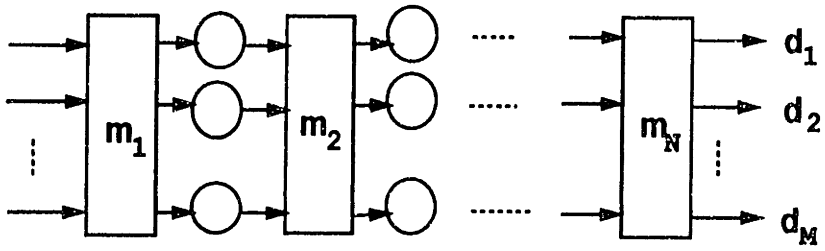

Figure 32: N-machine, M-part-type system

7 N-machine, M-part-type systems

In this section we study the N-machine, M-part-type production lines. As illustrated in Fig. 32, the system consists of N machines and $(N-1)M$ buffers. For Machine i ($i = 1, \ldots, N$), the failure rate is p_i and the repair rate is r_i. M part types are produced. Each Type j ($j = 1, \ldots, M$) part needs an operation with processing time τ_{ij} on Machine i ($i = 1, \ldots, N$). Buffers are located between machines. Buffer (i, j) holds only Type j parts. The Type j parts travel in a fixed sequence: Machine 1, Buffer $(1, j)$, Machine 2, ..., Buffer $(N-1, j)$, Machine N. The demand for Type j parts is d_j ($j = 1, \ldots, M$). 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 both starved and 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.

7.1 Dynamic optimization

For the system described above, the production control is formulated as a dynamic programming problem:

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x, b)dt \mid x(t_0), \alpha(t_0)\} \tag{86}$$

subject to:

$$\begin{aligned} &\textstyle\sum_{j=1}^{M} \tau_{ij} u_{ij} \leq \alpha_i \quad (i = 1, \ldots, N); \\ &u_{ij} \geq 0, \qquad\qquad (i = 1, \ldots, N; j = 1, \ldots, M); \end{aligned}$$

where the dynamics and the buffer constraints of the system are

$$\begin{aligned} &\dot{x}_{ij} = u_{ij} - d_j \qquad (i = 1, \ldots, N; j = 1, \ldots, M); \\ &\dot{b}_{ij} = u_{ij} - u_{i+1j} \quad (i = 1, \ldots, N-1; j = 1, \ldots, M); \\ &B_{ij} \geq b_{ij} \geq 0, \qquad (i = 1, \ldots, N-1; j = 1, \ldots, M); \end{aligned}$$

in which the function $g(x, b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive or too negative; u_{ij} and x_{ij} are the production rate and surplus of the i^{th} operation of Type j parts; and B_{ij} and b_{ij} are buffer size and level of Buffer (i, j).

7.2 Feedback control law

The solution of the optimization problem (86) satisfies the following linear program if the optimal cost-to-go J is known [See Appendix A]:

$$\min_u \{\textstyle\sum_{i=1}^{N} \sum_{j=1}^{M} \dfrac{\partial J}{\partial x_{ij}} u_{ij}\} \tag{87}$$

subject to:

$$\begin{aligned} &\textstyle\sum_{j=1}^{M} \tau_{ij} u_{ij} \leq \alpha_i \quad (i = 1, \ldots, N); \\ &u_{ij} \geq 0, \qquad\qquad (i = 1, \ldots, N; j = 1, \ldots, M); \end{aligned}$$

where

$$\begin{aligned}
&\dot{x}_{ij} = u_{ij} - d_j \qquad (i = 1, \ldots, N; j = 1, \ldots, M);\\
&\dot{b}_{ij} = u_{ij} - u_{i+1j} \quad (i = 1, \ldots, N-1; j = 1, \ldots, M);\\
&B_{ij} \geq b_{ij} \geq 0, \qquad (i = 1, \ldots, N-1; j = 1, \ldots, M).
\end{aligned}$$

7.3 System behavior specification

In order to approximate the value function J, we specify the following system behavior requirement: When Machine i fails, keep the adjacent machines producing without changing the production plan until a related buffer is empty or full.

Let $\tilde{J}$ be an approximate of the optimal J function which give us the system behavior as specified above. It is possible to show that the system behavior specification imposes that the coefficient boundary defined by $\partial \tilde{J}/\partial x_{ij} = 0$ is a hyperplane perpendicular to the axis x_{ij} and goes through the hedging point.

7.4 The conditional constraints

Define $z = \{z_{ij}; i = 1, \ldots, N; j = 1, \ldots, M\}$ to be the hedging point which is assumed to be independent of α. The following set of conditional constraints is imposed to the feedback controller to guide the system moving along the boundaries in x-space.

$$\begin{aligned}
&\text{if } x_{ij} = z_{ij},\\
&\quad B_{ij} > b_{ij} > 0,\\
&\text{and } \alpha_i = 1, \qquad \text{then} \quad u_{ij} = d_j, \quad (i = 1, \ldots, N; j = 1, \ldots, M);\\
&\text{if } b_{ij} = 0, \qquad \text{then} \quad u_{ij} \geq u_{i+1,j}, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M);\\
&\text{if } b_{ij} = B_{ij}, \qquad \text{then} \quad u_{ij} \leq u_{i+1,j}, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M).
\end{aligned}$$

7.5 The linear program

In order to comply the system behavior requirement of Section 7.3, the feedback controller becomes:

$$\min_u \{\sum_{i=1}^{N} \sum_{j=1}^{M} a_{ij}(x, \alpha, t)(x_{ij} - z_{ij})u_{ij}\} \tag{88}$$

subject to:

$$\sum_{j=1}^{M} \tau_{ij} u_{ij} \leq \alpha_i \quad (i = 1, \ldots, N);$$
$$u_{ij} \geq 0, \qquad (i = 1, \ldots, N; j = 1, \ldots, M);$$

if $x_{ij} = z_{ij}$,

$B_{ij} > b_{ij} > 0$,

and $\alpha_i = 1$, then $u_{ij} = d_j$, $(i = 1, \ldots, N; j = 1, \ldots, M)$;

if $b_{ij} = 0$, then $u_{ij} \geq u_{i+1,j}$, $(i = 1, \ldots, N-1; j = 1, \ldots, M)$;

if $b_{ij} = B_{ij}$, then $u_{ij} \leq u_{i+1,j}$, $(i = 1, \ldots, N-1; j = 1, \ldots, M)$;

where

$$\dot{x}_{ij} = u_{ij} - d_j, \qquad (i = 1, \ldots, N; j = 1, \ldots, M);$$
$$\dot{b}_{ij} = u_{ij} - u_{i+1j}, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M);$$
$$B_{ij} \geq b_{ij} \geq 0, \qquad (i = 1, \ldots, N-1; j = 1, \ldots, M);$$

in which $a_{ij}(x, \alpha, t)$ is a positive function over the feasible region in x-space. For simplicity, we choose $a_{ij}(x, \alpha, t) = 1$, (for all i, j). In the linear program, the hedging point and the buffer sizes are unknown. They are determined in the next subsection.

7.6 Control parameter estimation

In this section we estimate the unknown parameters for the real-time production rate control. In doing so, we extend the results of the two-machine, two-part-type system in the preceding section to the N-machine, M-part-type system.

7.6.1 The capacity allocation

In order to allocate capacity for each part type, we imagine that we cut the machines in the original system into partial machines and separate them into M single-part-type approximate linear systems (Fig. 33). Then we choose the parameters and capacity for each partial machine such that the approximate linear systems are as close as possible to the original system. The approximate linear system is a conservative approximate since we do not consider the interactions among different part types.

Define m_{ij} be the partial machine which performs the i^{th} operation in approximate linear system j $(i = 1, \ldots, N; j = 1, \ldots, M)$. Each partial machine does only one operation. Approximate linear system j $(j = 1, \ldots, M)$ consists of N partial machines

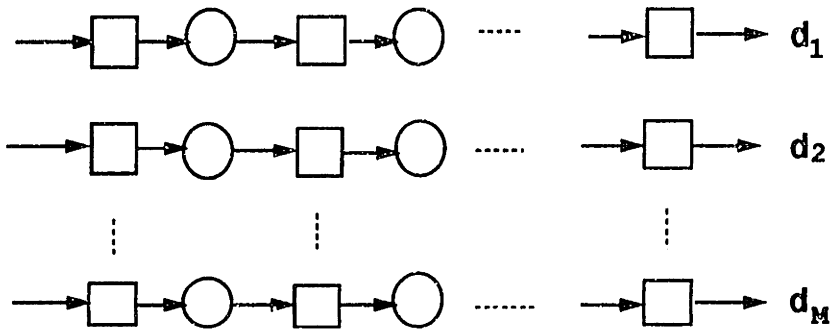

Figure 33: The approximate linear systems of the N-machine, M-part-type case

and $N-1$ buffers.

Define R_{ij} and P_{ij} be the repair and failure rate of Partial Machine m_{ij} $(i = 1,\ldots,N; j = 1,\ldots,M)$. Define D_{ij} be the isolated capacity of Partial Machine m_{ij} $(i = 1,\ldots,N; j = 1,\ldots,M)$, which is the maximal demand that can be achieved by m_{ij}. Note that the partial machine parameters R, P, and D are only used for the calculation of the hedging point and buffer sizes. The real-time production control is calculated by solving (88).

Partial Machines m_{ij} $(j = 1,\ldots,M)$ must be up and down at the same time as Machine i of the original system is. Therefore, the partial machine parameters are

$$\begin{aligned} R_{ij} &= r_i, \quad (i = 1,\ldots,N; j = 1,\ldots,M); \\ P_{ij} &= p_i, \quad (i = 1,\ldots,N; j = 1,\ldots,M). \end{aligned} \tag{89}$$

By taking the time average of the instantaneous capacity constraints in (88), we have

$$\sum_{j=1}^{M} \tau_{ij}\bar{u}_{ij} \leq \frac{r_i}{r_i + p_i}, \quad (i = 1,\ldots,N); \tag{90}$$

where $\bar{u}_{ij}$ is the time average of the production rate u_{ij}.

For a partial machine, the isolated capacity is the upper bound of the average

production rate. Therefore, D_{ij} should satisfy

$$\sum_{j=1}^{M} \tau_{ij} D_{ij} = \frac{r_i}{r_i + p_i}, \quad (i = 1, \ldots, N). \tag{91}$$

Assume that a feasible demand set $(d_j;\ j = 1, \ldots, M)$ is given. With the same reasoning as for the two-machine, two-part-type systems in Section 6, we choose the isolated capacities of the partial machines to be proportional to the demands.

$$\begin{Bmatrix} D_{i1} \\ D_{i2} \\ \ldots \\ D_{iM} \end{Bmatrix} = \frac{1}{\rho_i} \begin{Bmatrix} d_1 \\ d_2 \\ \ldots \\ d_M \end{Bmatrix}, \quad (i = 1, \ldots, N); \tag{92}$$

where ρ_i is the capacity coefficient of the Machine i in the original system which is chosen such that (91) is satisfied.

Plugging (92) into (91), after manipulation, we have

$$\rho_i = \frac{1}{r_i}(r_i + p_i)(\tau_{i1} d_1 + \ldots + \tau_{iM} d_M), \quad (i = 1, \ldots, N). \tag{93}$$

The demand set is feasible if and only if

$$\rho_i \leq 1, \quad (i = 1, \ldots, N).$$

7.6.2 The buffer hedging levels and spaces

Define z_{ij}^b to be the *hedging level* of Buffer (i, j). It is the number of parts in the buffer when the system reaches the hedging point. It satisfies

$$z_{ij}^b = z_{ij} - z_{i,j+1}, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M). \tag{94}$$

Define z_{ij}^s to be the *hedging space* of Buffer (i, j). It is the room left for more parts in the buffer when the system reaches the hedging point. It is given by

$$z_{ij}^s = B_{ij} - z_{ij}^b, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M). \tag{95}$$

Let f^s_{ij} and f^b_{ij} be the starvation and blockage fraction as defined in Section 3. By applying the results of single-part-type systems in Section 5 to approximate linear system j ($j = 1, \ldots, M$), the hedging buffer levels and spaces are determined by solving the following non-linear program

$$\min_{z^b, z^s, f^b, f^s} \sum_{i=1}^{N-1} \{z^b_{ij} + z^s_{ij}\} \tag{96}$$

subject to:

$$\frac{z^b_{i-1,j}}{d_j} - \frac{f^s_{i-1,j}}{P_{i-1,j}} + \frac{R_{i-1,j} + P_{i-1,j}}{R_{i-1,j} P_{i-1,j}} f^s_{ij} + \frac{f^b_{ij}}{R_{i-1,j}} - \frac{z^b_{i-1,j} f^s_{ij}}{d_j}$$

$$- \frac{z^b_{i-1,j} f^b_{ij}}{d_j} + \frac{f^s_{i-1,j} f^b_{ij}}{P_{i-1,j}} = \frac{1}{R_{i-1,j}}, \quad (i = 2, \ldots, N);$$

$$\frac{z^s_{ij}}{d_j} + \frac{f^s_{ij}}{R_{i+1,j}} + \frac{R_{i+1,j} + P_{i+1,j}}{R_{i+1,j} P_{i+1,j}} f^b_{ij} - \frac{f^b_{i+1,j}}{P_{i+1,j}} - \frac{z^s_{ij} f^s_{ij}}{d_j}$$

$$- \frac{z^s_{ij} f^b_{ij}}{d_j} + \frac{f^s_{ij} f^b_{i+1,j}}{P_{i+1,j}} = \frac{1}{R_{i+1,j}}, \quad (i = 1, \ldots, N-1);$$

$$\begin{aligned}
f^s_{1j} &= 0, \qquad f^b_{Nj} = 0; \\
f^s_{ij} + f^b_{ij} &\leq 1 - \tfrac{d_j}{D_{ij}}, \quad (i = 1, \ldots, N); \\
f^s_{ij} &\geq 0, \qquad f^b_{ij} \geq 0, \quad (i = 1, \ldots, N); \\
z^b_{ij} &\geq 0, \qquad z^s_{ij} \geq 0, \quad (i = 1, \ldots, N-1).
\end{aligned}$$

7.6.3 The buffer sizes and the average buffer levels

The buffer sizes are given by

$$B_{ij} = z^b_{ij} + z^s_{ij}, \quad (i = 1, \ldots, N-1; j = 1, \ldots, M).$$

The average buffer levels are

$$\bar{b}_{ij} = z^b_{ij} + (\Delta_{i+1,j} - \Delta_{ij}), \qquad (i = 1, \ldots, N-1; j = 1, \ldots, M); \tag{97}$$

where Δ_{ij} is the *surplus loss* which is approximately given by

$$\Delta_{ij} = \{\frac{R_{ij}P_{ij}}{R_{ij}+P_{ij}}\}\{\frac{d_j}{2}\}\{\frac{(R_{ij}+P_{ij})D_{ij}}{(R_{ij}+P_{ij})D_{ij}-R_{ij}d_j}\}\{(\frac{1}{R_{ij}})^2+(\frac{f_{ij}^s}{P_{ij}})^2+(\frac{f_{ij}^b}{P_{ij}})^2\},$$

$$(i=1,\ldots,N; j=1,\ldots,M).$$

7.6.4 The hedging point

The components of the hedging point are given by

$$z_{Nj} = \frac{d_j}{2}(\frac{R_{Nj}P_{Nj}D_{Nj}}{(R_{Nj}+P_{Nj})D_{Nj}-R_{Nj}d_j})\{(\frac{1}{R_{Nj}})^2+(\frac{f_{Nj}^s}{P_{Nj}})^2\};$$

$$z_{ij} = z_{ij}^b + z_{i+1,j}, \quad (i=1,\ldots,N-1; j=1,\ldots,M). \tag{98}$$

7.7 The algorithm

We have constructed a real-time feedback control algorithm for N-machine, M-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 rate p_i, the repair rate r_i, and the processing time, τ_{ij} for part type j ($j=1,\ldots,M$) on Machine i ($i=1,\ldots,N$), and the demands.

Step 2: Assign parameters, R_{ij} and P_{ij}, and allocate isolated capacity D_{ij} for partial machine m_{ij} in the approximate linear systems, according to (89) and (92).

Step 3: Calculate the buffer hedging level z_{ij}^b and hedging space z_{ij}^s ($i=1,\ldots,N; j=1,\ldots,M$), and the starvation and blockage fractions for each approximate linear system by solving the the nonlinear program (96). Then, calculate the buffer size, B_{ij}, by summing the buffer hedging level and hedging space.

Step 4: Calculate the components, z_{ij} ($i=1,\ldots,N; j=1,\ldots,M$), of the hedging point according to (98).

Step 5: Using the feedback information of surplus x_{ij} ($i=1,\ldots,N; j=1,\ldots,M$) and

machine state α_i $(i = 1, \ldots, N)$, calculate the production rates, u_{ij} $(i = 1, \ldots, N; j = 1, \ldots, M)$, in real time by solving the linear program (88).

Step 6: The loading times for each machine are determined by 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 7: If any one of the demands or the machine parameters changes, go to Step 2.

7.8 Example

A three-machine, three-part-type system is used for demonstration. The parameters are chosen as follows:

$$r_1 = 0.5, \quad r_2 = 0.8, \quad r_3 = 0.6,$$
$$p_1 = 0.1, \quad p_2 = 0.01, \quad p_3 = 0.2,$$

$$\tau_{11} = 0.5, \quad \tau_{12} = 0.3, \quad \tau_{13} = 0.4,$$
$$\tau_{21} = 0.3, \quad \tau_{22} = 0.2, \quad \tau_{23} = 0.4,$$
$$\tau_{31} = 0.4, \quad \tau_{32} = 0.3, \quad \tau_{33} = 0.5,$$

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

Given that $d_1 = 0.8$, $d_2 = 0.6$, and $d_3 = 0.3$ lots/day, the buffer sizes and hedging point are calculated at the top level by solving (96) and (98). They are listed in the following:

$$B_{11} = 1, \quad B_{12} = 1, \quad B_{13} = 1,$$
$$B_{21} = 1, \quad B_{22} = 1, \quad B_{23} = 1,$$

$$z_{11} = 0.85, \quad z_{21} = 0.55, \quad z_{31} = 0.55,$$
$$z_{12} = 0.64, \quad z_{22} = 0.41, \quad z_{32} = 0.41,$$
$$z_{13} = 0.32, \quad z_{23} = 0.21, \quad z_{33} = 0.21.$$

Figs. 34-36 illustrate the cumulative productions for different part types. Three part types are produced simultaneously. 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

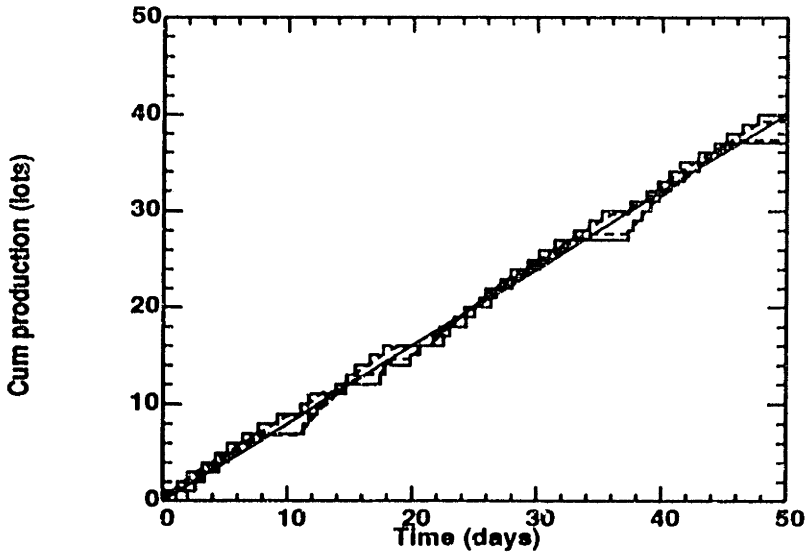

Figure 34: The cumulative production of Part Type 1

of the final products at Machine 3. The vertical and horizontal distance between the upper and lower curves indicate the instantaneous WIP inventory and throughput time respectively.

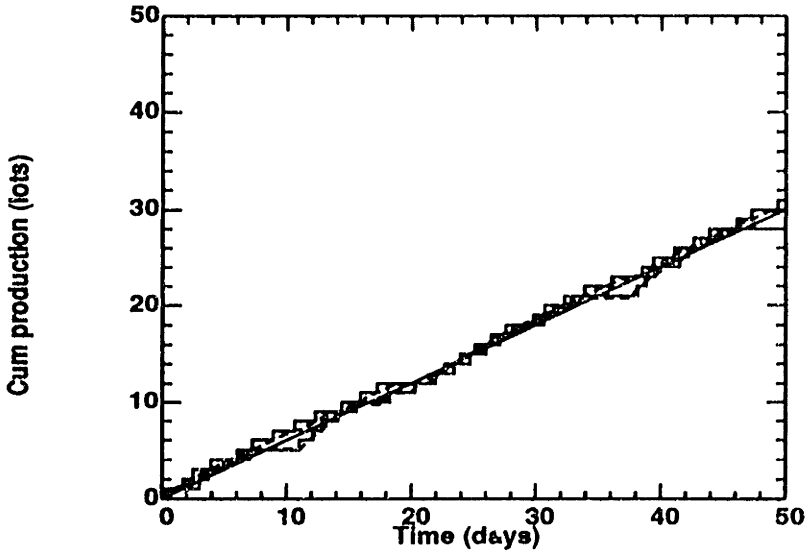

Figure 35: The cumulative production of Part Type 2

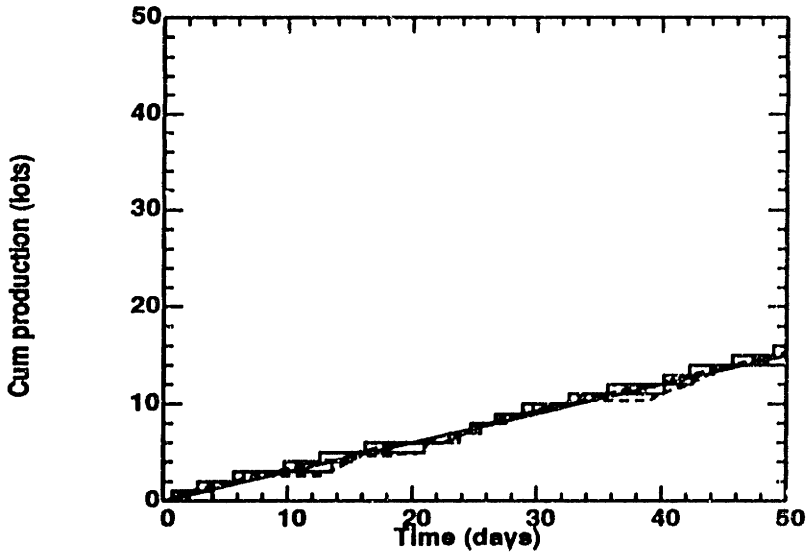

Figure 36: The cumulative production of Part Type 3

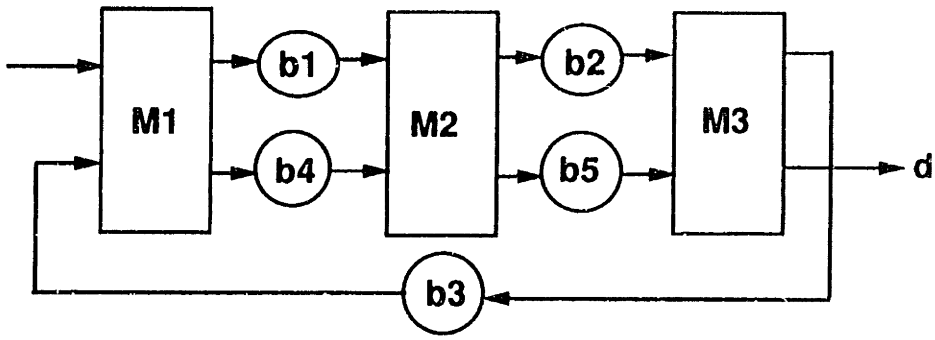

Figure 37: A three-machine, one-part-type reentrant system

8 Single-part-type reentrant systems

In this section, we study single-part-type reentrant systems. The system under study consists of N machines. For Machine i ($i = 1, 2, \ldots, N$), the time to fail and time to repair are modeled by exponentially distributed random variables with means $1/p_i$ and $1/r_i$ respectively. A single part type is produced with a total of L operations in a predefined sequence. Let θ_{ij} be the *operation index* which is binary and satisfies

$$\sum_{i=1}^{N} \theta_{ij} = 1 \quad (j = 1, \ldots, L) \tag{99}$$

If $\theta_{ij} = 1$, the j^{th} operation needs processing time τ_{ij} on Machine i. We assume that there are buffers between every consecutive pair of operations. Buffers are homogeneous, i.e., they hold identical parts that have had the same amount of work. A total of $L - 1$ buffers are therefore located between machines. Fig.37 illustrates an three-machine, one-part-type example which is to be used as a demonstration example at the end of this section. We assume that the machine that performs the first operation is never starved of raw parts, and that the machine that performs the last operation is never prevented from doing that operation by being blocked. The reentrant system is a useful model for the process flow in a semiconductor fabrication facility.

8.1 Dynamic optimization

For the single-part-type reentrant system, the production flow rate control is formulated as a dynamic optimization problem:

$$J(x(t_0), \alpha(t_0), t_0) = \min_u E\{\int_{t_0}^{T} g(x,b)dt \mid x(t_0), \alpha(t_0)\} \tag{100}$$

subject to:

$$\begin{aligned} &\textstyle\sum_{\{j|\theta_{ij}=1\}} \tau_{ij} u_j \leq \alpha_i, \quad (i = 1, 2, \ldots, N); \\ &u_j \geq 0, \qquad\qquad\qquad (j = 1, \ldots, L); \end{aligned}$$

where $g(x, b)$ is a convex function which penalizes $x(t)$ and $b(t)$ for being too positive or too negative. The system dynamics and buffer constraints are

$$\begin{aligned} &\dot{x}_j = u_j - d, \qquad (j = 1, \ldots, L); \\ &\dot{b}_j = u_j - u_{j+1}, \quad (j = 1, \ldots, L-1); \\ &B_j \geq b_j \geq 0, \qquad (j = 1, \ldots, L-1); \end{aligned} \tag{101}$$

where B_j is the buffer size which is to be determined.

8.2 Feedback control law

If the optimal value function J is known, a necessary optimality condition for production policy u is the following linear program.

$$\min_u \{\sum_{j=1}^{L} \frac{\partial J}{\partial x_j} u_j\} \tag{102}$$

subject to:

$$\begin{aligned} &\textstyle\sum_{\{j|\theta_{ij}=1\}} \tau_{ij} u_j \leq \alpha_i, \quad (i = 1, 2, \ldots, N); \\ &u_j \geq 0, \qquad\qquad\qquad (j = 1, \ldots, L). \end{aligned}$$

The system dynamics are governed by (101). By inspecting (102), we see the following: The objective function of (102) is linear in production rate u. The capacity constraints form a convex polyhedral set in u-space. Therefore, the linear program (102) divides the x-space into mutually exclusive regions which correspond to the extreme points

of the constraint set. Within each of the regions, the production rates are constant. Since the coefficients of u in the objective of (102) are functions of x and the right hand side of the constraint set is the random machine state, the linear program represents a feedback controller. When the machine state α and production surplus x are fed back from the production site, the scheduler (102) generates new productin rate u. The production rates 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.

As we did in the earlier sectins, the optimal value function J is assumed to be differentiable with respect to x and t. The point in x-space at which the gradient of J is 0 is called the *hedging point*, which is the desirable operating state of the system. The feedback controller (102) always attempts to drive the system to the hedging point, and to keep it there.

In the linear program, the J function and the buffer sizes are unknown. They are estimated in the subsequent subsections.

8.3 System behavior specification

In this subsection, we specify the desirable system behavior objective, which is used to approximate the value function J. The following is the specification on system behavior:

When Machine i fails, keep the adjacent machines producing without changing the production plan until a related buffer is empty or full.

As we see in following subsections, the behavior requirement helps to reduce the complexity of the scheduling algorithm and make it easier for implementation. For example, when a single machine in a huge factory fails, we do not want change the production plans 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 hierarchical structure.

8.4 The boundary shape in x-space

By using the system behavior requirement of the previous section, we can determine the desirable boundary shape in x-space. Assume that the system has reached the hedging point. The production rate decision is then $u_j = d$ $(j = 1, \ldots, L)$, so the system stays at the hedging point indefinitely. Then consider the following case. Suppose that at time t after the system has reached the hedging point, all machines except Machine k go down ($\alpha_k = 1$ and $\alpha_i = 0$, for $i \neq k$). According to the behavior requirement and the capacity constraints to the feedback controller (102), the production decision should be

$$u_j = d, \quad \text{if } \theta_{kj} = 1, \quad (j = 1, \ldots, L); \tag{103}$$

$$u_j = 0, \quad \text{if } \theta_{kj} = 0, \quad (j = 1, \ldots, L); \tag{104}$$

until Machine k is either starved or blocked. Before the starvation or blockage occurs, x_j $(\theta_{kj} = 1)$ is constant and the other production surpluses, x_l $(\theta_{kl} = 0)$, decrease at rate d. Let $\tilde{J}$ be an approximate of the optimal value function J, which yiels satisfactory system behavior. It is possible to show that the coefficient boundary defined by $\partial \tilde{J}/\partial x_j = 0$ must be perpendicular to the x_j axis in x-space. Otherwise, on the boundary, we have

$$x_j \neq \text{constant}, \quad (\text{for some j such that } \theta_{kj} = 1);$$

which contradicts the observation (103). From the discussion above, in order to satisfy the system behavior requirement of Section 8.3, the coefficient boundaries in x-space must have the following properties:

(a) The coefficient boundary, a hyperplane defined by $\partial \tilde{J}/\partial x_j = 0$, must be perpendicular to the x_j axis,

(b) All coefficient boundaries must intersect at the hedging point.

These properties are used to determine the form of the objective function of the feedback controller (102) in Section 8.6.

8.5 The conditional constraints

To avoid chattering on the boundaries in x-space and to comply with the buffer constraints, a set of conditional constraints are imposed to the real-time production scheduler.

Define $z = (z_j; j = 1, \ldots, L)$ to be the hedging point which is the desirable operating state of the system. The conditional constraints are

$$\begin{array}{lll} \text{if } x_j = z_j, \quad B_j > b_j > 0, & & \\ \quad \text{and } \alpha_i = 1, \ (\theta_{ij} = 1), & \text{then} & u_j = d, \quad (j = 1, \ldots, L); \\ \text{if } b_j = 0, & \text{then} & u_j \geq u_{j+1}, \ (j = 1, \ldots, L-1); \\ \text{if } b_j = B_j, & \text{then} & u_j \leq u_{j+1}, \ (j = 1, \ldots, L-1); \end{array} \tag{105}$$

which imply that when the production surplus x_j reaches its component of the hedging point, z_j, the production rate, u_j, should be equal to the demand. This is so that chattering on the attractive boundary can never occur [16]. When Buffer j is empty, the machine which performs the $j+1^{th}$ operation cannot be faster than the upstream machine. When Buffer j is full, the machine which performs the j^{th} operation cannot be faster than the downstream machine.

8.6 The linear program

To ensure that the linear coefficient boundaries ($\partial \tilde{J}/\partial x_j = 0$) in x-space are perpendicular to axes, and go through the hedging point, the linear program (102) becomes

$$\min_u \{\sum_{j=1}^{L} a_j(x, \alpha, t)(x_j - z_j)u_j\} \tag{106}$$

subject to:

$$\begin{array}{ll} \sum_{\{j|\theta_{ij}=1\}} \tau_{ij} u_j \leq \alpha_i, & (i = 1, 2, \ldots, N); \\ u_j \geq 0, & (j = 1, \ldots, L); \end{array}$$

$$\begin{array}{lll} \text{if } x_j = z_j, \quad B_j > b_j > 0, & & \\ \quad \text{and } \alpha_i = 1, \ (\theta_{ij} = 1), & \text{then} & u_j = d, \quad (j = 1, \ldots, L); \\ \text{if } b_j = 0, & \text{then} & u_j \geq u_{j+1}, \ (j = 1, \ldots, L-1); \\ \text{if } b_j = B_j, & \text{then} & u_j \leq u_{j+1}, \ (j = 1, \ldots, L-1); \end{array}$$

where $a_j(x, \alpha, t)$ is a positive function. For simplicity, we choose $a_j(x, \alpha, t) = 1$, $(j = 1, \ldots, L)$.

In the linear program, the hedging point and the buffer sizes are still unknown.

They are estimated in the next subsection.

8.7 Control parameter estimation

In this subsection we estimate the unknown parameters of the production control linear program (106) by extending the results for the production lines to reentrant systems.

8.7.1 The capacity allocation

In order to allocate long-term capacity for each operation, we conceptually slice the machines in the original system and organize them into a single-part-type tandem production line. Then we use the results of single-part-type systems to determine the hedging point and buffer sizes in following subsections.

Define m_j to be the partial machine which performs the j^{th} operation. Each partial machine does only one operation. The approximate linear system consists of L partial machines and $L-1$ buffers. Note that the approximate linear system may contain more than one partial machine corresponding to a real machine since the flow is reentrant.

Define R_j and P_j to be the repair and failure rate of Partial Machine m_j ($j = 1,\ldots,L$). Define D_j to be the isolated capacity of Partial Machine m_j which is the maximal demand that can be achieved by m_j.

Partial Machine m_j must be up and down at the same time as Machine i (for $\theta_{ij} = 1$) of the original system. Therefore, the partial machine parameters are

$$\begin{aligned} R_j &= r_i, \quad \text{if } \theta_{ij} = 1, \quad (j = 1,\ldots,L); \\ P_j &= p_i, \quad \text{if } \theta_{ij} = 1, \quad (j = 1,\ldots,L). \end{aligned} \tag{107}$$

By taking the time average of the instantaneous capacity constraints in (106), we have

$$\sum_{\{j|\theta_{ij}=1\}} \tau_{ij}\bar{u}_j \leq \frac{r_i}{r_i + p_i}, \quad (i = 1,\ldots,N), \tag{108}$$

where $\bar{u}_j$ is the time average of production rate $u_j(t)$.

For a partial machine, the average production rate is bounded from above by the

isolated capacity. Therefore, D_j must satisfy

$$\sum_{\{j|\theta_{ij}=1\}} \tau_{ij} D_j = \frac{r_i}{r_i + p_i}, \quad (i = 1, \ldots, N). \tag{109}$$

At each real machine, we would like to allocate the same capacity to all the reentrant operations. This is because that all reentrant operations have the same demand. In a long term, none of the operations has higher priority than the others. In other words, all partial machines corresponding to the same real machine have the same isolated capacity.

Let Γ_i be the isolated capacity of all partial machines which correspond to real machine i. We have

$$D_j = \Gamma_i, \quad \text{if } \theta_{ij} = 1, \quad (j = 1, \ldots, L). \tag{110}$$

Plugging (110) into (109), we get

$$\Gamma_i = \left\{ \frac{1}{\sum_{\{j|\theta_{ij}=1\}} \tau_{ij}} \right\} \left\{ \frac{r_i}{r_i + p_i} \right\}, \quad (i = 1, \ldots, N). \tag{111}$$

8.7.2 The buffer hedging levels and spaces

Define z_j^b to be the *hedging level* of Buffer j. It is the number of parts in the buffer when the system reaches the hedging point. It satisfies

$$z_j^b = z_j - z_{j+1}, \quad (j = 1, \ldots, L-1). \tag{112}$$

Define z_j^s to be the *hedging space* of Buffer j. It is the room left for more parts in the buffer when the system reaches the hedging point. It is given by

$$z_j^s = B_j - z_j^b, \quad (j = 1, \ldots, L-1). \tag{113}$$

Let f_j^s and f_j^b be the starvation and blockage fraction. By applying the results of single-part-type production lines in Section 5 to the approximate linear system, the

hedging buffer levels and spaces are governed by the following nonlinear program:

$$\min_{z^b, z^s, f^b, f^s} \sum_{j=1}^{L-1} \{z_j^b + z_j^s\} \tag{114}$$

subject to:

$$\frac{z_{j-1}^b}{d} - \frac{f_{j-1}^s}{P_{j-1}} + \frac{R_{j-1}+P_{j-1}}{R_{j-1}P_{j-1}} f_j^s + \frac{f_j^b}{R_{j-1}} - (\frac{z_{j-1}^b}{d}) f_j^s$$

$$-(\frac{z_{j-1}^b}{d}) f_j^b + \frac{f_{j-1}^s}{P_{j-1}} f_j^b = \frac{1}{R_{j-1}}, \quad (j = 2, \ldots, L);$$

$$\frac{z_j^s}{d} + \frac{f_j^s}{R_{j+1}} + \frac{R_{j+1}+P_{j+1}}{R_{j+1}P_{j+1}} f_j^b - \frac{f_{j+1}^b}{P_{j+1}} - (\frac{z_j^s}{d}) f_j^s$$

$$-(\frac{z_j^s}{d}) f_j^b + \frac{f_{j+1}^b}{P_{j+1}} f_j^s = \frac{1}{R_{j+1}}, \quad (j = 1, \ldots, L-1);$$

$$\begin{aligned} f_1^s = 0, \qquad & f_L^b = 0; \\ f_j^s + f_j^b \ \leq \ & 1 - \tfrac{d}{D_j}, \ (j = 1, \ldots, L); \\ f_j^s \geq 0, \qquad & f_j^b \geq 0, \ (j = 1, \ldots, L); \\ z_j^b \geq 0, \qquad & z_j^s \geq 0, \ (j = 1, \ldots, L-1). \end{aligned}$$

8.7.3 The buffer sizes and average buffer levels

The buffer sizes are given by

$$B_j = z_j^b + z_j^s, \ \ (j = 1, \ldots, L-1).$$

The average buffer levels are

$$\bar{b}_j = z_j^b + (\Delta_{j+1} - \Delta_j), \quad (j = 1, 2, \ldots, L-1), \tag{115}$$

where Δ_j is the surplus loss at Partial Machine m_j, which is approximately given by (see Section 4.7.4)

$$\Delta_j = (\frac{R_j P_j}{R_j + P_j})(\frac{d}{2})(\frac{(R_j+P_j)D_j}{(R_j+P_j)D_j - R_j d})\{(\frac{1}{R_j})^2 + (\frac{f_j^s}{P_j})^2 + (\frac{f_j^b}{P_j})^2\}, \tag{116}$$

$$(j = 1, 2, \ldots, L).$$

8.7.4 The hedging point

The components of the hedging point are given by

$$z_L = (\frac{d}{2})\left(\frac{R_L P_L D_L}{(R_L + P_L)D_L - R_L d}\right)\left\{(\frac{1}{R_L})^2 + (\frac{f_L^s}{P_L})^2\right\};$$

$$z_j = z_j^b + z_{j+1}, \qquad (j = 1, \ldots, L-1). \tag{117}$$

8.8 The algorithm

The steps of the production control algorithm for single-part-type reentrant systems are summarized in the following:

Step 1: Collect the input data set, which consists of the failure rate p_i, the repair rate r_i, the processing time τ_{ij}, the operation index θ_{ij}, and the demand.

Step 2: Assign parameters, R_j and P_j, and allocate isolated capacity D_j for Partial Machine m_j, according to (107) and (110).

Step 3: Calculate the buffer hedging level z_j^b and hedging space z_j^s $(j = 1, \ldots, L-1)$, and the starvation and blockage fractions for each approximate linear system by solving the the nonlinear program (114). Then, calculate the buffer size, B_j, by summing the buffer hedging level and hedging space.

Step 4: Calculate the components, z_j $(j = 1, \ldots, L)$, of the hedging point according to (117).

Step 5: Using the feedback information of surplus x_j $(j = 1, \ldots, L)$ and machine state α_i $(i = 1, \ldots, N)$, calculate the production rates, u_j $(j = 1, \ldots, L)$, in real time by solving the linear program (106).

Step 6: The loading times for each machine are determined by the staircase strategy. That is, whenever the actual cumulative production is less than the integral of the production rate, load a part into the machine. If there is more than one operation eligible for loading, choose the one which is farthest behind.

Step 7: If either the demand or any one of the machine parameters changes, go to Step 2.

8.9 Example

In previous sections, we developed a algorithm for scheduling of single-part-type reentrant systems. In this section, we demonstrate a simulation example of the production control algorithm.

The system used for the demonstration consists of three machines and five buffers (Fig.37). One part type is produced. Each part needs six operations following the sequence: Machine 1, Buffer 1, Machine 2, Buffer 2, Machine 3, Buffer 3, Machine 1, Buffer 4, Machine 2, Buffer 5, Machine 3.

The parameters are

$$r_1 = 0.5, \quad r_2 = 0.33, \quad r_3 = 0.5,$$
$$p_1 = 0.1, \quad p_2 = 0.02, \quad p_3 = 0.1,$$

$$\tau_{11} = 0.5, \quad \tau_{22} = 0.3, \quad \tau_{33} = 0.4,$$
$$\tau_{14} = 0.3, \quad \tau_{25} = 0.3, \quad \tau_{36} = 0.4,$$

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

Given that the demand $d = 0.8$ lots/day, the buffer sizes and hedging point are calculated by solving (114) and (117), using a commercially available software package [8]. They are listed in the following:

$$B_1 = 1, \quad B_2 = 1, \quad B_3 = 5, \quad B_4 = 1, \quad B_5 = 1;$$

$$z_{11} = 3.36, \quad z_{22} = 3.36, \quad z_{33} = 3.36,$$
$$z_{14} = 0.87, \quad z_{25} = 0.87, \quad z_{36} = 0.87,$$

where the buffer sizes are rounded up to integers. The simulation program that we use is called HIERCSIM which was developed by B. Darakananda [12]. Fig.38 illustrates the cumulative production. The straight line is the cumulative demand. The upper curve is the cumulative input of the raw Type 1 parts at Machine 1. The

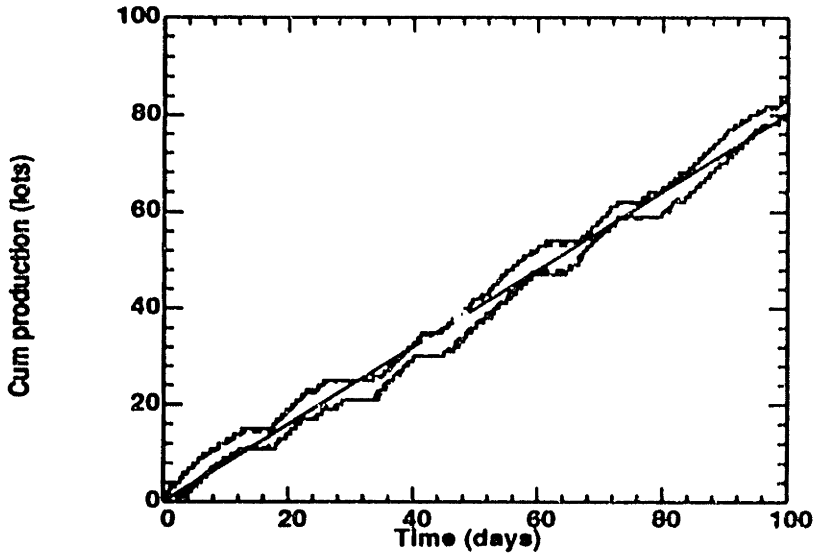

Figure 38: The simulation result of cumulative production of a single-part-type reentrant system

lower curve is the cumulative output of the final Type 1 products at Machine 3. 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. The vertical and horizontal distance between the upper and lower curves indicate the instantaneous WIP inventory and throughput time respectively.

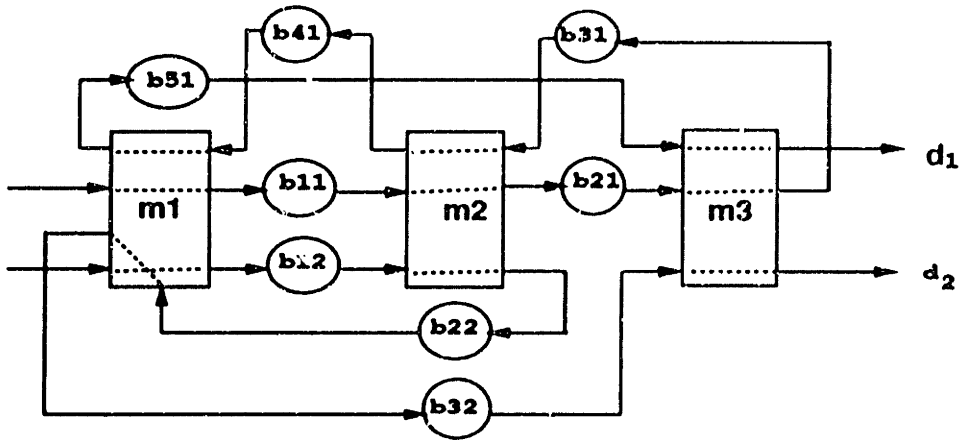

Figure 39: Three-machine, two-part-type reentrant system

9 Multiple-part-type reentrant systems

In this section, we study multiple-part-type reentrant systems. The system under study consists of N machines. For Machine i ($i = 1, 2, \ldots, N$), the failure rate is p_i and the repair rate is r_i. M part types are produced. Type k parts require a total of L_k operations following a predefined sequence. Let θ_{ijk} be the *operation index* which is binary and satisfies

$$\sum_{i=1}^{N} \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M). \tag{118}$$

If $\theta_{ijk} = 1$, the j^{th} operation of Type k parts needs a processing time τ_{ijk} on Machine i. We assume that there are buffers between every consecutive pair of operations. A total of $\sum_{k=1}^{M} L_k - M$ buffers are therefore located between machines. Fig. 39 illustrates an three-machine, two-part-type example which is to be used as an example at the end of this section. We assume that, for each part type, no machine can be starved for the first operation or blocked for the last operation. This multi part type reentrant system is a more suitable model for the reentrant production flow in semiconductor fabrication facilities than the models developed in previous sections.

9.1 Dynamic optimization

The following dynamic programming problem is formulated to determine the production control policy,

$$J(x(t_0),\alpha(t_0),t_0) = \min_u E\{\int_{t_0}^{T} g(x,b)dt \mid x(t_0),\alpha(t_0)\} \tag{119}$$

subject to:

$$\sum_{\{j,k|\theta_{ijk}=1\}} \tau_{ijk}u_{jk} \leq \alpha_i, \quad (i = 1,2,\ldots,N);$$
$$u_{jk} \geq 0, \quad (j = 1,\ldots,L_k; k = 1,\ldots,M);$$

where the system dynamics and buffer constraints are

$$\begin{aligned}
\dot{x}_{jk} &= u_{jk} - d_k, && (j = 1,\ldots,L_k; k = 1,\ldots,M);\\
\dot{b}_{jk} &= u_{jk} - u_{j+1,k}, && (j = 1,\ldots,L_k - 1; k = 1,\ldots,M);\\
B_{jk} &\geq b_{jk} \geq 0, && (j = 1,\ldots,L_k - 1; k = 1,\ldots,M).
\end{aligned}$$

9.2 The feedback controller

The optimal production policy u determined by the dynamic optimization (119) satisfies the following linear program if the J function is known [See Appendix A]:

$$\min_u \{\sum_{k=1}^{M}\sum_{j=1}^{L_k}(\frac{\partial J}{x_{jk}})u_{jk}\} \tag{120}$$

subject to:

$$\sum_{\{j,k|\theta_{ijk}=1\}} \tau_{ijk}u_{jk} \leq \alpha_i, \quad (i = 1,2,\ldots,N);$$
$$u_{jk} \geq 0 \quad (j = 1,\ldots,L_k; k = 1,\ldots,M);$$

where the system dynamics and buffer constraints are

$$\begin{aligned}
\dot{x}_{jk} &= u_{jk} - d_k, && (j = 1,\ldots,L_k; k = 1,\ldots,M);\\
\dot{b}_{jk} &= u_{jk} - u_{j+1,k}, && (j = 1,\ldots,L_k - 1; k = 1,\ldots,M);\\
B_{jk} &\geq b_{jk} \geq 0, && (j = 1,\ldots,L_k - 1; k = 1,\ldots,M).
\end{aligned}$$

It is possible to show that the linear program (120) divides the x-space into mutu-

ally exclusive regions. Within each of the regions, the production rates are constant. The linear program represents a feedback controller. When the machine state and the production surplus are fedback from the shop-floor, the scheduler (120) computes new rate $u(t)$. The production rates do not have to be calculated at every time instant. They need only to be computed when the machine state changes or when the production surplus reaches a boundary. As we did in the rpevious sections, the value function J is assumed to be differentiable with respect to x and t. The point in x-space at which the gradient of J is zero is called *hedging point* which is assumed to be independent of α. It is the desirable operating state of the system.

In the linear program, the gradient of J function and the buffer sizes are unknown. We construct approximations of these unknowns in the following subsections.

9.3 System behavior specification

In this section we specify the system behavior requirement which is used to determine the boundaries in x-space in the next subsection. In order to reduce complexity of the scheduling problem and smooth the implementation procedure of the production control algorithm, we would like to impose independence among the machines in the system such that the impact of machine failures is minimized. The following is the system behavior specification:

When Machine i fails, keep the adjacent machines producing without changing the production plan until a related buffer is empty or full.

9.4 The boundary shape in x-space

In this subsection, we determine the boundary shape in x-space by using the system behavior specification of the preceding subsection. In doing so, we suppose that the system has reached the hedging point. Then the production rates are $U_{jk} = d_k$ $(j = 1, \ldots, L_k; k = 1, \ldots, M)$, so the system stays at the hedging point indefinitely. Let us consider the following case. Suppose that at time t after the system has reached the hedging point, all machines but Machine i fail ($\alpha_i = 1$, $\alpha_l = 0$, for $l \neq i$). According to the system behavior requirement and the instantaneous capacity

constraints of (120), the production decision changes to

$$u_{jk} = d_k, \quad \text{if } \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M); \tag{121}$$

$$u_{jk} = 0, \quad \text{if } \theta_{ijk} = 0, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M); \tag{122}$$

until Machine i is either starved or blocked. Before the starvation or blockage occurs, x_{jk} ($\theta_{ijk} = 1$) is constant and the other production surpluses, x_{lk} ($\theta_{ilk} = 0$), decrease at rate d. Let $\tilde{J}$ be an approximate of the optimal J function which gives us the satisfatory system behavior. It is possible to show that the coefficient boundary defined by $\partial\tilde{J}/\partial x_{jk} = 0$ must be perpendicular to the x_{jk} axis. Otherwise, on the boundary, we have

$$x_{jk} \neq \text{constant}, \quad (\text{for some } j \text{ and } k \text{ such that } \theta_{ijk} = 1);$$

which contradicts the observation (121). The observation leads to the following properties of the boundaries in x-space: (i) The coefficient boundary defined by $\partial\tilde{J}/\partial x_{jk} = 0$ represents a hyperplane in x-space; (ii) the coefficient boundary is perpendicular to the x_{jk} axis in x-space; (iii) all coefficient boundaries intersect at the hedging point.

9.5 The conditional constraints

In order to avoid chattering, we introduce a set of conditional constraints to guide the system moving along the boundaries in x-space. Define $z = (z_{jk}; j = 1, \ldots, L_k; k = 1, \ldots, M)$ to be the hedging point which is the desirable operating state of the system. The conditional constraints are

$$\begin{array}{lll}
\text{if } x_{jk} = z_{jk}, & & \\
B_{jk} > b_{jk} > 0, \text{ and} & & \\
\alpha_i = 1, \ (\theta_{ijk} = 1), & \text{then } u_{jk} = d_k, & (j = 1, \ldots, L_k; k = 1, \ldots, M); \\
\text{if } b_{jk} = 0, & \text{then } u_{jk} \geq u_{j+1,k}, & (j = 1, \ldots, L_k - 1; k = 1, \ldots, M); \\
\text{if } b_{jk} = B_{jk}, & \text{then } u_{jk} \leq u_{j+1,k}, & (j = 1, \ldots, L_k - 1; k = 1, \ldots, M).
\end{array} \tag{123}$$

9.6 The linear program for real-time production control

To ensure that the system behaves as specified in Section 9.3 and to avoid chattering on the boundaries in x-space, the linear program (120) becomes

$$\min_{u}\{\sum_{k=1}^{M}\sum_{j=1}^{L_k}(x_{jk}-z_{jk})u_{jk}\} \tag{124}$$

subject to:

$$\sum_{\{j,k|\theta_{ijk}=1\}}\tau_{ijk}u_{jk}\leq\alpha_i, \quad (i=1,2,\ldots,N);$$
$$u_{jk}\geq 0, \quad (j=1,\ldots,L_k;k=1,\ldots,M);$$

if $x_{jk}=z_{jk}$,

$B_{jk}>b_{jk}>0$,

and $\alpha_i=1$, $(\theta_{ijk}=1)$, then $u_{jk}=d_k$, $(j=1,\ldots,L_k;k=1,\ldots,M)$;

if $b_{jk}=0$, then $u_{jk}\geq u_{j+1,k}$, $(j=1,\ldots,L_k-1;k=1,\ldots,M)$;

if $b_{jk}=B_{jk}$, then $u_{jk}\leq u_{j+1,k}$, $(j=1,\ldots,L_k-1;k=1,\ldots,M)$;

where the system dynamics and buffer constraints are

$$\dot{x}_{jk}=u_{jk}-d_k, \quad (j=1,\ldots,L_k;k=1,\ldots,M);$$
$$\dot{b}_{jk}=u_{jk}-u_{j+1,k}, \quad (j=1,\ldots,L_k-1;k=1,\ldots,M);$$
$$B_{jk}\geq b_{jk}\geq 0, \quad (j=1,\ldots,L_k-1;k=1,\ldots,M).$$

In the linear program, the hedging point and the buffer sizes are still unknown. They are determined in the next subsection.

9.7 Control parameter estimation

In this subsection we estimate the unknown parameters of the feedback control linear program (124). The results of single-part-type reentrant systems in Section 8 to the multiple-part-type reentrant systems.

9.7.1 The capacity allocation

To allocate capacity for each operation, we conceptually slice the machines in the original system and organize them into M single-part-type linear systems. An approximate linear system may contain more than one partial machine corresponding to a real machine since the flow is reentrant.

Define m_{jk} be the partial machine which performs the j^{th} operation in approximate linear system k ($j = 1, \ldots, L_k; k = 1, \ldots, M$). Each partial machine does only one operation. Approximate linear system k ($k = 1, \ldots, M$) consists of L_k partial machines and L_k-1 buffers.

Define R_{jk} and P_{jk} be the repair and failure rate of Partial Machine m_{jk} ($j = 1, \ldots, L_k; k = 1, \ldots, M$). Define D_{jk} be the isolated capacity of Partial Machine m_{jk} which is the maximal demand that can be achieved by m_{jk}.

Partial Machines m_{jk} ($\theta_{ijk} = 1$) must be up and down at the same time as Machine i of the original system. Therefore, the partial machine parameters are

$$\begin{aligned} &R_{jk} = r_i, \quad \text{if } \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M); \\ &P_{jk} = p_i, \quad \text{if } \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M). \end{aligned} \tag{125}$$

By taking the time average of the instantaneous capacity constraints in (124), we have

$$\sum_{\{j,k|\theta_{ijk}=1\}} r_{ijk}\bar{u}_{jk} \leq \frac{r_i}{r_i + p_i}, \quad (i = 1, \ldots, N), \tag{126}$$

where $\bar{u}_{jk}$ is the time average of production rate u_{jk}.

For a partial machine, the average production rate is bounded by the isolated capacity. Therefore , D_{jk} should satisfy

$$\sum_{\{j,k|\theta_{ijk}=1\}} r_{ijk} D_{jk} = \frac{r_i}{r_i + p_i}, \quad (i = 1, \ldots, N). \tag{127}$$

At each machine, we would like to allocate same capacity to all operations on the same part type. This is because that all operations on the same part type have the same demand. In other words, all partial machines which are corresponding to the same real machine and perform operations on the same part type have the same isolated capacity.

Let Γ_{ik} be the isolated capacity of all partial machines which correspond to Ma-

chine i and perform operations on Type k parts. Consequently, we have

$$D_{jk} = \Gamma_{ik}, \quad \text{if } \theta_{ijk} = 1, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M). \tag{128}$$

Plugging (128) into (127), we get

$$\sum_{k=1}^{M} \omega_{ik}\Gamma_{ik} = \frac{r_i}{r_i + p_i}, \quad (i = 1, \ldots, N), \tag{129}$$

where

$$\omega_{ik} = \sum_{\{j|\theta_{ijk}=1\}} \tau_{ijk}.$$

The part type with higher demand should be assigned more capacity at each machine. We choose the isolated capacities of the partial machines to be proportional to the demands.

$$\begin{Bmatrix} \Gamma_{i1} \\ \Gamma_{i2} \\ \ldots \\ \Gamma_{iM} \end{Bmatrix} = \frac{1}{\rho_i} \begin{Bmatrix} d_1 \\ d_2 \\ \ldots \\ d_M \end{Bmatrix}, \quad (i = 1, \ldots, N), \tag{130}$$

where ρ_i is the capacity coefficient of the Machine i in the original system which is chosen such that (129) is satisfied. Plugging (130) into (127), after manipulation, leads to

$$\rho_i = \frac{1}{r_i}(r_i + p_i)(\omega_{i1}d_1 + \ldots + \omega_{iM}d_M), \quad (i = 1, \ldots, N). \tag{131}$$

The demand set is feasible if and only if

$$\rho_i \leq 1, \quad (i = 1, \ldots, N).$$

9.7.2 The buffer hedging levels and spaces

Define z^b_{jk} to be the *hedging level* of Buffer (j, k). It is the number of parts in the buffer when the system reaches the hedging point. It satisfies

$$z^b_{jk} = z_{jk} - z_{j+1,k}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M). \tag{132}$$

Define z^s_{jk} to be the *hedging space* of Buffer (j,k). It is the room left for more parts in the buffer when the system reaches the hedging point. It is given by

$$z^s_{jk} = B_{jk} - z^b_{jk}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M). \tag{133}$$

Let f^s_{jk} and f^b_{jk} be the starvation and blockage fraction as defined in Section 3.

Each approximate linear system is a single-part-type production line. For approximate linear system k ($k = 1, \ldots, M$), the hedging buffer levels and spaces are governed by the following nonlinear program:

$$\min_{z^b, z^s, f^b, f^s} \sum_{j=1}^{L_k - 1} \{z^b_{jk} + z^s_{jk}\} \tag{134}$$

subject to:

$$\frac{z^b_{j-1,k}}{d_k} - \frac{f^s_{j-1,k}}{P_{j-1,k}} + \frac{R_{j-1,k} + P_{j-1,k}}{R_{j-1,k} P_{j-1,k}} f^s_{jk} + \frac{f^b_{jk}}{R_{j-1,k}} - \left(\frac{z^b_{j-1,k}}{d_k}\right) f^s_{jk}$$

$$-\left(\frac{z^b_{j-1,k}}{d_k}\right) f^b_{jk} + \frac{f^s_{j-1,k}}{P_{j-1,k}} f^b_{jk} = \frac{1}{R_{j-1,k}}, \quad (j = 2, \ldots, L_k);$$

$$\frac{z^s_{jk}}{d_k} + \frac{f^s_{jk}}{R_{j+1,k}} + \frac{R_{j+1,k} + P_{j+1,k}}{R_{j+1,k} P_{j+1,k}} f^b_{jk} - \frac{f^b_{j+1,k}}{P_{j+1,k}} - \left(\frac{z^s_{jk}}{d_k}\right) f^s_{jk}$$

$$-\left(\frac{z^s_{jk}}{d_k}\right) f^b_{jk} + \frac{f^b_{j+1,k}}{P_{j+1,k}} f^s_{jk} = \frac{1}{R_{j+1,k}}, \quad (j = 1, \ldots, L_k - 1);$$

$$\begin{aligned}
f^s_{1k} &= 0, & f^b_{L_k k} &= 0; \\
f^s_{jk} + f^b_{jk} &\leq 1 - \tfrac{d_k}{D_{jk}}, & (j &= 1, \ldots, L_k); \\
f^s_{jk} &\geq 0, & f^b_{jk} &\geq 0, & (j &= 1, \ldots, L_k); \\
z^b_{jk} &\geq 0, & z^s_{jk} &\geq 0, & (j &= 1, \ldots, L_k - 1).
\end{aligned}$$

9.7.3 The buffer sizes and average buffer levels

The buffer sizes are given by

$$B_{jk} = z^b_{jk} + z^s_{jk}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M). \tag{135}$$

The average buffer levels are

$$\bar{b}_{jk} = z^b_{jk} + (\Delta_{j+1,k} - \Delta_{jk}), \quad (j = 1, 2, \ldots, L_k - 1; k = 1, \ldots, M), \tag{136}$$

where Δ_{jk} is the surplus loss at Partial Machine m_{jk}, which is approximately given by (see Section 4.7.4)

$$\Delta_{jk} = \left(\frac{R_{jk}P_{jk}}{R_{jk}+P_{jk}}\right)\left(\frac{d_k}{2}\right)\left(\frac{(R_{jk}+P_{jk})D_{jk}}{(R_{jk}+P_{jk})D_{jk} - R_{jk}d_k}\right)\left\{\left(\frac{1}{R_{jk}}\right)^2 + \left(\frac{f^s_{jk}}{P_{jk}}\right)^2 + \left(\frac{f^b_{jk}}{P_{jk}}\right)^2\right\}, \tag{137}$$

$$(j = 1, 2, \ldots, L_k; k = 1, \ldots, M).$$

9.7.4 The hedging point

The components of the hedging point are given by

$$z_{L_kk} = \left(\frac{d_k}{2}\right)\left(\frac{R_{L_kk}P_{L_kk}D_{L_kk}}{(R_{L_kk}+P_{L_kk})D_{L_kk} - R_{L_kk}d_k}\right)\left\{\left(\frac{1}{R_{L_kk}}\right)^2 + \left(\frac{f^s_{L_kk}}{P_{L_kk}}\right)^2\right\};$$

$$z_{jk} = z^b_{jk} + z_{j+1,k}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M). \tag{138}$$

9.8 The algorithm

The steps of the production control algorithm for multiple-part-type systems are summarized in the following:

Step 1: Collect the input data set, which consists of the failure rate p_i, the repair rate r_i, the processing time, τ_{ijk}, and the operation index, θ_{ijk}, for the j^{th} operation of Part Type k ($k = 1, \ldots, M$) on Machine i ($i = 1, \ldots, N$), and the demands.

Step 2: Assign parameters, R_{jk} and P_{jk}, and allocate isolated capacity D_{jk} for partial-machine m_{jk} in the approximate linear systems, according to (125) and (128).

Step 3: Calculate the buffer hedging level z^b_{jk} and hedging space z^s_{jk} ($j = 1, \ldots, L_k - 1; k = 1, \ldots, M$), and the starvation and blockage fractions for each approximate linear system by solving the the nonlinear program (134). Then, calculate the buffer size, B_{jk}, by summing the buffer hedging level and hedging space.

Step 4: Calculate the components, z_{jk} ($j = 1, \ldots, L_k; k = 1, \ldots, M$), of the hedging point according to (138).

Step 5: Using the feedback information of surplus x_{jk} ($j = 1, \ldots, L_k; k = 1, \ldots, M$) and machine state α_i ($i = 1, \ldots, N$), calculate the production rates, u_{jk} ($j = 1, \ldots, L_k$; $k = 1, \ldots, M$), in real time by solving the linear program (124).

Step 6: The loading times for each machine are determined by the staircase strategy. That is, whenever the actual cumulative production is less than the integral of the production rate, load a part into the machine. If there is more than one part type eligible for loading, choose the one which is farthest behind.

Step 7: If any one of the demands or the machine parameters changes, go to Step 2.

9.9 Example

The system used for the demonstration consists of three machines and eight buffers (Fig.39). Two part types are produced. Type 1 parts need six operations following the sequence: Machine 1, Buffer (1,1), Machine 2, Buffer (1,2), Machine 3, Buffer (1,3), Machine 2, Buffer (1,4), Machine 1, Buffer (1,5), Machine 3. Type 2 parts need four operations following the sequence: Machine 1, Buffer (2,1), Machine 2, Buffer (2,2), Machine 1, Buffer (2,3), Machine 3. The parameters are chosen as follows:

$$r_1 = 0.5, \quad r_2 = 0.5, \quad r_3 = 0.5,$$
$$p_1 = 0.1, \quad p_2 = 0.1, \quad p_3 = 0.1,$$

$$\tau_{111} = 0.1, \quad \tau_{221} = 0.1, \quad \tau_{331} = 0.1,$$
$$\tau_{241} = 0.1, \quad \tau_{151} = 0.1, \quad \tau_{361} = 0.1,$$
$$\tau_{112} = 0.1, \quad \tau_{222} = 0.1, \quad \tau_{132} = 0.1,$$
$$\tau_{342} = 0.1$$

where the unit of r and p is 1/day. The unit of τ is a day. the unit of parts is a lot. The operation index matrices are

$$\Theta_1 = [\theta_{ij1}] = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix};$$

$$\Theta_2 = [\theta_{ij2}] = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Given that $d_1 = 2.0$ and $d_2 = 1.5$ lots/day, the buffer sizes and hedging point are calculated by solving the nonlinear program (134), using a commercially available software package [8]. They are listed in the following:

$$z_{111} = 15.03, \quad z_{221} = 15.03, \quad z_{331} = 9.89,$$
$$z_{241} = 9.89, \quad z_{151} = 2.01, \quad z_{361} = 1.82,$$
$$z_{112} = 1.44, \quad z_{222} = 1.44, \quad z_{132} = 1.44,$$
$$z_{342} = 1.44;$$

$$B_{11} = 5, \quad B_{21} = 5, \quad B_{31} = 5, \quad B_{41} = 11,$$
$$B_{51} = 2, \quad B_{12} = 1, \quad B_{22} = 1, \quad B_{32} = 1;$$

where the buffer sizes are rounded up to integers. We use the three level hierarchical structure for the simulation which is described in Section 2.9. The simulation program that we use is called HIERCSIM which was developed by B. Darakananda [12]. Two part types are produced simultaneously. Fig.40 illustrates the cumulative production of Type 1 parts. The straight line is the cumulative demand. The upper curve is

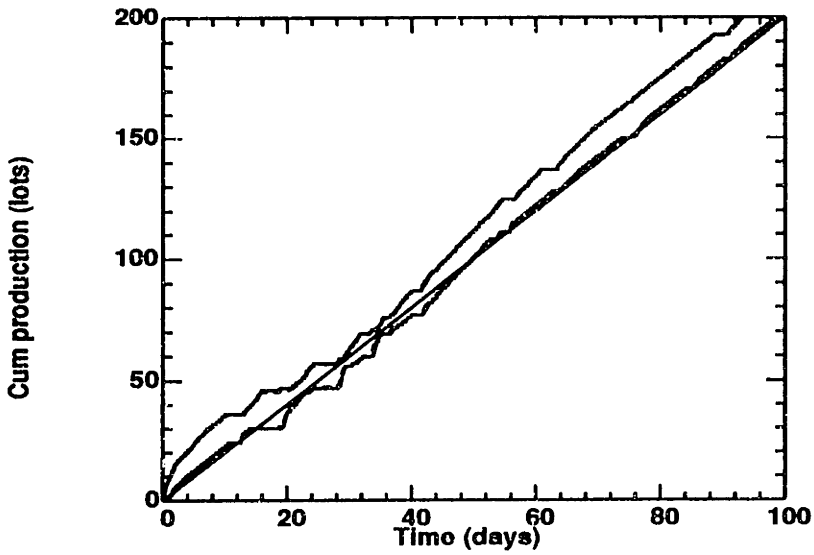

Figure 40: The simulation result of cumulative production of part type 1

the cumulative input of the raw Type 1 parts at Machine 1. The lower curve is the cumulative output of the final Type 1 products at machine 3. 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. The vertical and horizontal distance between the upper and lower curves indicate the instantaneous WIP inventory and throughput time respectively. The simulation results of Type 2 parts are shown in Fig.41.

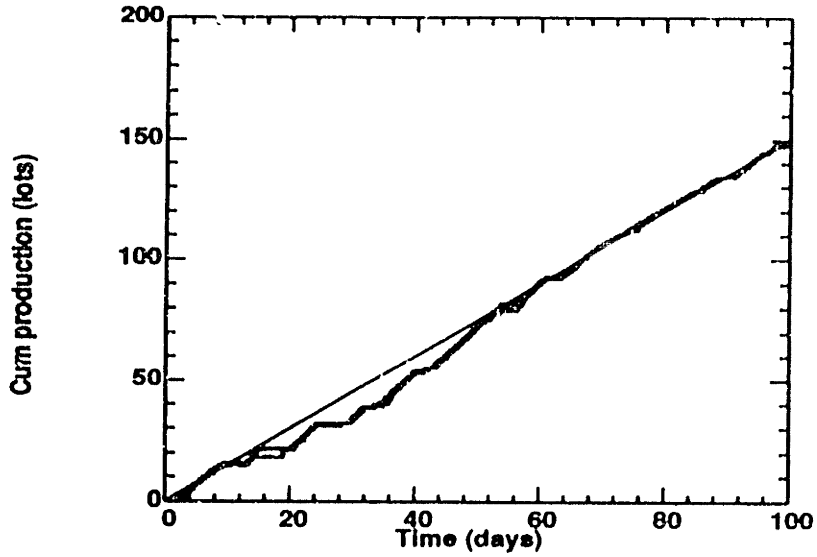

Figure 41: The simulation result of cumulative production of part type 2

10 Performance measurement

In previous sections, we developed algorithms to schedule productions in manufacturing systems. Simulation examples have shown that the algorithms work well. In this section, we establish the bounds on the level of WIP inventory and cycle time, based on the algorithms. In doing so, we first consider the N-machine, one-part-type tandem production lines. Then, we extend the results to the more general case, multiple-part-type reentrant systems.

10.1 Single-part-type production lines

Consider the N-machine, one-part-type system described in Section 5. There are N machines and $N-1$ buffers in the system. For Machine i ($i = 1, \ldots, N$), the failure rate p_i, the repair rate r_i, the processing time τ_i, and the isolated capacity D_i are given. For Buffer i ($i = 1, \ldots, N-1$), the buffer size B_i and the average buffer level $\bar{b}_i$ are determined by solving (63) and (64) in Section 5.7.3. Suppose that the initial surplus is

$$x^0 = (x_1^0, \ldots, x_N^0);$$

and the initial buffer level is

$$\begin{aligned} b^0 &= (b_1^0, \ldots, b_{N-1}^0) \\ &= (x_2^0 - x_1^0, \ldots, x_N^0 - x_{N-1}^0). \end{aligned}$$

Define

$$e_i = \bar{b}_i - b_i^0, \quad (i = 1, \ldots, N);$$

which is refered to as the *initial backlog* of Buffer i if it is positive or the *initial excess* of Buffer i if it is negative.

10.1.1 The worst case bounds on WIP inventory

The WIP inventory in the system consists of the material in buffers and the parts being processed on machines. Let W be the level of total WIP inventory in the system. Since each buffer level is a nonnegative quantity bounded by its size and since a machine can process only a lot at a time, we then have in the worst case

$$0 \leq W \leq N + \sum_{i=1}^{N-1} B_i; \tag{139}$$

where the right hand side is the total space in the buffers and the holding devices on the machines. The buffer sizes are rounded up to integers.

10.1.2 The bounds on the average WIP level

Because of the initial conditions, the system experiences an *initial delay* before it reaches steady state. For instance, if we start with a empty system, it takes a certain amount of time for the first lot to go through the system. The total work before the system reaches the steady state is refered to as the *initial work load*, which has the unit of *time*×*part*. Fig.42 illustrates the relationship between the initial delay and the average WIP level. For a given set of machine parameters and initial conditions, the shaded area represents the initial work load, which includes filling out the buffers to the average buffer levels and getting the first lot out of the system. Let I be the initial work load, which has the following bounds,

$$I_l \leq I \leq I_u; \tag{140}$$

where

$$I_l = \left[\sum_{j=1}^{N-1}(\bar{b}_j - b_j^0)\sum_{i=1}^{j}\tau_i + \sum_{i=1}^{N}\tau_i\right]^+ ; \tag{141}$$

$$I_u = \sum_{j=1}^{N-1} \mid \bar{b}_j - b_j^0 \mid \sum_{i=1}^{j}\tau_i + \sum_{i=1}^{N}\tau_i; \tag{142}$$

in which $A^+ = A$ if $A \geq 0$, and $A^+ = 0$ if $A < 0$. In the quantity I_l, we consider the cancellation between *initial excesses* and *backlogs*. Meanwhile, the initial excesses are treated as the same as the initial backlogs when determining I_u. Note that I_l and I_u are the same if the initial buffer levels are zero. Let U_i be the maximal production rate which can be achieved by Machine i, $(i = 1, \ldots, N)$. Define U_{min} the maximal rate which can be achieved by the system. It is given by

$$U_{min} = \min\{U_1, U_2, \ldots, U_N\}.$$

Let t_d be the *initial delay*. By inspecting the geometric relation in Fig.42, the initial delay t_d and the initial work load I should satisfy

$$\frac{t_d^2 d}{2} \leq I \leq \frac{t_d^2 U_{min}}{2}; \tag{143}$$

which leads to

$$\sqrt{\frac{2I}{U_{min}}} \leq T_d \leq \sqrt{\frac{2I}{d}}. \tag{144}$$

Combining (140) and (144), we have

$$\sqrt{\frac{2I_l}{U_{min}}} \leq T_d \leq \sqrt{\frac{2I_u}{d}}. \tag{145}$$

Let $\bar{W}$ be the average WIP level in the system which is given by

$$\bar{W} = \sum_{i=1}^{N-1}\bar{b}_i + t_d d, \tag{146}$$

where $\sum_{i=1}^{N-1}\bar{b}_i = \bar{x}_1$ is the vertical distance between the cumulative input at Machine 1 and the cumulative demand in Fig. 42, and $t_d d$ is the vertical distance between the cumulative demand and the cumulative output at Machine N. Eqs. (145) and (146)

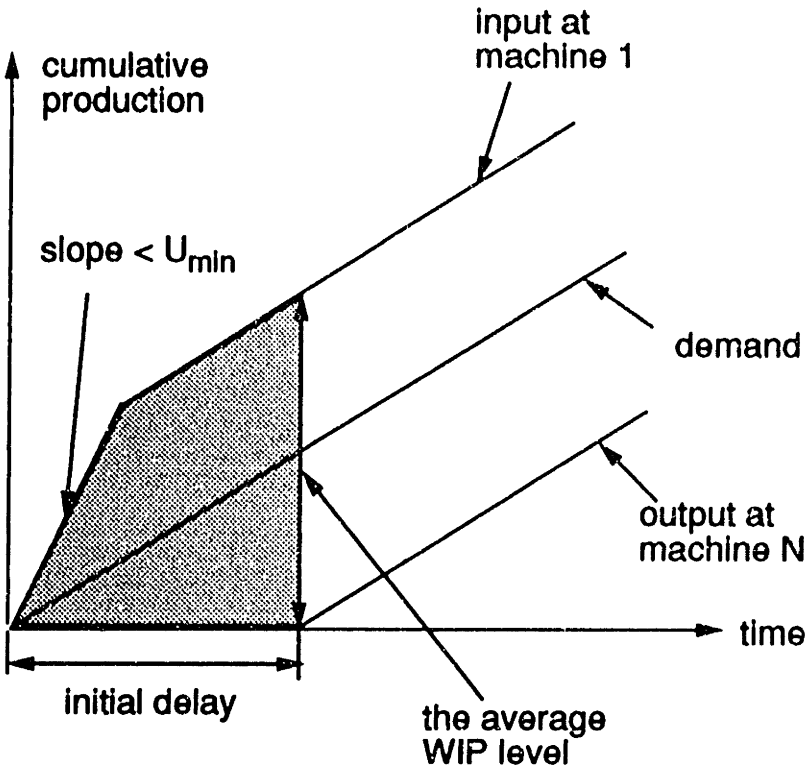

Figure 42: The average WIP level and initial delay

lead to

$$\sum_{i=1}^{N-1} \bar{b}_i + \sqrt{\frac{2I_l}{U_{min}}}d \leq \bar{W} \leq \sum_{i=1}^{N-1} \bar{b}_i + \sqrt{2I_u d}. \tag{147}$$

It is important to notice that the bounds on the average WIP level are based on a heuristic estimate of long term system behavior. It might be invalid for transient conditions or short term estimates. The average WIP depends on the initial buffer levels. That is because the system is starved or blocked so much as to have just enough capapcity to acheive the demand. Therefore, the delay caused by the initial conditions contributes to the average WIP level in steady state.

10.1.3 The bounds on the average cycle time

Each part travels through the system with an average rate d. Let T be the average *cycle time* that a part spends in the system. Then by Little's law ($\bar{W} = dT$) and

(147), we have the bounds on the average cycle time,

$$(\frac{1}{d})\sum_{i=1}^{N-1}\bar{b}_i + \sqrt{\frac{2I_l}{U_{min}}} \leq T \leq (\frac{1}{d})\sum_{i=1}^{N-1}\bar{b}_i + \sqrt{\frac{2I_u}{d}}. \tag{148}$$

10.1.4 Example

In this subsection we discuss an example to show the measurements of system performance. The five-machine, one-part-type system in Section 5.9 is used for the demonstration. In the system there are five machines and four buffers. The parameters are chosen as follows:

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

where the unit of r and p is 1/day. The unit of τ is a day. The unit of parts is a lot. The initial buffer levels are chosen to be zero.

Fig.43 illustrates the the worst case upper bound on the level of WIP inventory given by (139) where the buffer sizes are calculated according to (63)and are rounded up to integers. When the demand is small, the bound is equal to $2N-1$. This is so because for each machine, there is a holding device with a capacity of one lot and a loading/unloading area of capacity one. As the demand is increased, the buffer sizes become bigger and bigger and so is the upper bound on the WIP level.

Fig.44 depicts the bounds on the average WIP level for different demand values. The solid graph is the lower bound and the dashed curve is the upper bound bound. The discrete circles are the simulation results for different demand values, which are close to the lower bound.

10.2 Multiple-part-type reentrant systems

In the previous subsection we established the bounds on the average WIP inventory in single-part-type production lines. We extend the results to multiple-part-type reentrant systems in this subsection. Consider the system described in Section 9 which consists of N machines. M part types are produced. A Type k part needs

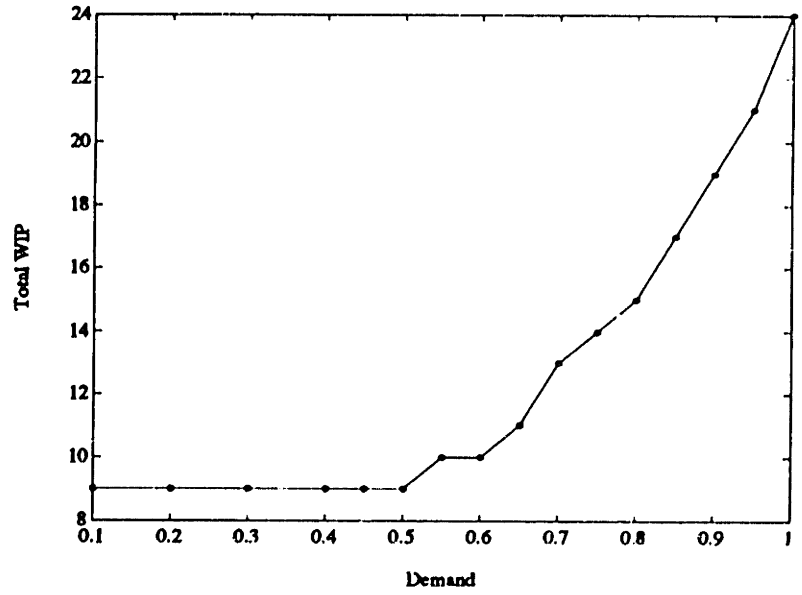

Figure 43: The worst case upper bound on WIP level

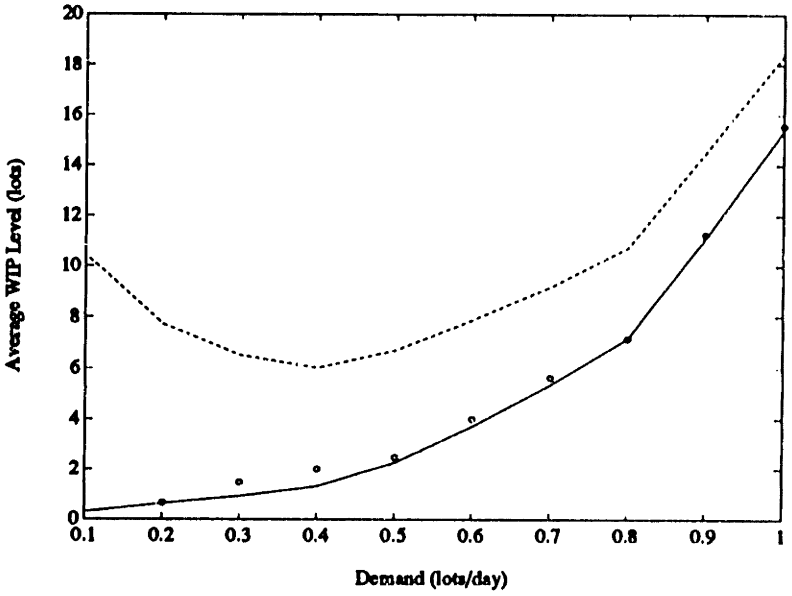

Figure 44: The bounds on the average WIP level

L_k operations ($k = 1, \ldots, M$). There are a total of $\sum_{k=1}^{M}(L_k - 1)$ buffers in the system. Using the approximate linear system method developed in Section 9, the original machines are sliced into partial machines which are arranged as single-part-type production lines. For partial machine m_{jk}, the failure rate P_{jk}, the repair rate R_{jk}, and the isolated capacity D_{jk} are determined in Section 9.7.1. The buffer sizes and the average buffer levels are given by (135) and (136). Suppose that the initial surplus is given by

$$x^0 = \{x^0_{1k}, \ldots, x^0_{L_k k}; \quad (k = 1, \ldots, M)\};$$

and the initial buffer level is

$$\begin{aligned} b^0 &= \{b^0_{1k}, \ldots, b^0_{L_k-1,k}; \quad (k = 1, \ldots, M)\} \\ &= \{x^0_{2k} - x^0_{1k}, \ldots, x^0_{L_k} - x^0_{L_k-1,k}; \quad (k = 1, \ldots, M)\}. \end{aligned}$$

10.2.1 The worst case bounds on WIP inventory

Let W be the total WIP inventory in the system. Since each buffer level is bounded by its size and a machine can process only a lot at a time, we then have in the worst case

$$0 \leq W \leq N + \sum_{k=1}^{M} \sum_{i=1}^{L_k-1} B_{ik}. \tag{149}$$

10.2.2 The bounds on the average WIP level

Let I_k be the initial work load of part type k ($k = 1, \ldots, M$), which satisfies

$$I_{lk} \leq I_k \leq I_{uk}; \tag{150}$$

where

$$I_{lk} = \left[\sum_{j=1}^{L_k-1} (\bar{b}_{jk} - b^0_{jk}) \sum_{i=1}^{j} \tau_{ik} + \sum_{i=1}^{L_k} \tau_{ik} \right]^+; \tag{151}$$

$$I_{uk} = \sum_{j=1}^{L_k-1} | \bar{b}_{jk} - b^0_{jk} | \sum_{i=1}^{j} \tau_{ik} + \sum_{i=1}^{L_k} \tau_{ik}. \tag{152}$$

Let U_{jk} be the maximal rate which can be achieved by partial machine m_{jk}. Let

$$U_{mink} = \min\{U_{1k}, U_{2k}, \ldots, U_{L_k k}\}, \quad (k = 1, \ldots, M);$$

which is the maximal rate which can be achieved by the approximated linear system k. Let $\bar{W}_k$ be the Type k average WIP inventory in the system. By applying the results of single-part-type production lines in Section 10.1 to approximate linear system k, the Type k average WIP inventory is bounded by

$$\sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{lk}}{U_{mink}}} d_k \leq \bar{W}_k \leq \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{2I_{uk} d_k}, \quad (k = 1, \ldots, M). \tag{153}$$

Let $\bar{W}$ be the total average WIP level in the system which satisfies

$$\sum_{k=1}^{M} \{ \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{lk}}{U_{mink}}} d_k \} \leq \bar{W} \leq \sum_{k=1}^{M} \{ \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{2I_{uk} d_k} \}. \tag{154}$$

10.2.3 The bounds on the average cycle time

A Type k part travels through the system with an average rate d_k. Let T_k be the average cycle time that a type k $(k = 1, \ldots, M)$ part spends in the system. It is bounded as follows:

$$(\frac{1}{d_k}) \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{lk}}{U_{mink}}} \leq T_k \leq (\frac{1}{d_k}) \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{uk}}{d_k}}, \quad (k = 1, \ldots, M). \tag{155}$$

Let T be the average cycle time which is determined by taking the average over part types

$$T = \frac{1}{M} \sum_{k=1}^{M} T_k.$$

The average cycle time is bounded as follows:

$$\frac{1}{M} \sum_{k=1}^{M} \{ (\frac{1}{d_k}) \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{lk}}{U_{mink}}} \} \leq T \leq \frac{1}{M} \sum_{k=1}^{M} \{ (\frac{1}{d_k}) \sum_{i=1}^{L_k-1} \bar{b}_{ik} + \sqrt{\frac{2I_{uk}}{d_k}} \}. \tag{156}$$

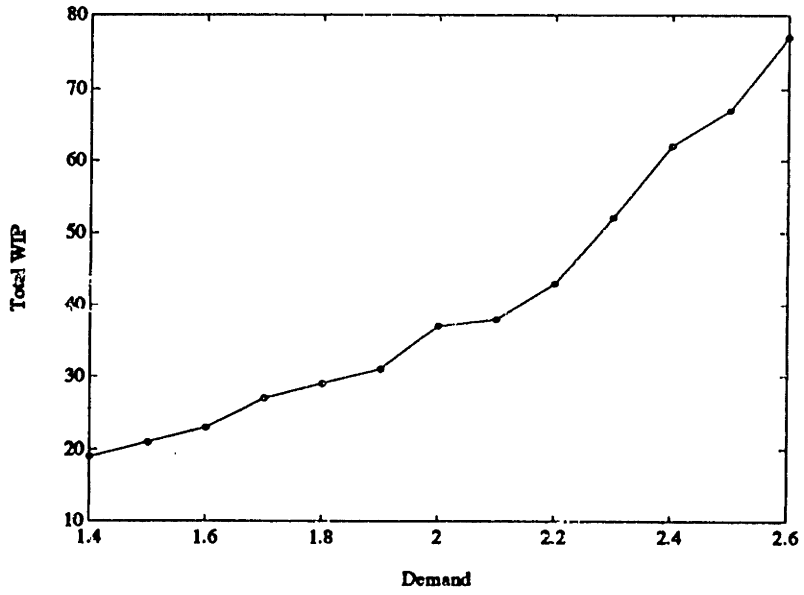

Figure 45: The worst case upper bound on WIP vs Demand d_1

10.2.4 Example

As an example, we calculate the bounds on WIP level for the three-machine, two-part-type reentrant system in Section 9.9. The system consists of three machines and eight buffers (see Fig. 39). The system parameters are listed in Section 9.9. All initial buffer levels are set to be zero. Fig.45 illustrates the worst case upper bound on the WIP level for different values of d_1 while d_2 is fixed to be 1.5 (lots/day). The bound becomes bigger as the demand is increased. Fig.46 shows the bounds on the average WIP level for defferent values of d_1 while d_2 is fixed to be 1.5 (lots/day). The solid graph is the lower bound and the dashed curve is the upper bound. The discrete circles are the simulation results which are close to the lower bound.

10.3 Summary

In this section, we established bounds on the average WIP inventory and the average cycle time based on the production flow control policy developed in previous sections. From the results of simulation, the lower bound on the average WIP is also a good indicator of the average level of WIP inventory. As an application of the production control algorithm, in the following section, we simulate the production control of a wafer fabrication facility — the MIT Integrated Circuit Laboratory.

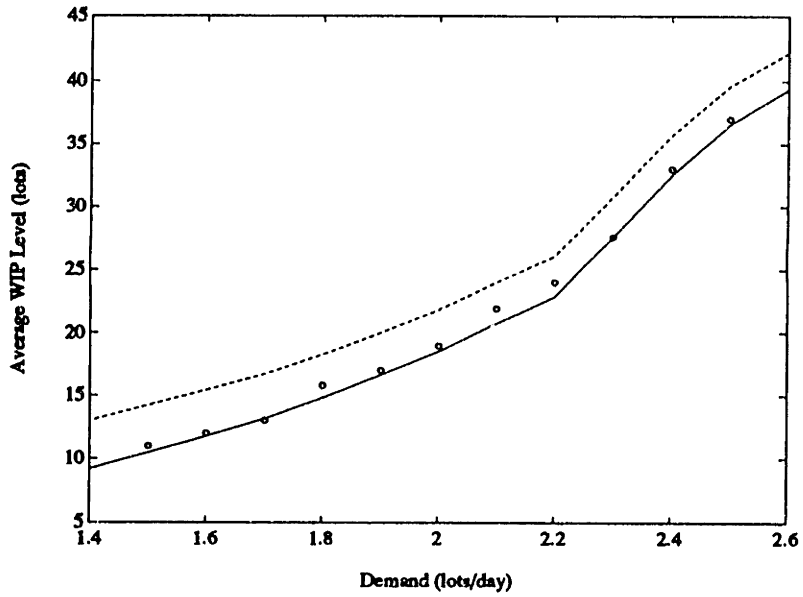

Figure 46: The bounds on the average WIP vs Demand d_1

11 Simulation of wafer fabrication production

In the previous sections, we developed a scheduling algorithm for real-time scheduling of manufacturing systems. In this section, as an application, we conduct digital simulations of the production control of wafer fabrication processes at the MIT Integrated Circuit Laboratory.

11.1 The MIT CMOS process

In the Integrated Circuit Laboratory of MIT, there is a *baseline* process, a 1.75 micron CMOS process, which is used to monitor equipment performance and device characteristics. The baseline process is enhancement-compatible so that new technology innovations can be tested in a real integrated circuit process. The process was designed modularly such that coupling between processing steps was minimized. Whenever possible, these baseline processing steps are incorporated into new processes and experiments.

In order to use the model of Section 3, we assume that all inspection operations in the baseline process are not restrictive. That is, none of the wafers fails the inspections. We only count the time for inspection and add it to the processing time of the preceding operation.

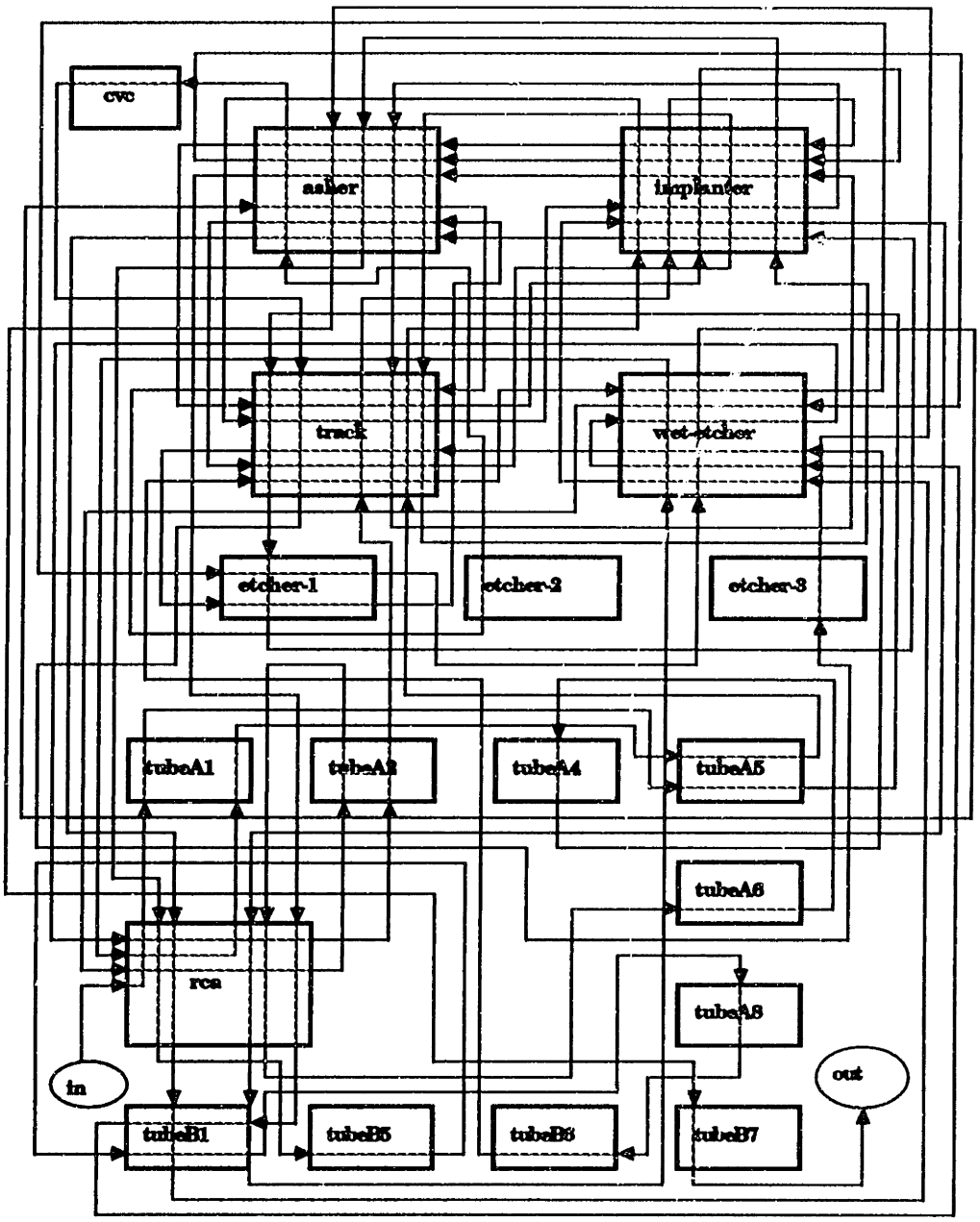

Figure 47: The MIT CMOS process

No.	machine	r	p	No.	machine	r	p
1	asher	0.5	0.01	11	tube-a4	0.5	0.002
2	cvc	0.5	0.001	12	tube-a5	0.33	0.017
3	etcher-1	0.5	0.01	13	tube-a6	0.33	0.017
4	etcher-2	0.5	0.01	14	tube-a8	0.33	0.017
5	etcher-3	0.5	0.01	15	tube-b1	0.5	0.002
6	implanter	0.5	0.01	16	tube-b5	0.5	0.002
7	photo-track	0.33	0.02	17	tube-b6	0.5	0.002
8	RCA	0.5	0.01	11	tube-b7	0.5	0.002
9	tube-a1	0.5	0.001	19	wet-etcher	0.5	0.001
10	tube-a2	1.0	0.002				

Table 5: The machine parameters for the CMOS process

Fig.47 illustrates the route of the baseline CMOS process which consists of 73 operations. A wafer is released into the system at the RCA station and leaves the system at Tube-b7 after the process is completed. There are 19 machines involved in the CMOS production. The machine parameters are listed in Table 5. The operations and the processing times of the CMOS process are listed in Table 6. The unit of processing time τ is a day. For a given demand 0.15 lots/day, the components of the hedging point are calculated according to the algorithm developed for single-part-type reentrant systems in Section 8 and are listed in Table 7. The unit of the components of the hedging point is a lot. There are 72 buffers in the system which are located between every two consecutive operations. The buffer sizes are rounded up to integers and listed in Table 8.

Fig.48 illustrates the simulation results of the cumulative production. The straight line is the cumulative demand. The upper curve is the cumulative input of the raw wafers at the RCA station. The lower graph is the cumulative output of the final products at Tube-a7. The vertical and horizontal distance between the upper and lower curves indicate the instantaneous WIP inventory and throughput time respectively.

11.2 A multi-process example

In this subsection we demonstrate a two-process example (Fig. 49). The first process is called poly-gate capacitor process which consists of seventeen operations and the

Operation	machine	τ	Operation	machine	τ
1	RCA	0.25	38	implanter	0.188
2	tube-a1	0.656	39	asher	0.188
3	tube-a5	0.344	40	wet-etcher	0.073
4	photo-track	0.448	41	RCA	0.25
5	etcher-1	0.156	42	tube-a2	0.5
6	implanter	0.188	43	RCA	0.25
7	asher	0.188	44	tube-a6	0.438
8	RCA	0.25	45	tube-a4	0.448
9	tube-b1	0.969	46	wet-etcher	0.063
10	wet-etcher	0.063	47	photo-track	0.448
11	implanter	0.188	48	etcher-1	0.156
12	RCA	0.25	49	asher	0.188
13	tube-b1	1.125	50	photo-track	0.375
14	wet-etcher	0.073	51	implanter	0.188
15	RCA	0.25	52	asher	0.188
16	tube-a1	0.656	53	photo-track	0.448
17	tube-a5	0.344	54	implanter	0.188
18	photo-track	0.448	55	asher	0.188
19	implanter	0.156	56	RCA	0.25
20	photo-track	0.375	57	tube-a5	0.344
21	implanter	0.156	58	tube-b1	0.156
22	asher	0.188	59	tube-a8	0.406
23	photo-track	0.448	60	tube-b6	0.25
24	implanter	0.188	61	photo-track	0.156
25	asher	0.188	62	wet-etcher	0.031
26	RCA	0.25	63	etcher-1	0.156
27	tube-b1	1.0	64	wet-etcher	0.031
28	wet-etcher	0.167	65	asher	0.188
29	wet-etcher	0.052	66	photo-track	0.448
30	RCA	0.25	67	etcher-2	0.156
31	tube-a2	0.5	68	asher	0.188
32	photo-track	0.448	69	cvc	0.344
33	implanter	0.188	70	photo-track	0.448
34	implanter	0.188	71	etcher-3	0.156
35	asher	0.188	72	asher	0.188
36	photo-track	0.448	73	tube-b7	0.25
37	implanter	0.188			

Table 6: The operations and processing times of the CMOS process

Operation	hedging	Operation	hedging	Operation	hedging
1	6.259	26	5.032	51	3.014
2	6.259	27	5.032	52	3.014
3	6.259	28	5.032	53	3.014
4	6.259	29	5.032	54	3.014
5	6.259	30	5.032	55	3.014
6	6.259	31	4.086	56	3.014
7	6.259	32	4.086	57	3.014
8	6.259	33	4.086	58	3.014
9	6.259	34	4.086	59	3.014
10	6.259	35	4.086	60	1.778
11	6.259	36	4.086	61	1.778
12	6.259	37	4.086	62	1.778
13	6.259	38	4.086	63	1.778
14	6.259	39	4.086	64	1.778
15	6.259	40	4.086	65	1.778
16	6.259	41	4.086	66	1.778
17	6.259	42	4.086	67	1.778
18	5.521	43	4.086	68	1.778
19	5.032	44	4.086	69	1.778
20	5.032	45	3.014	70	1.778
21	5.032	46	3.014	71	1.778
22	5.032	47	3.014	72	1.778
23	5.032	48	3.014	73	1.778
24	5.032	49	3.014		
25	5.032	50	3.014		

Table 7: The components of the hedging point

No.	buffer size	No.	buffer size	No.	buffer size
1	1	25	1	49	1
2	1	26	1	50	1
3	1	27	1	51	1
4	2	28	1	52	1
5	1	29	1	53	3
6	1	30	1	54	1
7	1	31	1	55	1
8	1	32	1	56	1
9	1	33	1	57	1
10	1	34	1	58	1
11	1	35	1	59	2
12	1	36	2	60	1
13	1	37	1	61	1
14	1	38	1	62	1
15	1	39	1	63	1
16	1	40	1	64	1
17	1	41	1	65	1
18	1	42	1	66	1
19	1	43	1	67	1
20	1	44	2	68	1
21	1	45	1	69	1
22	1	46	1	70	1
23	1	47	1	71	1
24	1	48	1	72	1

Table 8: The buffer sizes

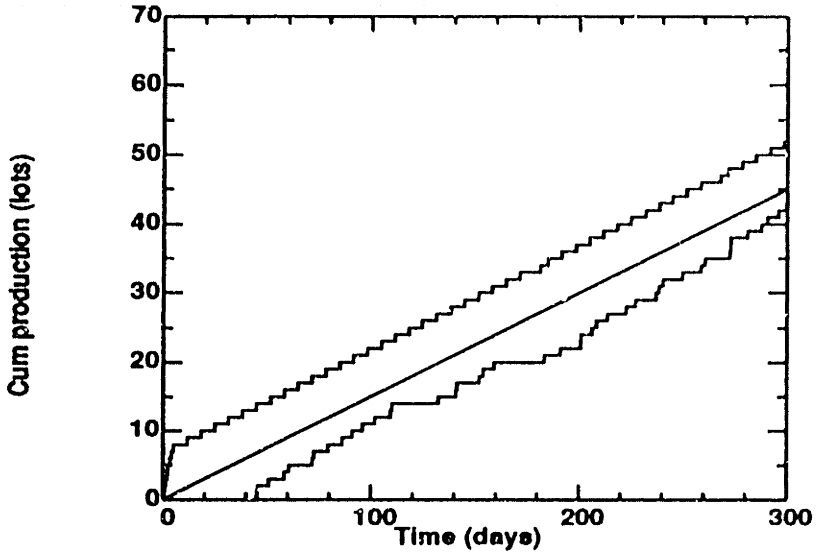

Figure 48: The simulation result of the cumulative production of the CMOS process

other is called poly-monitor process which has seven operations. The *Type 1* wafers follow the poly-gate capacitor process: RCA, Buffer (2,1), Tube-b1, Buffer (2,2), Photo-track, Buffer (2,3), Wet-etcher, Buffer (2,4), Tube-a6, Buffer (2,5), RCA, Buffer (2,6), Tube-a1, Buffer (2,7), Asher, Buffer (2,8), Plasma, Buffer (2,9), Wet-etcher, Buffer (2,10), Asher, Buffer (2,11), Tube-a4, Buffer (2,12), wet-etcher, Buffer (2,13), Photo-track, Buffer (2,14), Plasma, Buffer (2,15), Asher, Buffer (2,16), Tube-a7. The *Type 2* wafers go through the system following the poly-monitor process: RCA, Buffer (1,1), Tube-b1, Buffer (1,2), Tube-b6, Buffer (1,3), Tube-a4, Buffer (1,4), Wet-etcher, Buffer (1,5), Photo-track, Buffer (1,6), Plasma. A total of ten machines are involved in the production procedure. The machine parameters are listed in Table 9 where the unit of r and p is 1/day. The routing information and processing times are in Tables 10 and 11. Given that the demand is 0.5 lots/day for the poly-monitor process and 0.6 lots/day for the poly-gate capacitor process, the components of the hedging point are calculated and listed in Tables 10 and 11 as well. The buffer sizes are rounded up to integers and listed in Table 12.

Figure 50 illustrates the simulation results of the cumulative production of the poly-monitor process. The straight line is the cumulative demand. The upper curve is the cumulative input of the raw wafers at the RCA station. The lower graph is the cumulative output of the final products at Tube-a7. The vertical and horizontal dis-

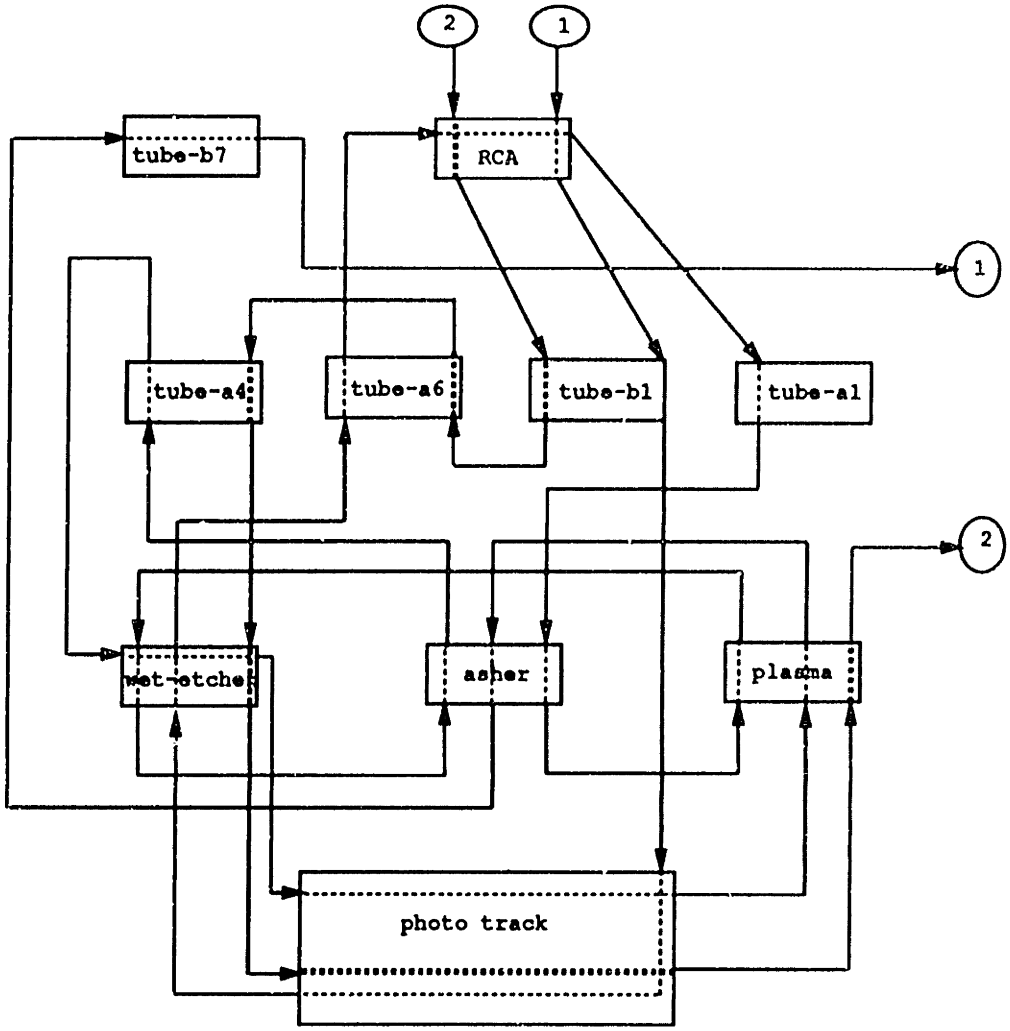

Figure 49: The two-process system

No.	machine	r	p	No.	machine	r	p
1	asher	0.5	0.01	6	tube-a4	0.5	0.002
2	plasma	0.33	0.02	7	tube-a6	0.33	0.017
3	photo-track	0.33	0.01	8	tube-b1	0.5	0.002
4	RCA	0.5	0.01	9	tube-b7	0.5	0.002
5	tube-a1	0.5	0.001	10	wet-etcher	0.5	0.001

Table 9: The machine parameters for the two-process example

Operation	machine	τ	hedging	Operation	machine	τ	hedging
1	RCA	0.25	2.443	10	wet-etcher	0.031	2.443
2	tube-b1	0.969	2.443	11	asher	0.188	2.443
3	tube-a6	0.469	2.443	12	tube-a4	0.469	2.443
4	wet-etcher	0.031	2.443	13	wet-etcher	0.063	0.91
5	tube-a6	0.469	2.443	14	photo-track	0.469	0.91
6	RCA	0.25	2.443	15	plasma	0.156	0.91
7	tube-a1	0.469	2.443	16	asher	0.188	0.91
8	asher	0.188	2.443	17	tube-b7	0.25	0.91
9	plasma	0.156	2.443				

Table 10: The processing times and hedging components of the poly-gate capacitor process

Operation	machine	τ	hedging	Operation	machine	τ	hedging
1	RCA	0.25	0.715	5	wet-etcher	0.063	0.715
2	tube-b1	0.656	0.715	6	photo-track	0.469	0.715
3	tube-a6	0.469	0.715	7	plasma	0.156	0.715
4	tube-a4	0.469	0.715				

Table 11: The processing times and hedging components of the poly-monitor process

Name	size	Name	size	Name	size
Buffer (1,1)	1	Buffer (1,9)	1	Buffer (2,1)	1
Buffer (1,2)	4	Buffer (1,10)	1	Buffer (2,2)	5
Buffer (1,3)	1	Buffer (1,11)	1	Buffer (2,3)	1
Buffer (1,4)	3	Buffer (1,12)	2	Buffer (2,4)	1
Buffer (1,5)	1	Buffer (1,13)	1	Buffer (2,5)	1
Buffer (1,6)	1	Buffer (1,14)	1	Buffer (2,6)	1
Buffer (1,7)	1	Buffer (1,15)	1		
Buffer (1,8)	1	Buffer (1,16)	1		

Table 12: The buffer sizes

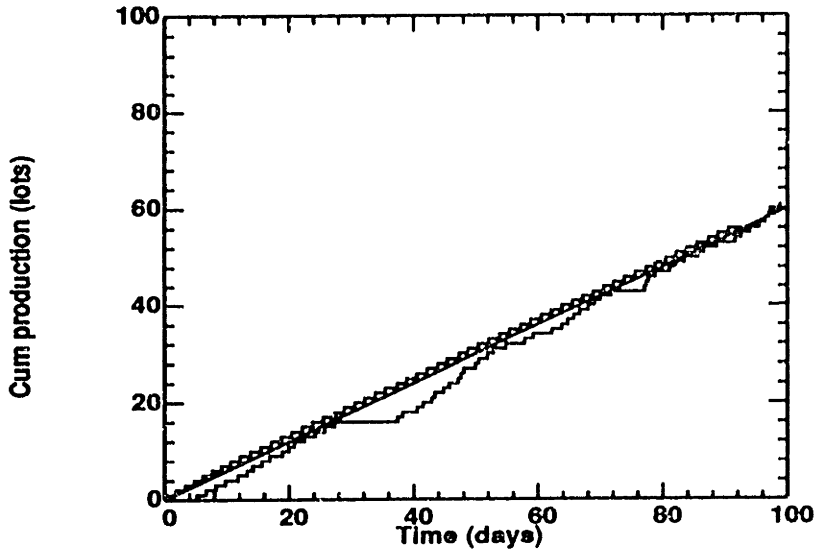

Figure 50: The simulation result of the cumulative production of the poly-monitor process

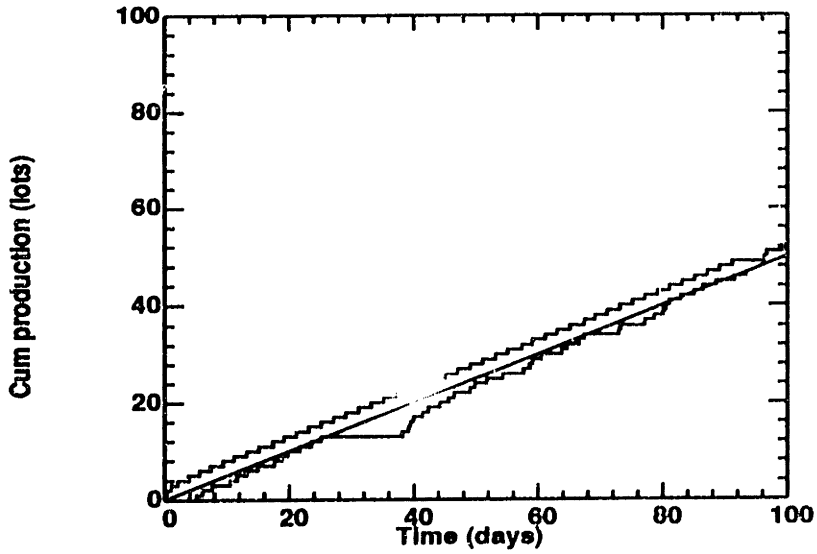

Figure 51: The simulation result of the cumulative production of the poly-gate capacitor process

tance between the upper and lower curves indicate the instantaneous WIP inventory and throughput time respectively. Similarly the simulation results of the cumulative production of the poly-gate capacitor process is shown in Figure 51. The two processes are performed simultaneously.

11.3 Summary

In this section we performed simulations of the MIT CMOS Process and a multiple process production in the MIT Integrated Circuit Laboratory, using the scheduling algorithm developed in previous sections. The simulation results show that the algorithm works very well for complex manufacturing systems such as the wafer fabrication facilities. The scheduling algorithm promises a much lower WIP inventory level and shorter cycle time than those in the practice of today's semiconductor fabrication industry.

No.	name	type
1	machine_name	string
2	status	integer
3	comment	string
4	mttf	time_duration
5	mttr	time_duration
6	maint_time	time_duration
7	control	integer
8	batchsize	integer
	recipes	list_of_recipes
10	schedules	list_of_schedules

Table 13: Data object: Machine

12 Some implementation issues

The implementation of the algorithm developed in previous sections has been under way in the MIT Integrated Circuit Laboratory since the beginning of 1991. In this section we discuss some basic issues invovled in the implemenetation such as the database schema and the I/O interface.

12.1 Data structures

The first and most important task in the development of a real-time production scheduling software is the database schema design. That is, we need to construct data structures so as to store and retrieve information efficiently in a database. The Database Management System used for the real-time scheduler is called GESTALT which is an object-oriented DBMS developed by Michael Hytens at MIT [21]. Since the database is shared by many applications, any database schema change will cause changes in the related applications. The data objects constructed for the real-time production controller are machine, recipe, process, operation, request, and schedule. As an example, the attributes of the data object "machine" are listed in Table 13. A complete list of the entries of the data objects is given in Appendix B.

12.2 The I/O interface

The user interface with the scheduler is through a form editor called *Fabform* [22]. This interface allows the user to deal with a syntax that is represented as a form. Many other applications in the Computer-Aided-Fabrication-Environment (CAFE) system of MIT including the menu and the machine operation utilities also use the same interface. The interface allows the system manager to specify or change any information like job priority and the lab users to register their requests. The scheduler, as it exists now, requires the system manager to provide a description of the lab (including machines, processes, recipes, and operations). This can be done by filling forms that have entries for all information needed by the scheduler. In the future, all or most of this data will be taken directly from the database based on the description of the lab by other applications like the Process-Flow-Representation (PFR) [27]. Fig. 52 illustrates the information that the user needs to provide for the scheduler, when he or she registers a new request.

12.3 Overall design of the scheduler

Fig. 53 illustrates the overall design of the scheduling system. The core consists of the scheduling algorithm module which is a collection of C programs. One such program calculates the rates of controllable events (production rates, maintenance rates, etc.) for a given lab capacity. This is triggered every time a machine stoppage occurs. These rates are then fed to other programs that determine the lot to be loaded. Thus the individual programs constituting the scheduler are arranged in a hierarchy parallel to the hierarchy explained in Section 4.8. One or more of these programs are triggered off by events in the lab like machine failures, job cpmpletion, etc. These programs in turn may trigger other programs till a tentative schedule for the lab machines is obtained in response to the triggering events.

The information flow between the scheduler, the database, the system manager, and the lab user are indicated by directional arrows in the graph. The scheduler's decision can be partly or completely overridden by the system manager at any time. The output from the scheduler can be easily used by other applications in CAFE [27] like the Machine Reservation System to make decisions about lab usage.

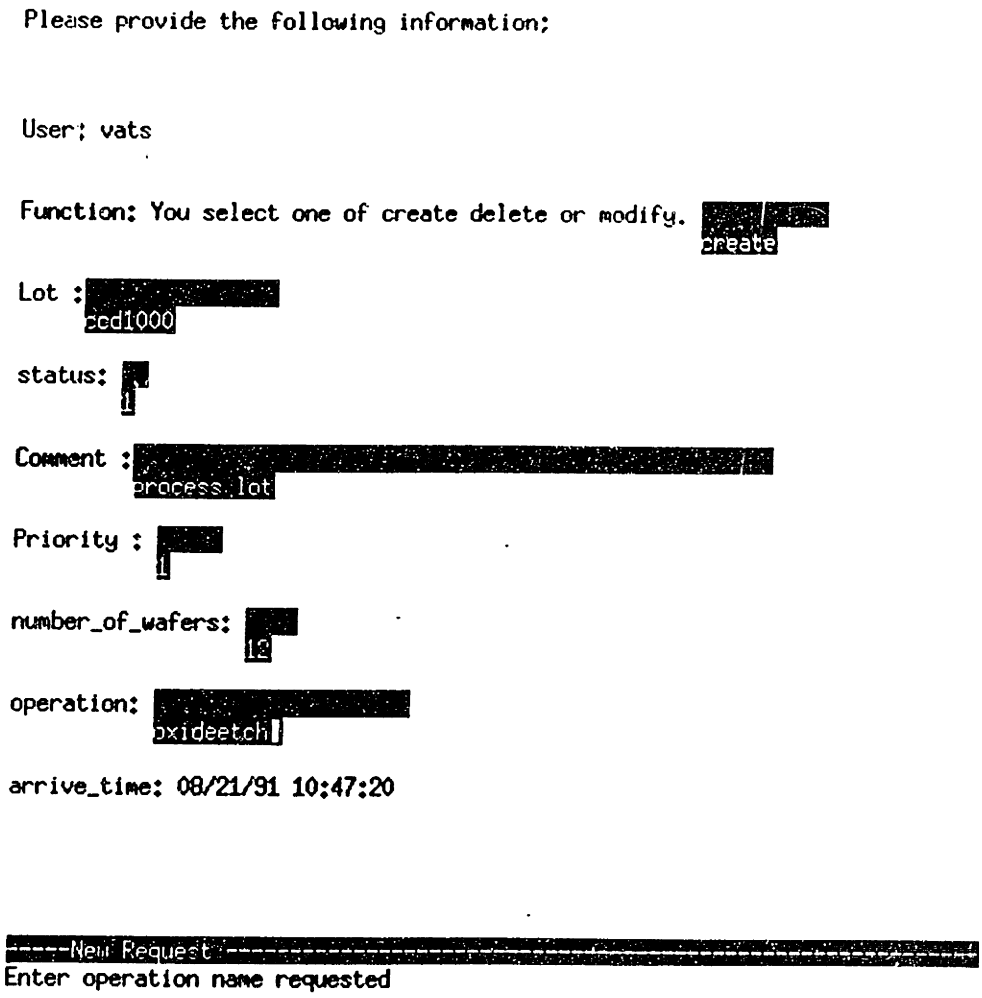

Figure 52: Creating a new request for the scheduler

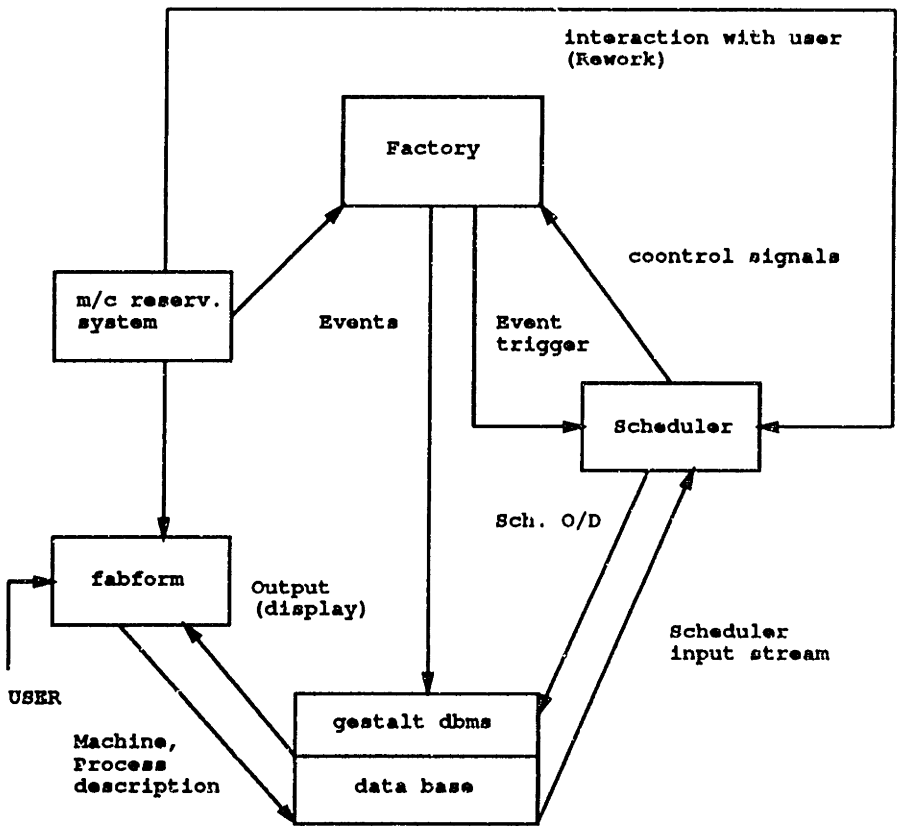

Figure 53: Information flow for the scheduler

12.4 Summary

In this section, we discussed some implementation issues involved in real-time scheduling a wafer fabrication. While the hierarchical framework is still undergoing development, it is experimentally applied to the management of the MIT Integrated Circuit Laboratory. Current research issues involve the analytic formulation and solution of a variety of problems such as setup change and rework, as well as software implementation.

13 Summary

A real-time feedback control algorithm is developed for scheduling manufacturing systems. The WIP inventory is allocated such that the system achieves a given demand in an efficient way. The major contribution of this thesis is that we explicitly introduce buffer sizes, average buffer levels, and starvation and blockage fraction as control parameters for the long term capacity planning for a given manufacturing system. A frequency-duration technique is developed to estimate the control parameters such that the average WIP inventory and the final product inventory are kept at low levels. The level of WIP inventory is involved in three phases of the decision making procedure. For the long term capacity planning, the buffer sizes and average buffer levels are allocated so as to have enough system capacity to achieve the demand. When a demand or machine parameter changes, we recompute the WIP allocation. The scheduling system recalculates the production rates in real-time whenever a machine fails or is starved or blocked. For selecting the actual loading times, starvation and blockage are also a concern of the decision making. When the system reaches the steady-state, the algorithm generates the same policy as in a KANBAN system. That is, the production rates are equal to the demand. Moreover, if the system drifts away from the steady-state due to random events such as machine failures and starvation or blockage, real-time policy changes are made by the algorithm such that the system recovers as soon as possible.

The simulation results verify that the algorithm works very well for complex manufacturing systems such as the wafer fabrication facilities. Based on the algorithms, the bounds on the average WIP inventory are established for system performance measurement.

The algorithm is modified in [4] such that the formulation for buffer hedging levels and spaces becomes linear. In many cases, the linear formulation gives us equally good results without the trouble of solving the nonlinear programming problem. It appears that the algorithms developed here can also be extended to more general systems in which the activities such as setup changes and multiple routing take place. They are possible topics for future research.

References

[1] P. Afentakis, B. Gavish, and U. Karmarkar. Computationally Efficient Optimal Solutions to the Lot-sizing Problem in Multi-stage Assembly Systems. *Management Science*, 30(2):222–239, 1984.

[2] R. Akella, Y. F. Choong, and S. B. Gershwin. Performance of Hierarchical Production Scheduling Policy. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, CHMT-7(3), 1984.

[3] R. Akella and P. R. Kumar. Optimal Control of Production Rate in a Failure Prone Manufacturing System. *IEEE Transaction on Automatic Control*, AC-31(2):116–126, 1986.

[4] S. X. Bai. Linear Formulation of Control Parameter Estimation for Real-Time Production Scheduling. Master's thesis, MIT, 1991.

[5] S. X. Bai and S. B. Gershwin. A Manufacturing Scheduler's Perspective on Semiconductor Fabrication. Technical report, MIT, Laboratory for Manufacturing and Productivity, VLSI-89-518, 1989.

[6] S. X. Bai, Srivatsan, and S. B. Gershwin. Hierarchical real-time scheduling of a semiconductor fabrication facility. *IEEE/CHMT IEMT (International Electronics Manufacturing Technology) Symposium, Washington, DC*, October, 1990.

[7] G. R. Bitran, E. A. Haas, and A. C. Hax. Hierarchical Production Planning: A Single-Stage System. *Operations Research*, 29(4):717–743, 1981.

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

[9] 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*, 65(4), 1986.

[10] H. Chen, M. Harrison, A. Mandelbaum, A. Van Ackere, and L. Wein. Empirical Evaluation of A Queueing Network Model for Semiconductor Wafer Fabrication. *Operations Research*, 36(2), 1988.

[11] R. Conway, W. Maxwell, J. O. McClain, and L. J. Thomas. The Role of Work-In-Process Inventory In Serial Production Lines. *Operations Research*, 36(2), 1988.

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

[13] M. A. H. Dempster, Lp. Jansen M. L. Fisher, B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan. Analytical Evaluation of Hierarchical Planning Systems. *Operations Research*, 29(4):707–716, 1981.

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

[15] 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*, 77(1):195–209, 1989.

[16] S. B. Gershwin, R. Akella, and Y. F. Choong. Short-Term Production Scheduling of an Automated Manufacturing Facility. *IBM Journal of Research and Development*, 29(4):392–400, 1985.

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

[18] S. C. Graves. A Review of Production Scheduling. *Operations Research*, 29(4):646–675, 1981.

[19] S. C. Graves. Using Lagrangean Relaxation Techniques to Solve Hierarchical Production Planning Problems. *Management Science*, 28(3):260–275, 1982.

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

[21] M. L. Heytens and R. S. Nikhil. GESTALT: An Expressive Database Programming System. Technical Report VLSI Memo Series, 88-484, MIT, 1988.

[22] R. Jayavant. A User/Programmer Guide to the Fabform User Interface. Technical Report VLSI Memo Series, 88-487, MIT, 1988.

[23] J. Kimemia and S. B. Gershwin. An Algorithm for the Computer Control of a Flexible Manufacturing System. *IIE Transactions*, 15(4):353–362, 1983.

[24] B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan. Job-Shop Scheduling by Implicit Enumeration. *Management Science*, 24(4):441–450, 1977.

[25] B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan. A General Bounding Scheme for the Permutation Flow-Shop Problem. *Operations Research*, 26(1):53–67, 1978.

[26] R. C. Leachman. Preliminary Design and Development of A Corporate-Level Production Planning System for the Semiconductor Industry. Technical Report ORC#:86-11, University of California at Berkeley, 1986.

[27] M. B. MacIlrath and D. E. Troxel. CAFE: The MIT Computer Aided Fabrication Environment. *IEMT Proceedings* , October, 1990.

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

[29] O. Z. Maimon and S. B. Gershwin. Dynamic Scheduling and Routing for Flexible Manufacturing Systems that have Unreliable Machines. *Operations Research*, 36(2):279–292, 1988.

[30] E. F. P. Newson. Multi-Item Lot Size Scheduling by Heuristic, Part I: With Fixed Resources. *Management Science*, 21(10):1186–1193, 1975.

[31] E. F. P. Newson. Multi-Item Lot Size Scheduling by Heuristic, Part II: With Variable Resources. *Management Science*, 21(10):1194–1203, 1975.

[32] C. H. Papadimitriou and P. C. Kannelakis. Flowshop Scheduling with Limited Temporary Storage. *Journal of the ACM*, 27(3), 1980.

[33] R. Rishel. Dynamic Programming and Minimum Principles for Systems with Jump Markov Disterburces. *SIAM Journal on Control*, 13(2), 1975.

[34] A. Sharifnia. Production Control of a Manufacturing System with Multiple Machine States. *IEEE Transactions on Automatic Control*, AC-33(7):620-625, 1988.

[35] J. Tsitsiklis. Convexity and Characterization of Optimal In a Dynamic Routing Problem. Technical Report LIDS-R-1178, MIT, 1982.

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

[37] H. M. Wagner and T. M. Whitin. Dynamic Version of the Economic Lot Size Model. *Management Science*, 5(1):89–96, 1958.

[38] L. M. Wein. Scheduling Semiconductor Wafer Fabrication. Technical report, Stanford University, 1987.

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

Appendix A

In this appendix, we look at some properties of the optimal policies for scheduling the manufacturing systems described in Section 3. To minimize the expected total cost, we wish to specify a production policy u that satisfies

$$\min_{u\in\Omega(\alpha)} E\{\int_{t_0}^{T} g[x(s), b(x(s))]ds \mid x(t_0) = x, \alpha(t_0) = \alpha\}$$

subject to:

$$\sum_{\{j,k|\theta_{ijk}=1\}} \tau_{jk}u_{jk} \leq \alpha_i, \quad (i = 1, \ldots, N);$$

$$u_{jk} \geq 0, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M);$$

$$\dot{x}_{jk} = u_{jk} - d_k, \quad (j = 1, \ldots, L_k; k = 1, \ldots, M);$$

$$\dot{b}_{jk} = u_{jk} - u_{j+1,k}, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M);$$

$$B_{jk} \geq b_{jk} \geq 0, \quad (j = 1, \ldots, L_k - 1; k = 1, \ldots, M);$$

where $g(x, b)$ is a convex function which satisfies

$$\lim_{\|x\|\to\infty} g(x, b) = \infty, \quad \lim_{\|b\|\to\infty} g(x, b) = \infty, \quad \text{and} \quad g(0, 0) = 0.$$

Define the "cost-to-go" as

$$J(x, \alpha, t) = E\{\int_{t}^{T} g[x(s), b(x(s))]ds \mid x(t) = x, \alpha(t) = \alpha\}.$$

The cost-to-go is thus the expected total penalty incurred by the controller for the remaining time, given that the surplus and machine states are x and α at time t. Let $u^0(x, \alpha, t)$ be the optimal production policy and $J^0(x, \alpha, t)$ be the optimal cost-to-go, which minimize the expected total cost. Let $\Omega(\alpha)$ be the constraint set of the optimization problem above.

Proposition 1 *Wherever the optimal cost-to-go $J^0(x, \alpha, t)$ is differentiable with respect to x and t, the optimal production policy $u^0(x, \alpha, t)$ satisfies the following linear program:*

$$\min_{u\in\Omega(\alpha)} \frac{\partial J^0}{\partial x} u.$$

Proof [23]: For any $\delta t > 0$,

$$\min_{u\in\Omega(\alpha)}\{J^0[x(t),\alpha(t),t]\} = \min_{u\in\Omega(\alpha)} E\{\int_t^{t+\delta t} g[x(s),b(x(s))]ds$$

$$+J^0[x(t+\delta t),\alpha(t+\delta t),t+\delta t]\}.$$

For small δt, this becomes, approximately,

$$\min_{u\in\Omega(\alpha)}\{J^0[x(t),\alpha(t),t]\} = \min_{u\in\Omega(\alpha)}\{g[x(t),b(x(t))]\delta t$$

$$+\sum_{\beta\neq\alpha(t)}\lambda_{\alpha\beta}\delta t J^0[x(t+\delta t),\beta,t+\delta t]$$

$$+(1+\lambda_{\alpha\alpha}\delta t)\{J^0[x(t),\alpha(t),t]$$

$$+\frac{\partial J^0[x(t),\alpha(t),t]}{\partial x}\dot{x}\delta t + \frac{\partial J^0[x(t),\alpha(t),t]}{\partial t}\}\}.$$

By letting δt go to zero and rearranging terms, this becomes

$$0 = \min_{u\in\Omega(\alpha)}\{g[x(t),b(x(t))] + \frac{\partial J^0}{\partial x}(u-d) + \frac{\partial J^0}{\partial t} + \sum_\beta \lambda_{\alpha\beta}J^0[x(t),\beta,t]\}.$$

Note that there is only one term in this equation which depends upon u. The equation can be rewritten as

$$0 = \{g[x(t),b(x(t))] - \frac{\partial J^0}{\partial x}d + \frac{\partial J^0}{\partial t} + \sum_\beta \lambda_{\alpha\beta}J^0[x(t),\beta,t]\} + \min_{u\in\Omega(\alpha)}\frac{\partial J^0}{\partial x}u.$$

Therefore, the optimal control policy u^0 should satisfy

$$\min_{u\in\Omega(\alpha)}\frac{\partial J^0}{\partial x}u.$$

Q.E.D.

Appendix B

The following is a list of the attributes of the data objects designed for the real-time scheduler of the MIT Integrated Circuit Laboratory.

- **machine:**
  1. **name:** string
  2. **status:** integer
  3. **comment:** string
  4. **mttf:** time_duration
  5. **mttr:** time_duration
  6. **maint_time:** time_duration
  7. **control:** integer
  8. **batchsize:** integer
  9. **recipes:** list_of_recipes
  10. **schedules:** list_of_schedules (ordered)
- **recipe:**
  1. **name:** string
  2. **status:** integer
  3. **comment:** string
  4. **optime:** time_duration
  5. **machine:** pointer_to_machine
  6. **control:** integer
  7. **operations:** list_of_operations
- schedule:
  1. **machine:** pointer_to_machine
  2. **scheduled_time:** time

3. **batch**: list_of_requests

- process:

1. **name**: string
2. **status**: integer
3. **comment**: string
4. **demand**: float
5. **priority**: integer
6. **operations**: list_of_operations (ordered)

- **operation:**

1. **name**: string
2. **process**: pointer_to_process
3. **recipe**: pointer_to_recipe
4. **max_rate**: float
5. **hedging**: float
6. **bufsize**: float
7. **production_rate**: float
8. **surplus**: float
9. **time_of_last_change**: time
10. **rework_percentage**: float
11. **user_inque**: integer
12. **waf_inque**: integer
13. **requests**: list_of_requests

- **request:**

1. **user_name**: string
2. **operation**: pointer_to_operation
3. **lot_number**: string

4. **status**: integer
5. **comment**: string
6. **priority**: integer
7. **number_of_waffers**: integer
8. **arrive_time**: time