Production Planning with Due-Date Constraints

by

Suguna Pappu

B.S., A.B., Mathematics and Statistics, 1985
Miami University
S. M., Operations Research, 1989
MIT

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

June 1992

Signature of Author________________________________
Sloan School of Management
May 15, 1992

Certified by________________________________
Stephen C. Graves
Professor of Management Science
Thesis Supervisor

Accepted by________________________________
Richard C. Larson
Co-Director, Operations Research Center

Contents

List of Figures

List of Tables

Abstract

We consider the problem of developing a production load plan in a dynamic environment for a make-to-order system. The demand process is assumed to be stochastic. Each unit of demand has associated with it a due-date: A due-date may be rigid and cannot be violated, or are non-rigid and can be violated at a cost. The objective of our problem is to choose a loading policy from the class of admissible policies to optimize a performance measure subject to the due-date constraints.

In particular, the performance measure we are interested in is a function of the smoothness of production and the mean inventory level. The production smoothness of an individual station is measured by the variance in the output process of the station. The smoothness of a multi-machine system is defined as a function of the variance in production level of the individual stations that comprise the system.

We develop production plans that minimize a function of the variance in production and inventory holding costs while meeting the due-dates of the jobs for a one machine system. Then, we relax the rigid due-date assumption and develop several models that incorporate tardiness costs which we use to identify the optimal loading policy. Finally, we generalize the one machine rigid due-date model to multi-machine systems, considering both a serial line and a job shop.

Advisor: Stephen C. Graves, Professor of Management Science

Acknowledgements

This work was supported jointly by the Leaders for Manufacturing program and by the National Science Foundation's Strategic Manufacturing Initiative Project " Decision Making in Manufacturing Systems" (NSF Grant DMM-8914181).

I begin by thanking two of my committee members, George Chryssolouris and Larry Wein for several helpful comments about my thesis. Next, I would like to thank Gabriel Bitran for taking an interest in my career and well-being these last two years, and Charlie Fine for being a ready ear to listen.

I am especially grateful to my advisor Steve Graves. I have completed both my master's and doctorate with Steve. He gave me the freedom to explore problems of my choice and was patient enough to wait until I found one that I liked. He is very analytical and insightful. I thank him for sharing some of that with me.

My friends at MIT have been a source of tremendous joy during my stay here. How could I possibly explain Vulcan goldfish, Joke # T2, mass murderers & shadows & EMs, epsilon-evil, kitchens & closets that aren't in anyone's house? Thanks for many good times:

Where I'm Goonie: Michele, Janet, Malini, Lalita, Pam
Where I'm Pippu: Ram, Joe, Sungsu, Eddie, a.k.a. the singing thugs
Where history goes back to the beginning of time : Peter, Jim
For a gentle nudge and a smile: Anna, Constance, Laura, Michelle, Paulette
When I needed a friend and mentor and aspired to be the same: Murali

And my family: Sridhar, for being as quirky and cool as I knew he was capable of being and still willing to acknowledge me as his sister. Ravi with whom I have already walked seven steps but now walk another step with each day.

Finally, I thank my parents for their unflagging faith:

My father for telling me again and again that a PhD is not an end but a beginning
My mother for believing that I could do anything that I put my mind to (plus a little bit more)

In memory of my uncle,
Venkat Rao

If you can keep your head while
all about you are losing theirs and blaming it on you
– Rudyard Kipling

Mama always told me not to look into the sights of the sun
Oh, but mama, thats where the fun is
– Bruce Springsteen

Chapter 1

Introduction

1.1 Problem Description and Motivation

1.1.1 The generic problem

A make-to-order production facility may find that some of its customers are very strict about enforcing the delivery date of orders, while others are willing to accept late jobs. With respect to the first set, the producer may feel compelled to satisfy the requested due-dates for strategic reasons. One incentive is that failure to do so will result in loss of the customer. If this customer represents a significant source of income, then the producer has little clout in the situation. We refer to these jobs as having *rigid* due dates. The other type of customer may be of low priority either due to its relative size or due to the coercive power of the producer. For this reason, it may not be compelling for the producer to satisfy the due-date requests of this class of jobs. We refer to this class of jobs as having *non-rigid* due-dates.

This scenario is typical for a producer who simultaneously operates in several different markets. The type of due-date requested and honored is indicative of the relative strength of the producer and consumer in each of the individual markets. In a highly competitive supply market one would expect to see rigid due-dates. In a more oligopolistic setting, in which the producer has a relatively captive market, we anticipate that there is some more interaction between the two agents. This type of producer is found in the middle of a supply chain. In other words, the customer to whom we will be referring is itself a producer for another good.

An important problem in this context is planning for production. In particular, the production manager must take into account operational costs and strategic considerations and develop a production strategy to accomodate the various demands. We first present an example of this problem and provide a brief overview of the specific issues at hand.

1.1.2 Example

We found a scenario similar to the one described above at the printing facility operated by the International Paper Corporation (IP) in Framingham, MA. In this operation, they print cartons for use in beverage containment, e.g., milk, orange juice. First, they receive rollstock paper from other IP locations. For processing, the paper is printed with the appropriate messages, cut, and scored, and then a vertical seam is placed on the carton. The customer will then fill the carton and perform any other necessary sealing operations.

To summarize the sequence of events: The customer places an order to the IP facility. This order consists of information on the physical specifications of the carton, an order quantity, and a due-date. IP's response to the order varies. For example, in the case that the mammoth Hood dairy is the customer, the order is accomodated in the schedule in such a way that the due-date can be satisfied. If, however, the customer is a smaller "mom-and-pop" operation, and the shopfloor is congested then IP may feel that it may be unable to meet the due-date request. In this case, they either attempt to do some negotiation to set back the requested due-date, or indicate that the work may be delayed and that the dairy will have to accomodate itself.

Although there is a well-established group of customers, the orders themselves arrive randomly. IP may have some idea of the requirements, but they are not informed about special promotions etc., ahead of time. This type of information is often kept secret for strategic reasons. Additionally, the products themselves are changing: the panels change frequently, e.g., new missing children are put on the cartons every two weeks.

The IP facility operates as a classic make-to-order system; the units ordered vary drastically from customer to customer, and from the same customer from period to period, and so only finished goods inventory for contracted orders can be held. In this type of environment there is a need for models for production planning and smoothing. These models should indicate in what manner the system will determine production levels to satisfy the demand requirements.

1.1.3 Elements of the model

We consider a dynamic, make-to-order production environment in which the demand arrivals are stochastic and each unit has a different due-date. We want to devise a production strategy to satisfy these due-date requirements according to different objectives. We discuss this further as follows.

Production Strategy

A production strategy is a plan indicating how and when work is to be processed. There are two perspectives of this: At a detailed level, we follow the progress of individual jobs. In this case, issues of sequencing, starting/completion times, and setups arise. In the example of carton printing, the order in which the jobs are processed is significant: for example, jobs which share colors require less time for setups since the rollers require less cleaning between jobs. From this perspective, the production strategy is referred to as *detailed scheduling* or *dispatching* .

This type of scheduling is done on the shopfloor, perhaps in conjunction with the actual machine operators. The time frame here is very tight. Precise information is kept about exact starting and completion times, specific processing requirements for the individual jobs.

At a more aggregate level of detail, the production plan is set via a *loading policy* . Loading is concerned with setting workload levels, not with the processing of individual jobs. As such, a total quantity of work is set for production in a particular period, with the decision of which among those awaiting processing to release determined by the particular dispatching rule being employed.

This perspective is typically held by the plant manager, who views the operation in a global way. The time scale is broader, with a period being anywhere from one day to a week or month.

In this thesis we are primarily concerned with developing a loading policy appropriate for our context. The more detailed type of analysis is too complicated in this dynamic-stochastic setting, given that most due-date problems of any complexity in a static environment are already NP-complete. More significantly, an aggregate perspective will be useful for the plant manager in terms of setting workforce levels, assessing input material requirements, and projecting overtime costs over a particular horizon. We return to these points in the discussion section.

Due-date Constraints

As we have indicated above there are two classes of jobs, each distinguished by whether its due-date is considered rigid or non-rigid. A constraint corresponding to a rigid due-date is easily interpretable– the production level of the system must be set in such a way to ensure that all orders in this class are delivered on time. A feasible production strategy satisfies all rigid due-date constraints.

A non-rigid due-date corresponds to the class of work where the due-date is *violable.* In particular, the customer may request a due-date, but is willing to accept late delivery of an order, albeit at some cost to the producer. This may take the form of price-cuts for the customer.

Objective Functions

The type of objective functions we propose are expressed in terms of costs, and for a given mathematical program representing the production strategy, we seek to minimize the cost function. Silver[67] identifies the primary costs of a production strategy in a stochastic environment:

1. Basic production costs
2. Cost of changes in production rate
3. Inventory and shortage costs

Shortage costs are not relevant in a make-to-order context. Instead we will propose that *waiting* costs are of interest, particularly in the case in which we have violable due-dates. The objective functions that we will propose are expressed as functions of these costs.

When the costs associated with changes in the production level are significant, one component of the objective is to minimize these variations; this is referred to as *smoothing.* It may be that the form of the cost function associated with the variability in the production level dominates. In this case the entire objective function is consistent with smoothing, and encompasses all of the relevant costs. We will often refer to planning and smoothing interchangeably, although they represent different concepts.

1.1.4 References

Planning

Previous research in production planning has concentrated exclusively on make-to-stock systems. For the sake of continuity, we do not describe this literature in too much detail here. Representatives of this work are given in an annotated bibliography.

In a make-to-stock system with stochastic demand requirements, one strategy for satisfying demand is to build up inventories, which then vary to absorb some of the randomness. Backordering is often permitted, i.e., inventory can go "negative". This is not possible in a make-to-order context. Cruickshanks, Drescher and Graves [84] were the first to look at the planning and smoothing problem in the context of a make-to-order system. They identify two strategies to accomodate demand fluctuations:

1. vary production to match the demand rate
2. maintain an approximately constant production rate while varying the delivery time of orders depending on the shop load

In their work they modify the problem and propose a third option. Suppose that the customer in addition to placing a particular order, also specifies a fixed delivery leadtime. The leadtime is the same for all customers. They define the planning window to be the difference between the delivery lead time and the production leadtime for a unit.

They identify variation in production level to be a significant cost. Thus, the objective is to achieve a smoothing benefit over this planning window. Suppose that the planning window is of length N and the production time is one period long. In their approach, they approximate the per-period production rates over this window by the strategy specified by option (2) above; the delivery time window quoted to the customer is of length N. They show via a simulation study that this can be a good approximation.

Due-dates

In the context of due-dates, there are three types of interactions of due-dates with the scheduling constraints.

1. Due-date is exogenous, i.e., quoted by the customer and it is rigid
2. Due-date is exogenous and is flexible
3. Due-date is endogenous, i.e., quoted by the producer and is either rigid or flexible

All of the research addressing scheduling with due-dates in a dynamic environment addresses the problem of setting an appropriate endogenous due-date. The work done in the first two areas is expressed in a static environment with capacity restrictions. The typical objective seeks either a feasible schedule (1) or minimization of a function of the tardiness of the given jobs (2).

Motivation for joint consideration

The context of planning together with exogenous due-dates restrictions in a dynamic environment has received no attention in the literature. Although there is an abundance of work related to due-date issues none of it is applicable to the scenario we have described.

There is tremendous scope for research in this area becuase of its relevance in today's increasingly competitive production environment. Stalk[91] identifies *time* as the key variable of competitive advantage in today's marketplace. The emergence of JIT technologies is an example of this. This heightens the need for planning schemes which explicitly consider due-dates.

The primary difference in the motivation of our problem compared to the bulk of the literature is that we want to plan production to accomodate externally generated stresses on the system.

1.2 Outline

In this thesis we will first present a model for a one-machine, single-product system, when all jobs have rigid due-dates. The structure we devise for this problem lends itself to a relaxation in which we can analyze the problem in the context of non-rigid due-dates. The analysis and techniques for the one-machine problem will then generalize to larger systems, e.g., serial lines and job-shops.

We begin by describing the problem in more detail and developing a basic model. We first propose a finite horizon dynamic programming formulation for the problem in which all due-dates are rigid. The solution of this formulation we refer to as a *linear loading* policy.

We then use this policy in the context of an infinite horizon problem. The policy is workload-dependent, requiring that the production level is set as a function of the workload queue each period. We determine the optimal policy as a function of the parameters. Then, we show that this policy is equivalent to an assignment policy in which the current queue length is ignored. Next, we examine the inventory implications of implementing such a policy.

We then describe two models in which all due-dates are non-rigid. First we present a model in which all of the customer requests are for the jobs to be done "as soon as possible." In the second model, we have the customers entering with a specific due-date request, but with the option that they are willing to accept work even after its due-date. The cost to the producer of a late delivery is included in the optimization program in addition to the original terms.

Then, in a multi-machine context we consider the problem of setting workload levels for an entire system of machines. To do this we first extend the concepts used in the 1-machine case to be appropriate in the larger system. An important consideration here is that our smoothing objective for a one-machine system is well-defined– measured by the variance in the output process of the system. In a multi-machine system, it is not clear how to evaluate the smoothing objective. We consider several objective functions that are functions of the variance in output processes of individual machines and propose heuristics. Additionally, now we must also account for the presence of work-in-process inventory, which is not relevant in a one-machine system.

Finally, we discuss the appropriateness of our models. We conclude with directions for future research.

Chapter 2

The One Machine System: Rigid Due-Dates

2.1 Outline

In this chapter we consider the problem of production planning for a one-machine, make-to-order system which must accomodate stochastic demand arrivals, each with a rigid due-date. We begin with a discussion of the costs and constraints of interest. Then we formulate a finite horizon dynamic program to describe the discrete-time, dynamic problem. However, the solution to this is too difficult to characterize in closed form. When we relax the constraints on the production level, we find that the optimal production level is linear in the quantity of unfinished work.

A well-known problem with finite horizon analyses is the resolution of end-term effects. We would like to evaluate the performance of a policy over an infinite horizon. In particular, we consider the linear policy suggested by the solution of the unconstrained finite horizon problem in the infinite horizon. The measure of performance is a function of the variation in the production level over time, and the long-run inventory holding costs.

2.2 A Model for Rigid Due-dates

We describe a discrete-time model of a one machine single-product system. We discuss our assumptions, notation, relevant costs, and constraints. We present the problem in a very general form, and then consider several, more tractable special cases.

2.2.1 Assumptions

- The system is uncapacitated. This is clearly a very strong assumption, but need not be taken literally. There are several perspectives on this. It is possible in some scenarios that the acquisition of additional capacity is easy. Consider a facility which has the potential to operate on three shifts, but normally operates on only one or two. Here, additional capacity has a cost in terms of overtime payments, but it is a feasible option to add shifts. Another useful interpretation of this is that we can assume that the system is costlessly uncapacitated, and see whether the production policies we propose are restricted by capacity limitations. If capacity does pose a constraint, then it might indicate that the purchase of additional capacity is warranted or that the production policy is not applicable and needs to be modified. Finally, suppose that in the objective function there is a term representing variance in production level. Minimizing this would serve as a kind of surrogate for a capacity constraint.

- Jobs arrive at the beginning of each period t, each with an integer leadtime j, indicating that the unit is due at the beginning of period $t+j$. We assume that this due-date cannot be violated, although we may process jobs early without penalty in order to satisfy the due-date. This work processed ahead of its due-date is held as finished goods inventory.

- Each job requires exactly one unit of processing time, and this is independent of the total amount of work processed concurrently. The idea here is that each unit of demand has the same quantity of processing requirements. This is not to say that the actual processing is the same for all jobs, but that the amount of time required is the same. In our carton printing example, the jobs differ in terms of the colors and print plates required; however, these distinctions are addressed in the setup stage, and once the press is running, the processing time requirements are essentially the same. We can then normalize the processing requirements to one unit.

  In the case that a job requires $k > 1$ units of processing, we can split this into k jobs of unit length, each with the same due-date.

- All work scheduled for production in period t will be completed by the beginning of period $t + 1$ (or at the end of period t). This assumption is not very strong when the processing time is significantly less than the length of a period.

- The revenue generated by each unit of work is identical, and so we consider only the cost minimization aspect of the problem.

- All work enters with a due-date no longer than N periods from the current time period, where N is the length of the time horizon considered by the planner.

2.2.2 Notation

Notationally, we need to account for two types of information. First, each period t we are interested in the realization of the stochastic demand quantities, and in the value chosen for the decision variable, the production level in period t. Second, we want to maintain information about the time remaining in the due-date window for each job. We will use a superscript and a subscript for this accounting, as described below.

Additionally, the notation we choose is indicative of the type of variable involved. In particular, a lower-case variable represents a scalar quantity, while an upper-case letter denotes a vector. The elements of a vector however, are given in upper-case form with the appropriate sub- and super-scripts. The exception to this is the use of greek letters, which also designate vector quantitites. Matrices are given by italicized upper-case letters; the elements of the matrix are given by lowercase letters.

- $D_t^i \; i = 1, \ldots, N$ is a random variable, denoting the demand in units of work that arrives *at the beginning of* period t and is due *at the beginning of* period $t + i$. Here, the subscript indicates the time of observation, and the superscript the length of the due-date window. D_t^i is independent of $D_t^j \; i \neq j$, and the mean and variance of D_t^i are $\overline{D^i}$, and σ_i^2, respectively.

  In vector notation we have D_t with elements D_t^i, $i = 1, \ldots, N$. We assume $D_t, D_{t+1}, \ldots$ are independent, identically distributed random vectors, with mean $\overline{D} = \{\overline{D^i}\}$ and covariance matrix, $Cov(D) = diag\{\sigma_i^2\}$.

- $\hat{D}_{t+1}$ is the sum of all of the demands due at the beginning of period $t + 1$, i.e., $\hat{D}_{t+1} = \sum_{i=1}^{N} D_{t-i+1}^i$. We use this notation to indicate the delivery quantity that has to be satsified during period t.

- $W_t^i \; i = 1, \ldots, N$ denotes the total quantity of work known to be due at the beginning of period $t + i$ that has not been processed at the beginning of period t. This observation is made after the arrival of D_t. W_t is the vector of unprocessed or remaining demand, ordered by due date at the beginning of the $t-$th period with elements W_t^i. Again, we have the subscript t denoting the time of observation, while the superscript i accounts for the time remaining in the due-date window. We depict this in Figure (2.2.2).

- P_t^i $i = 1, \ldots, N$ is the control variable set at the beginning of period t (but after observing D_t) indicating the quantity of W_t^i that contributes to the production quantity during period t. This is conceptually a little tricky. Here we want to reiterate the distinction between a loading policy and a dispatch protocol. The loading policy indicates how much work is to be processed in a particular period without regard for the identities of individual jobs. The dispatch protocol distinguishes the jobs and then indicates the actual order in which they are processed.

  In setting a load quantity, we want to consider all of the unfinished work remaining in the system, in this case, ordered by the due-date, and given by W_t. Looking over the length of the planning horizon, we think of "drawing" work from periods further out in the horizon to contribute to the production level set for this period. In other words, we explicitly consider how much work we anticipate for future periods in setting this period's production. Depending on our objective function we may choose to process some of the work early as a hedge against future uncertainty. In Figure (2.2.2) we show how this may be done.

  We will not address the form of the dispatch protocol in this section, but defer the discussion until the context of non-rigid due-dates.

  P_t is the production vector in period t with elements P_t^i. The actual quantity processed in period t is given by $\sum_{i=1}^{N} P_t^i$. In shorthand, this is eP_t , where e denotes the N-vector of 1's. For ease of notation, $p_t = eP_t$, which is the total production during period t.

- z_t is a scalar quantity denoting the completed work due at the beginning of period $t+1$ or later, i.e., finished goods inventory at the beginning of period t.

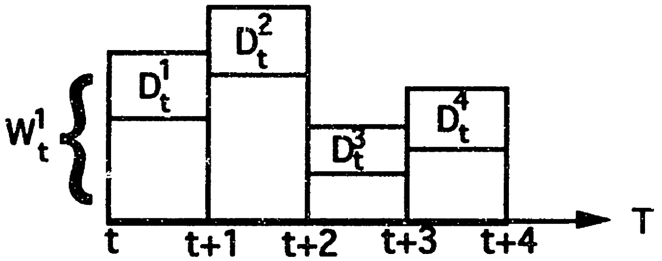

Figure 2-1: Height of stack k is W_t^k

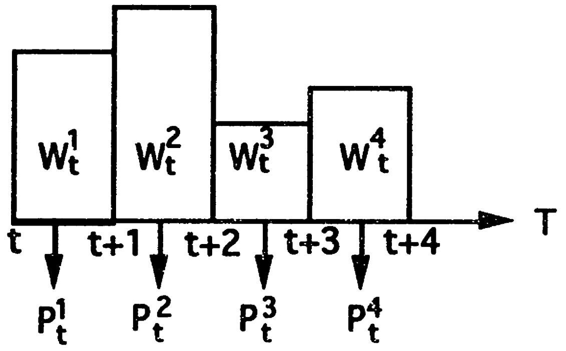

Figure 2-2: Production contribution from each W_t^k

The description of the model is given via the following:

Stochastic characterization of D_t

Dynamics of the state variables: $W_t,\ z_t$

Constraints on P_t

Costs

Objective Function

2.2.3 Dynamics of the state variables: $W_t,\ z_t$

The state of the system at the beginning of period t is given by the quantity $(W_t,\ z_t)$. It evolves according to:

$$W_{t+1}^{i} = W_t^{i+1} - P_t^{i+1} + D_{t+1}^{i} \tag{2.1}$$

In other words, in period t, a certain portion of W_t^{i+1} is set for production, given by P_t^{i+1}. The remainder becomes a part of W_{t+1}^{i} in period $(t+1)$ along with the new arrivals in period $(t+1)$ that are due at the beginning of period $t+i+1$.

In vector notation, this is given by:

$$W_{t+1} = C(W_t - P_t) + D_{t+1} \tag{2.2}$$

where C is an $N \times N$ $0-1$ matrix, with $1's$ on the upper diagonal and $0's$ elsewhere. Premultiplication by C advances the workloads by one period.

The flow balance for z_t is given by:

$$z_{t+1} = z_t + p_t - \hat{D}_{t+1} \tag{2.3}$$

This indicates that the finished goods inventory in period $t+1$ is equal to the finished goods inventory level of period t, adjusted by the net value of an addition of the production in period t and subtraction of demand due at the beginning of period $t+1$.

2.2.4 Constraints

There are two types of constraints associated with our planning problem: One addresses due-date satisfaction and the second requires production to be non-negative and bounded by the available work.

The requirement to meet due dates is expressed by:

$$z_t \geq \hat{D}_t \ \ \forall\, t \tag{2.4}$$

In other words, the finished goods inventory at the beginning of the period t must be sufficient to cover the demands due at the beginning of the period.

There is no backlogging in this model, and so, the production level must always be non-negative:

$$P_t^i \geq 0 \ \ \forall i,\ t \tag{2.5}$$

Additionally, we have a make-to-order system, so that we do not produce more than has been ordered:

$$P_t^i \leq W_t^i \ \ \forall i,\ t \tag{2.6}$$

In fact, we can make these constraints a little more concise. Note that P_t^1 is not really a decision variable at all, but in any feasible solution must be equal to the level of unfinished work due in period $t+1$, W_t^1. Thus, constraints (2.4) - (2.6) are equivalently given by

$$0 \leq P_t^i \leq W_t^i \ \ i = 2, \ldots, N \, \forall t \tag{2.7}$$

and

$$P_t^1 = W_t^1 \, \forall t \tag{2.8}$$

From now, we only refer to the constraints (2.7) and (2.8).

2.2.5 Costs and Objective Function

As we have noted earlier, there are three primary costs of interest in setting a production policy:

1. "Regular" production costs.
2. Cost of changes in production rate, or "smoothing" costs.
3. Inventory costs.

We assume production costs are given by a linear function, increasing in the output level. However, these costs are independent of the production policy and can thus be ignored.

We assume the cost of changes in the production rate can be represented as a convex function. Suppose we know the long run, optimal level of production, p^*, based on historical data, stationarity of the demand processes, etc. The system is set up to accomodate this; for example, this accounts for the labor force level. A production level higher than this requires the use of overtime, which grows at least linearly with the excess over p^*. Work released at a lower level indicates that the system is being under-utilized. Again, given as a cost we can envison linear costs, increasing in either direction away from p^*. Taken together, the two costs can be given by either a quadratic, or more generally, a convex function, minimized at the *apriori* designated p^*, where $p^* = \sum \overline{D}^j$

The minimization of these particular costs is often called a smoothing objective, in refcrence to the attempt to "smooth" out the variations in the production level. More generally, the planning problem itself is, in somewhat of a misnomer, called the *smoothing* problem; this is valid when the variation in production level is identified as the primary cost of interest.

There are two types of inventory costs, corresponding to holding finished goods inventory and to holding unfinished work. In the make-to-order environment of our model, and with the assumption that all work requires exactly one unit of processing time, the traditional issues associated with WIP inventory do not arise in the one machine scenario. We return to this in the multi-machine model. However, if we assume that a customer does not take receipt of work until its specified due date, then it is of interest to see how much finished goods inventory is held. In particular, there is a well-known tradeoff between smoothing the output process and holding inventory. In this make-to-order system some of the work is done ahead of time to achieve the desired smoothing benefits. In doing so, however, we must bear the associated storage costs. We assume that the inventory costs are incurred according to the level of goods being stored and the length of time for which they are held. This we represent by a linear function.

Before we quantify these costs, we want to identify a third type of cost, a waiting cost. This is the cost representing the *delay* in processing work after its arrival. This concept does not have tremendous applicability in the context of rigid due-dates, given that all work generates the same revenue, as long as the due-dates are satisfied. However, this cost becomes important when we introduce flexible due-dates. We mention it here for the sake of completeness, but discuss the details later.

Combining all of these costs we propose that a cost function convex in the state variables (W_t, z_t) and the production level P_t is appropriate. This form of function has been suggested by many researchers in the study of production planning for make-to-stock systems. Some have chosen a specific quadratic function, while several others have considered a general convex form. Representatives of these models are given in the bibliography.

To be more specific, the per period cost c_t is given by $c_t(W_t, z_t, P_t)$. We suggest that $c_t(W_t, z_t, P_t)$ have the following form:

$$c_t(W_t, z_t, P_t) = f_t(p_t - p^*)^2 + g_t z_t \quad \forall t \tag{2.9}$$

for positive constants f_t, g_t. Th? values may be generated empirically or may otherwise be given to reflect the relative importance of the different components. The first term captures the smoothing cost, assigning a penalty for production in either direction away from the optimum. The second term represents finished goods inventory costs.

There are several types of objective functions that may be appropriate, dependent upon whether the time horizon of the problem is finite or infinite. For a finite horizon of length T we consider

$$\min_{P_1, P_2, \ldots, P_{T-1}} \sum_{t=1}^{T} c_t(W_t, z_t, P_t) \tag{2.10}$$

For a problem with an infinite horizon a reasonable objective function is

$$\min_{P_t} E_{D_t}[c_t(W_t, z_t, P_t)] \tag{2.11}$$

From this, we now proceed by considering in turn a finite horizon and then an infinite horizon problem. In particular, we analyze the finite horizon problem with the idea that it offers insight into the analysis of the infinite horizon problem.

2.2.6 A Finite Horizon Dynamic Programming Formulation

Consider a finite horizon version of the problem over a planning horizon of length T. Note the distinction between the length of the planning horizon and the length of the due-date window, which has an upper bound of N. Without loss of generality, we only consider scenarios for which $N << T$.

In addition to the per period cost described above, we need to assume that there is a terminal cost incurred in period T. We assume that this cost has a similar form, given by:

$$c_T(W_T, z_T) = W_T^T \mathcal{F}_T W_T + g_T z_T \tag{2.12}$$

with known constant scalar g_T and a positive semidefinite matrix of constants $\mathcal{F}_T$. This latter quantity is designed to capture a flavor of the production costs of the remaining unfinished work. Note that there is no production decision in the final period, and thus we do not include it as an argument in the terminal cost term.

The control P_t is constrained to take values from the set of feasible control policies $\mathcal{P}_t(W_t, z_t)$, where

$$\mathcal{P}_t(W_t, z_t) = \{P_t^i(W_t, z_t) | 0 \leq P_t^i \leq W_t^i \;\; i = 2, \ldots, N \; \forall t, P_t^1 = W_t^1 \; \forall t\} \tag{2.13}$$

Note that the set of feasible controls is independent of z_t. Let μ be a mapping of the state (W_t, z_t) into the control P_t. A control policy is a sequence of such functions $\pi = \{\mu_0, \mu_1, \ldots, \mu_{T-1}\}$.

Suppose we have a sequence of functions $\pi = \{\mu_0, \mu_1, \ldots, \mu_{T-1}\}$ where μ_t maps (W_t, z_t) into $P_t = \mu_t(W_t, z_t)$, and is such that $\mu_t(W_t, z_t) \in \mathcal{P}_t(W_t, z_t)$. All such policies π are called admissible. The class of all admissible policies we denote Π.

For an admissible policy $\pi = \{P_0, \ldots P_{T-1}\}$ and given an initial (W_0, z_o), the total expected cost over the planning horizon is

$$J_\pi(W_0, z_0) = E_{D_t, t=0,\ldots,T-1}\{\sum_{t=0}^{T-1} c_t(W_t, z_t, P_t) + c_T(W_T, z_T)\} \tag{2.14}$$

where c_t are given, for $t = 1, \ldots, T$.

The optimal control policy π^* minimizes the cost where

$$J_{\pi^*}(W_0, z_0) = \min_{\pi \in \Pi} J_\pi(W_0, z_0) \tag{2.15}$$

In the standard DP methodology, the approach to solve the finite horizon program uses Bellman's ***principle of optimality.*** For more details on the methodology, see Bertsekas[87].

In our context, we can state the principle of optimality as follows: Let

$$\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{T-1}^*\} \tag{2.16}$$

be an optimal policy.

Consider the subproblem at time t with state (W_t, z_t). We want to minimize the "cost-to-go" from time t to time T, denoted $J_t(W_t, z_t)$ and given as :

$$J_t(W_t, z_t) = \min_{P_t, P_{t+1}\ldots} E\{c_T(W_T, z_T) + \sum_{k=t}^{T-1} c_k[W_k,\ z_k,\ \mu(W_k, z_k)]\} \tag{2.17}$$

and assume that when using π^*, the state (W_t, z_t) occurs with positive probability. Then the truncated policy $\{\mu_t^*, \mu_{t+1}^*, \ldots, \mu_{T-1}^*\}$ is optimal for the subproblem. Continuing for all t, and using this in the dynamic programming algorithm, we have that

$$J_{\pi^*}(W_0, z_0) = J_0(W_0, z_0) \tag{2.18}$$

where the function $J_0(W_0, z_0)$ is given by the recursion

$$\begin{aligned} J_T(W_T, z_t) &= c_T(W_T, z_T) \\ J_t(W_t, z_t) &= \min_{\pi\in\Pi} E_{D_{t+1}}\{c_t(W_t, z_t, P_t) + J_{t+1}(W_{t+1}, z_{t+1})\} \end{aligned} \tag{2.19}$$

If $\pi_t^* = \mu_t^*(W_t, z_t)$ minimizes the right-hand-side of (2.19) for each (W_t, z_t) and t, then $\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{T-1}^*\}$ is optimal.

The solution proceeds via backwards recursion beginning with period $(T-1)$. At each stage, the optimization problem we need to solve is:

$$J_t(W_t, z_t) = \min_{P_t} f_t(p_t - p^*)^2 + g_t z_t + E_{D_{t+1}}\{J_{t+1}(W_{t+1}, z_{t+1})\} \tag{2.20}$$

Note that there are only $(N-1)$ decision variables, P_t^i $i = 2, \ldots, N$ although we will refer to the entire vector P_t as a decision quantity.

Period $(T-1)$ We want to solve:

$$J_{T-1}(W_{T-1}, z_{T-1})$$

$$\begin{aligned} &= \min_{P_{T-1}} f_{T-1}(eP_{T-1} - p^*)^2 + g_{T-1}z_{T-1} \\ &\quad +E_{D_T}\{J_T(W_T, z_T)\} \\ &= \min_{P_{T-1}} f_{T-1}(eP_{T-1} - p^*)^2 + g_{T-1}z_{T-1} \\ &\quad +E_{D_T}\{W_T^T \mathcal{F}_T W_T + g_T z_T\} \\ &= \min_{P_{T-1}} f_{T-1}(eP_{T-1} - \sum_{j=1}^{N} \overline{D}^j)^2 + g_{T-1}z_{T-1} \\ &\quad +E_{D_T}\{[C(W_{T-1} - P_{T-1}) + D_T]^T \mathcal{F}_T [C(W_{T-1} - P_{T-1}) + D_T] \\ &\quad +g_T(z_{T-1} + P_{T-1} - \hat{D}_T)\} \end{aligned} \tag{2.21}$$

where the second expression follows by substituting in (2.12). Then, using (2.2) and (2.3) to substitute for W_T and z_T the third expression follows.

This is difficult to solve in a closed form expression for arbitary values of $\mathcal{F}_T, g_T$ since we cannot guarantee satisfaction of $0 \leq P_{T-1}^i \leq W_{T-1}^i, i = 2, \ldots, N$. Suppose, instead that we solve the unconstrained version of the problem. The solution to this problem is a lower bound to the constrained version. In the case that the solution is feasible then it is optimal. Is there any insight that we can gain by considering this less restrictive version? In particular, recall that the primary motivation of this analysis is not to characterize the solution to the finite horizon problem exactly, but to suggest a type of policy that is reasonable for the infinite horizon problem. Our approach will be to see what the form of the solution to the unconstrained problem is for the finite horizon case. We then add conditions to impose feasibility and evaluate the policy's behavior in the infinite horizon.

Differentiating (2.21) with respect to P_{T-1}, and solving, we find that the Kuhn-Tucker conditions apply and that we get

$$P_{T-1}^* = \mathcal{A}_{T-1} W_{T-1} + B_{T-1} \tag{2.22}$$

where $\mathcal{A}_{T-1} = \{a_{ij}\}$ is an $N \times N$ matrix of constants, and B_{T-1} is a vector of constants.

In particular these constants arise in solving for $N \times N$ system of equations that we get by setting the gradient of (2.21) to 0. Since (2.21) already has an assumed quadratic form, this follows.

Substituting P^*_{T-1} into (2.21) we find that the form of $J_{T-1}(W_{T-1}, z_{T-1})$ is given by

$$J_{T-1}(W_{T-1}, z_{T-1}) = W^T_{T-1}\tilde{\mathcal{F}}_{T-1}W_{T-1} + \tilde{g}_{T-1}z_{T-1} + \tilde{h}_{T-1}W_{T-1} \tag{2.23}$$

for some constants $\tilde{\mathcal{F}}_{T-1}, \tilde{g}_{T-1}, \tilde{h}_{T-1}$. These constants are generated when the linear form of P^*_{T-1} is substituted into (2.21).

Continuing the recursion in this manner, we find that the optimization problem at each stage has a solution of the form

$$J_t(W_t, z_t) = W^T_t\tilde{\mathcal{F}}_tW_t + \tilde{g}_tz_t + \tilde{h}_tW_t \tag{2.24}$$

following from a solution

$$P^*_t = \mathcal{A}_tW_t + B_t \tag{2.25}$$

where $\tilde{\mathcal{F}}_t, \tilde{g}_t, \tilde{h}_t$ and $\mathcal{A}_t, B_t$ are calculated constants. Given terminal costs for $\mathcal{F}_T, g_T$, the constants $\tilde{\mathcal{F}}_t, \tilde{g}_t, \tilde{h}_t$ and $\mathcal{A}_t, B_t$ are calculable for all t.

Presumably, for a given set of constants, we can solve the constrained finite horizon problem through some sort of enumerative technique. We have examined the unconstrained version, however, because we are more interested in the form of the solution than in the exact values. Thus, we have shown that the optimal solution is linear in the unfinished workload vector. In the next section, we offer sufficient conditions for feasibility and consider only those linear policies within that set.

2.2.7 An infinite horizon problem

In the context of an infinite horizon problem, the objective function that we will consider is given by:

$$\min_{P_t} E_{D_t}[c_t(W_t, z_t, P_t)] \tag{2.26}$$

Because we are considering steady-state behavior of the system and of particular production policies, we drop the time dependence on the per period cost, i.e., $f_t = f, g_t = g \; \forall t$ for a given f and g. Equation (2.26) becomes:

$$\min_{P_t} E_{D_t}[f(p_t - p^*)^2 + gz_t] \tag{2.27}$$

and applying the expectation through:

$$\min_{P_t} fVar[p_t] + gE[z_t] \tag{2.28}$$

Motivated by the form of the policy generated by the finite horizon problem, we propose analyzing the policy linear in W_t for the infinite horizon problem. Several modifications need to be made to accomodate such a policy for the infinte horizon version of the problem. First, the parameters should be independent of t, with a form:

$$P_t = \mathcal{A}W_t + B \tag{2.29}$$

where $\mathcal{A}$ is an square matrix of dimension N and B is an N vector. Given such a policy, we assume that a constant amount of production is guaranteed each period as determined by the elements of B (or, if any elements of B are negative, then a constant amount is being withdrawn).

We contend that this is a reasonable class of policy because of our analysis of the finite horizon model. The time-index is dropped because we are assuming an infinite-history for the problem. We have previously defined that feasibility corresponds to

$$0 \leq P_t^i \leq W_t^i \;\; i = 2, \ldots, N \; \forall t$$

and

$$P_t^1 = W_t^1 \; \forall t$$

The approach here will be to choose the optimal policy from the class of policies that satisfy the following conditions for feasibility

Property 1 — *Sufficient conditions for feasibility are:*

1. $B = 0$ and

2. $\mathcal{A} = \{a_{ij}\}$ is diagonal with $a_{11} = 1, \; 0 \leq a_{ii} \leq 1, \; i = 2, \ldots, N$

The first condition is not only sufficient, but it is necessary for feasibility. For example, when $W_t = 0$, but $B \neq 0$, $P_t = B$ violates the feasibility requirement. Together with the first condition, then, we argue that the characterization of $\mathcal{A}$ is sufficient for feasibility. For one, $a_{11} = 1$ corresponds to $P_t^1 = W_t^1$, and thus all work that remains in period t with a due-date of $t+1$ is produced in the current period. Additionally, by construction the constraints $0 \leq P_t^i \leq W_t^i \; i = 2, \ldots N$ hold. The $0 \leq P_t^i$ portion indicates that the production level is always non-negative, i.e., no backlogging of demand. $P_t^i \leq W_t^i$ follows since we have a make-to-order system and only process orders that have actually been placed.

This production release policy can be more compactly described as

$$p_t = \alpha^T W_t \tag{2.30}$$

where α is an $N \times 1$ vector with elements $\alpha_i = a_{ii}$. α_i represents the fraction of W_t^i that will be produced in period t. We have $\alpha_1 = 1$. Equation (2.30) follows because

$$\begin{aligned} p_t &= e^T P_t \\ &= e^T A W_t \\ &= \alpha^T W_t \end{aligned}$$

The dynamics of W_t become:

$$\begin{aligned} W_{t+1}^{i-1} &= W_t^i - P_t^i + D_{t+1}^{i-1} \\ &= W_t^i - \alpha_i W_t^i + D_{t+1}^{i-1} \\ &= (1-\alpha_i) W_t^i + D_{t+1}^{i-1} \end{aligned}$$

Or, more concisely, in vector form

$$W_{t+1} = C\hat{A}W_t + D_{t+1} \tag{2.31}$$

where C is as described above. $\hat{A}$ is an $N \times N$ diagonal matrix with elements $\hat{a}_{ii} = 1 - \alpha_i$, i.e., $\hat{A} = I - A$, where I is the identity matrix.

$$\begin{aligned} z_{t+1} &= z_t + p_t - \hat{D}_{t+1} \\ &= z_t + \alpha^T W_t - \hat{D}_{t+1} \end{aligned} \tag{2.32}$$

where $\hat{D}_{t+1}$ is the accumulation of all of the demands due at the beginning of period $t+1$, i.e., $\hat{D}_{t+1} = \sum_{i=1}^{N} D_{t-i+1}^i$.

Now, to choose the best linear policy from the class given by (2.29), we solve the following program, denoted $\Pi(Var[p_t], E[z_t])$

$$\min_{P_t} fVar[p_t] + gE[z_t] \qquad\qquad \Pi(Var[p_t], E[z_t])$$

s.t.

$$0 \le \alpha_i \le 1 \quad i = 1, \ldots, N$$
$$\alpha_1 = 1$$

The assumption of the linear production policy enables us to evaluate the quantities $Var[p_t]$, $E[z_t]$ as a function of α and the demand realizations D_t.

First, to compute the unfinished workload level of period $t+1$, we assume an infinite history and then recursively substitute (2.31) into itself :

$$\begin{aligned} W_{t+1} &= C\hat{A}W_t + D_{t+1} \\ &= \vdots \\ &= \sum_{k=0}^{\infty}[C\hat{A}]^k D_{t-k+1} \end{aligned} \tag{2.33}$$

One can show that $[C\hat{A}]^k = [0], k \ge N$, and so, the summation in (2.33) will converge. Evaluating the mean, we have the vector

$$\begin{aligned} E[W_t] &= E[\sum_{k=0}^{\infty}[C\hat{A}]^k\overline{D}] \\ &= E[\sum_{k=0}^{N-1}[C\hat{A}]^k\overline{D}] \\ &= [I - C\hat{A}]^{-1}\overline{D} \end{aligned} \tag{2.34}$$

where

$$[I - C\hat{A}] = \begin{bmatrix} 1 & -(1-\alpha_2) & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & -(1-\alpha_3) & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & -(1-\alpha_4) & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \dots & 1 & -(1-\alpha_N) \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

and inverting this quantity,

$$[I - C\hat{A}]^{-1} = \begin{bmatrix} 1 & (1-\alpha_2) & (1-\alpha_2)(1-\alpha_3) & (1-\alpha_2)(1-\alpha_3)(1-\alpha_4) & \dots & 0 & 0 \\ 0 & 1 & (1-\alpha_3) & (1-\alpha_3)(1-\alpha_4) & \dots & 0 & 0 \\ 0 & 0 & 1 & (1-\alpha_4) & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \dots & 1 & (1-\alpha_N) \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

More concisely,

$$[I - C\hat{A}]^{-1} = \{b_{ij}\}$$

where

$$b_{ii} = 1$$

$$b_{ij} = 0 \text{ for } i > j$$

$$b_{ij} = (1-\alpha_{i+1})(1-\alpha_{i+2})\dots(1-\alpha_j)\text{for } i < j$$

From this we then have that

$$(1,\ \alpha_2,\ \alpha_3,\ \ldots)[I - C\hat{A}]^{-1} = (1,1,1,\ldots)$$

By (2.33) the production level is given by:

$$\begin{aligned} p_t &= \alpha^T W_t \\ &= \alpha^T \sum_{k=0}^{\infty} [C\hat{A}]^k D_{t-k} \end{aligned} \tag{2.35}$$

Applying the expectation operator to (2.35) and given the stationarity of the demand process,

$$\begin{aligned} E[p_t] &= \alpha^T [I - C\hat{A}]^{-1} \overline{D} \\ &= \sum_{j=1}^{N} \overline{D}^j \\ &= \sum_{j=1}^{N} \overline{D}^j \end{aligned} \tag{2.36}$$

which follows from $E[W_t]$ and evaluation of $[I - C\hat{A}]^{-1}$. as anticipated. Similarly, from (2.35), we find that

$$Var[p_t] = \alpha^T \sum_{k=0}^{\infty} [C\hat{A}]^k Cov(D) [\hat{A}^T C^T]^k \alpha \tag{2.37}$$

We show how this simplifies to a more intuitive representation and is easier computationally.

Property 2

$$Var[p_t] = \sum_{i=1}^{N} a_i \sigma_i^2 \tag{2.38}$$

$$where \quad a_i \quad = (1 - \alpha_i)^2 a_{i-1} + \alpha_i^2$$

$$a_1 \quad = 1$$

There are two ways in which to evaluate the expression for the variance in equation (2.37). The first way is to examine each of the terms in the expression explicitly and then to manipulate the matrices directly. The second way to evaluate this quantity is more intuitive, and we make an argument for this by using a specific example to generate the terms.

Given a matrix A, let the i-jth element be denoted A_i^j. Suppose that we have a diagonal N-matrix $\mathcal{M} = diag\{m_{ii}\}$. Applying the update operator C to this matrix, we find that the resulting matrix $C\mathcal{M}$ moves the elements up onto successively higher diagonals. In the matrix representation, this is given by

$$(C\mathcal{M})_i^j = \begin{cases} m_{i+1,i+1} & j = i+1 \\ 0 & j \neq i+1 \end{cases}$$

In general, the i-jth element of the quantity $[C\mathcal{M}]^k$

$$([C\mathcal{M}]^k)_i^j = \begin{cases} m_{i+k,i+k} & j = i+k \\ 0 & else \end{cases}$$

We refer to the upper diagonal on which there are non-zero terms as the kth diagonal. In what follows we present two ways in which to evaluate (2.37).

Approach 1: Matrix Manipulation

Consider the kth term of the expression, which we denote by S_k

$$S_k = [C\hat{A}]^k Cov(D)[\hat{A}^T C^T]^k$$

From above, we know that

$$([C\hat{A}]^k)_i^j = \begin{cases} (1-\alpha_{i+k}) & j = i+k \\ 0 & else \end{cases}$$

Recall that $Cov(D)$ is a diagonal matrix. Applying the pre- and post-multiplication terms, we find that S_k is also a diagonal matrix.

$$(S_k)_i^j = \begin{cases} (1-\alpha_{i+k})^2\sigma_{i+k}^2 & i = 1, 2, \ldots, N-k \\ 0 & i = N-k+1, N-k+2, \ldots, N \end{cases}$$

When we add up these terms for all of the matrices in (2.37), and then apply a pre- and post-muliplication of the vector α we get the desired result.

Approach 2: Enumeration Argument

Alternatively, we argue that (2.37) is simply an enumeration of all of the paths that work can take and the corresponding impact on the variance in the production level of the station. Here a "path" refers to the cumulative information indicating how many periods the job sat in the queue, given its due-date slack at the time, and the period in which it was finally produced.

We make this argument via a specific example. Consider the term a_3 which we argue accounts for the variation in production of D_i^3. Writing out the terms,

$$\begin{aligned} a_3 &= (1-\alpha_3)^2 a_2 + \alpha_3^2 \\ &= (1-\alpha_3)^2[(1-\alpha_2)^2 a_1 + \alpha_2^2] + \alpha_3^2 \\ &= (1-\alpha_3)^2(1-\alpha_2)^2 + (1-\alpha_3)^2\alpha_2^2 + \alpha_3^2 \end{aligned}$$

The third term corresponds to the contribution by the $\alpha_3 D_t^3$ quantity processed immediately upon its entry to the system. $(1-\alpha_3)\alpha_2 D_t^3$ is delayed one period and then processed, accounting for the second term. Finally, a quantity equal to $(1-\alpha_3)(1-\alpha_2)D_t^3$ is delayed a full two periods, and then processed the period before its due-date.

□

For computing the mean level of finished goods inventory, repeated substitution of z_t in (2.32) yields:

$$\begin{aligned} z_{t+1} &= z_t + p_t - \hat{D}_{t+1} \\ &= z_{t-1} + p_{t-1} - \hat{D}_t + p_t - \hat{D}_{t+1} \\ \vdots \\ &= \sum_{j=0}^{t-1}(p_{t-j} - \hat{D}_{t+1-j}) + z_0 \\ &= \sum_{j=0}^{\infty}(\alpha^T \sum_{i=0}^{N-1}(C\hat{A})^i D_{t+1-i-j} - \hat{D}_{t+1-j}) \end{aligned} \tag{2.39}$$

where the expression for p_{t-j} is derived above, and we assume that $z_0 = 0$.

Property 3 *Taking expectations on (2.39), we find*

$$E[z_{t+1}] = \sum_{j=2}^{N} b_j \overline{D}^j \tag{2.40}$$

$$\begin{aligned} \text{where} \quad b_j &= b_{j-1}(1-\alpha_j) + (j-1)\alpha_j \\ \text{and} \quad b_1 &= 0 \end{aligned}$$

The argument for the validity of this expression is similar to that of the production variance. For example, a portion of D_t^3 is held for 2 days after processing, another portion is held for 1 day after processing, and the remainder is processed immediately prior to delivery. The factors corresponding to these are embedded in the b_3 value. □

From here, rather than consider the entire optimization program all at once, we proceed in a sequential fashion. We examine in turn each of the individual terms and then the full program. This will enable us to characterize completely a type of solution and then to indicate how it is affected by the introduction of conflicting objectives. In particular, we want to decide on how to set the α_i values by taking into account how the performance measures depend upon the parameters.

The case in which there are only holding costs to minimize is simple. Then, we consider only the smoothing properties of a policy, as measured by the variance in the production level. Finally, we consider the full program and see how to set the parameters when the objective function has the two conflicting terms.

Minimization of Holding Cost: $\Pi(E[z_t])$

Consider only having to incur a finished goods inventory costs:

$$\min_{P_t} gE[z_t] \qquad\qquad \Pi(E[z_t])$$

s.t.

$$0 \leq \alpha_i \leq 1 \quad i = 1, \ldots, N$$

$$\alpha_1 = 1$$

To solve this program we argue as follows:

Consider the feasible policy

$$\alpha_1 = 1, \quad \alpha_j = 0, \; j = 2, \ldots, N$$

By Equation (2.40) we have that $b_j = 0 \; \forall j$. In this case, $E[z_t] = 0$ and this must be an optimal policy, since we know that $z_t \geq 0$.

This policy corresponds to delaying the processing of a job until the period before its due-date. There are no penalties associated with the production level, and thus, since the system is uncapacitated

Minimization of Smoothing Costs: $\Pi(Var[p_t])$

In this case, there are only costs associated with the variability in production level The program corresponding to this is:

$$\min_{\alpha} fVar[p_t] \qquad\qquad \Pi(Var[p_t])$$

s.t.

$$0 \leq \alpha_i \leq 1$$
$$\alpha_1 = 1$$

Because f is a constant in the optimization, we will assume that $f = 1$.

Property 4 $\alpha_i = \frac{1}{i} \Rightarrow a_i = \frac{1}{i}$

We proceed inductively:

The property is true for $i = 1$: $\alpha_1 = 1$ and $a_1 = 1$

Assume true for $i = n - 1$: $\alpha_{n-1} = \frac{1}{n-1}$ and $a_{n-1} = \frac{1}{n-1}$

For $i = n$:

$$\begin{aligned} a_n &= (1-\alpha_n)^2 a_{n-1} + \alpha_n^2 \\ &= (1-\frac{1}{n})^2 \frac{1}{n-1} + (\frac{1}{n})^2 \\ &= \frac{1}{n} \end{aligned} \qquad (2.41)$$

□

When $\alpha_i = \frac{1}{i}$ then from (2.41),

$$Var[p] = \sum_{i=1}^{N} \frac{1}{i}\sigma_i^2 \qquad (2.42)$$

Property 5 $a_i = \frac{1}{i}$ *minimizes*

$$\sum_{i=1}^{N} a_i \sigma_i^2$$

$$\begin{aligned} \textit{where} \quad a_i &= (1-\alpha_i)^2 a_{i-1} + \alpha_i^2 \\ a_1 &= 1 \end{aligned}$$

The recursion

$$a_i = (1-\alpha_i)^2 a_{i-1} + \alpha_i^2 \qquad (2.43)$$

is a convex function of α_i for $a_{i-1} \geq 0$. Since $a_1 = 1 > 0$, the sequence $\{a_i\}$ generated by (2.43) is non-negative for $0 \leq \alpha_i \leq 1$. For a fixed value of a_{i-1} (generated via the sequence $a_{i-2}, a_{i-3} \ldots a_1$), (2.43) is minimized for

$$\alpha_i = \frac{a_{i-1}}{1 + a_{i-1}} \qquad (2.44)$$

	$\sigma_i^2 = \frac{1}{i}$, $\forall i$ $\alpha_i = \frac{1}{i}$			$\sigma_i^2 = 1$, $\forall i$ $\alpha_i = \frac{1}{i}$			$\sigma_i^2 = i$, $\forall i$ $\alpha_i = \frac{1}{i}$		
N	$\sum \sigma_i^2$	$Var(p_t)$	$\Delta\%$	$\sum \sigma_i^2$	$Var(p_t)$	$\Delta\%$	$\sum \sigma_i^2$	$Var(p_t)$	$\Delta\%$
3	1.83	1.36	26	3	1.83	39	6	3	50
5	2.28	1.46	36	5	2.28	54	15	5	67
10	2.93	1.55	47	10	2.93	71	55	10	82

Table 2.1: % Reduction production variance $\Delta\%$, $\alpha_i = \frac{1}{i}$ vs input

Substituting this into (2.43),

$$a_i = \frac{a_{i-1}}{1 + a_{i-1}} \tag{2.45}$$

Choosing $\alpha_i = \frac{a_{i-1}}{1+a_{i-1}}$ and with initial value $a_1 = 1$, the entire sequence $\{a_i\}$ follows with each $a_i = \frac{1}{i}$. To show optimality: In (2.43), a_i is increasing in a_{i-1} and so $\min a_i$ follows when $\{\min a_{i-1}\}$ is set, and α_i is given by (2.44).

To get a sense for what the exact value of the objective function ($\Pi(Var(p_t))$ is, we evaluate it for a few specific values of N at the optimal solution, $\alpha_i = \frac{1}{i}$.

Our choices for σ_i^2 in Table (2.1) above indicate the relative quantities of the variance of the individual demand streams. $\sigma_i^2 = 1$ represents the equai level of variability in each of the demands. $\sigma_i^2 = i$ captures the increasing level of uncertainty of demand streams with long due-date window requests. Conversely, $\sigma_i^2 = \frac{1}{i}$ associates the most variance with demand with very short due-date windows. For this last case, the least smoothing is possible, as we see comparing the $\Delta\%$ values, denoting the percentage decrease in the output stream versus the incoming demand.

The striking observation about the result we have here is that the optimal α_i is independent of the value of σ_i^2. At first, this seems counterintuitive; however, the structure of the recursion function is such that all of the σ_i^2 terms drop out and the policy is simply a function of the requested due-date leadtime.

Minimization of costs of variance and finished goods inventory

Suppose now we append the finished goods costs to the objective function. We expect that the optimal parameter values $0 \leq \alpha_i \leq \frac{1}{i} \forall i$, with the introduction of this conflicting term in the objective function. This follows because the inventory term will seek to delay the production of work so as to not incur any storage costs. Analytically, the problem becomes harder because the inclusion of this inventory term in the objective function disrupts the nice structure we had when we only considered one of the terms individually.

Formally, the program under consideration is:

$$\min_{\alpha} f \sum_{i=1}^{N} a_i \sigma_i^2 + g \sum_{i=1}^{N} b_i \overline{D}^i \qquad \Pi(Var[p_t], E[z_t])$$

s.t.

$$0 \leq \alpha_i \leq 1$$
$$\alpha_1 = 1$$

where the a_i, b_i terms are generated by

$$\begin{aligned} a_{i+1} &= a_i(1-\alpha_{i+1})^2 + \alpha_{i+1}^2 & a_1 = 1 \\ b_{i+1} &= b_i(1-\alpha_{i+1}) + i\alpha_{i+1} & b_1 = 0 \end{aligned}$$

As we have noted above, this program is difficult to solve to optimality because of the complete dependence of the a_i, b_i terms on each other. We have to resort to some bounding procedures to find good solutions.

Bounds on the Optimal Solution and a Sequential Search Heuristic An upper bound to the program $\Pi(Var[p_t], E[z_t])$ is given by the evaluation of any feasible set of $\{\alpha_i\}$. In this section we want to suggest reasonable bounds and discuss how to improve upon them. To do this, we make assumptions about the actual or relative values of the constants $f, g, \sigma_i^2, \overline{D}^i, i = 1, \ldots, N$. In particular, we assume that $f = g = 1$.

Upper Bounds Plugging the set $\alpha_i = \frac{1}{i}$ into $\Pi(Var[p_t], E[z_t])$ we find that the value of the objective function is:

$$\sum_{i=1}^{N} \frac{1}{i}\sigma_i^2 + \frac{1}{2}\sum_{i=1}^{N} \frac{(i-1)}{i}\overline{D}^i \tag{2.46}$$

Because $\alpha_i = \frac{1}{i}$ is feasible, we have that (2.46) is an upper bound on the optimal solution. Additional assumptions about σ_i^2 and $\overline{D}^i$ lead to more specific bounds. For example, we evaluate (2.46) by assuming values for the parameters associated with a particular demand stream $\overline{D}_t^i$, σ_i^2. As could be expected, this is not a very strong bound.

The next heuristic we consider we call a sequential search heuristic. In evaluating the terms corresponding to the jth demand stream, we have:

$$\begin{aligned} a_{i+1} &= a_i(1-\alpha_{i+1})^2 + \alpha_{i+1}^2 \\ b_{i+1} &= b_i(1-\alpha_{i+1}) + i\alpha_{i+1} \end{aligned}$$

The choice for α_{i+1} is a function of $[\alpha_1, \alpha_2, \ldots, \alpha_i]$, but not a function of any α_j $j > i+1$. Since

$$a_{i+1}\sigma_{i+1}^2 + \beta_{i+1}\overline{D}^{i+1} = (a_i(1-\alpha_{i+1})^2 + \alpha_{i+1}^2)\sigma_{i+1}^2 + (b_i(1-\alpha_{i+1}) + i\alpha_{i+1})\overline{D}^{i+1}$$

is convex in α_{i+1}, we propose doing a local 1-directional search of the form:

- Initialize: $\alpha_1 = 1, a_1 = 1, b_1 = 0$
- Given fixed $[\alpha_1, \alpha_2, \ldots, \alpha_i]$
- Solve

  $$\min_{\alpha_{i+1}} a_{i+1}\sigma^2_{i+1} + \beta_{i+1}\overline{D}^{i+1}$$

  which corresponds to

  $$\alpha_{i+1} = \frac{2a_i\sigma^2_{i+1} + (b_i - i)\overline{D}^{i+1}}{2(a_i + 1)\sigma^2_{i+1}}$$

  We can guarantee that $a_i < 1$ for $i > 1$. This follows because in the ratio given for α_{i+1}, $(b_i - i) \leq 0$for all i, and the ratio $\frac{2a_i\sigma^2_{i+1}}{2(a_i+1)\sigma^2_{i+1}} = \frac{a_i}{(a_i+1)} < 1$. Thus, to guarantee feasibility

  $$\alpha^*_{i+1} = \max\{0, \frac{2a_i\sigma^2_{i+1} + (b_i - i)\overline{D}^{i+1}}{2(a_i + 1)\sigma^2_{i+1}}\}$$

Lower Bounds Finding a lower bound to the program is more difficult. This is because the traditional approach is to dualize a constraint and then apply some kind of updating scheme, such as subgradient steps or dual-ascent methods. Unfortunately, the constraint that is the most desirable to relax, $\alpha_1 = 1$, is difficult to relax without developing an entirely new model. The difficulty arises because of the recursive constraints– relaxing one constraint effectively relaxes all of them!

One lower bound we can get is the optimal solution to either the smoothing problem or the inventory problem in isolation. However, the inventory problem is trivial because the optimal cost of that problem is 0– recall that the optimal α results in all work being produced one period before its due-date.

With smoothing as the sole objective, we found $\alpha_i = \frac{1}{i}$ to be the optimal parameter values. In particular, we now offer that the optimal value of that program is trivially a *lower bound* on the optimal value of $\Pi(Var[p_t], E[z_t])$. This follows because we have minimized the variance function and so introduction of any non-negative valued function of inventory has to be at least as great.

However, as one would expect, this lower bound is pretty poor when inventory holding costs are significant. When $f = g = 1$, this equates the importance of the costs due to smoothing and holding inventory. When $f = 1$ and $g \rightarrow 0$ it becomes a good bound, and in fact, all three of the solutions are equal.

However, in terms of comparison studies, we see how the sequential heuristic (which yields feasible solutions– $\alpha_1 = 1$ is the intialization value) outperforms the values corresponding to the $\Pi(\alpha_i = \frac{1}{i})$ solution. In the sequence of figures that follow, Figures (2.2.7)-(2.2.7), we assume that $f = g = 1$ and that $\overline{D}^j = 1 \forall j$. Then, we vary the σ_i^2 levels. For a given value of N, there are N demand classes each with $\overline{D}^j = 1$ and different variance levels. The figures depict the value of $\Pi(Var[p_t], E[z_t])$ according to the two solutions.

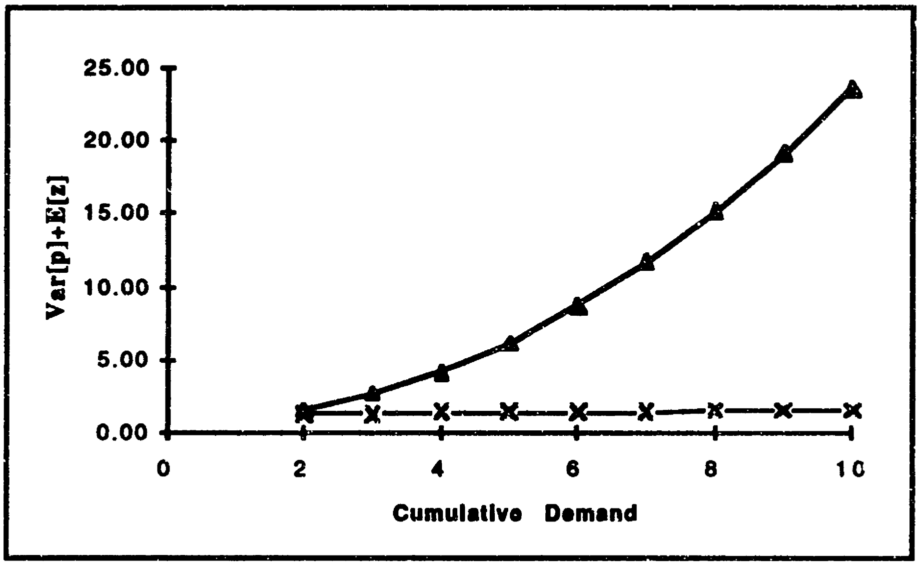

Figure 2-3: $\Pi(Var[p_t], E[z_t]) : \sigma_i^2 = \frac{1}{i^2}, \overline{D}^i = 1$

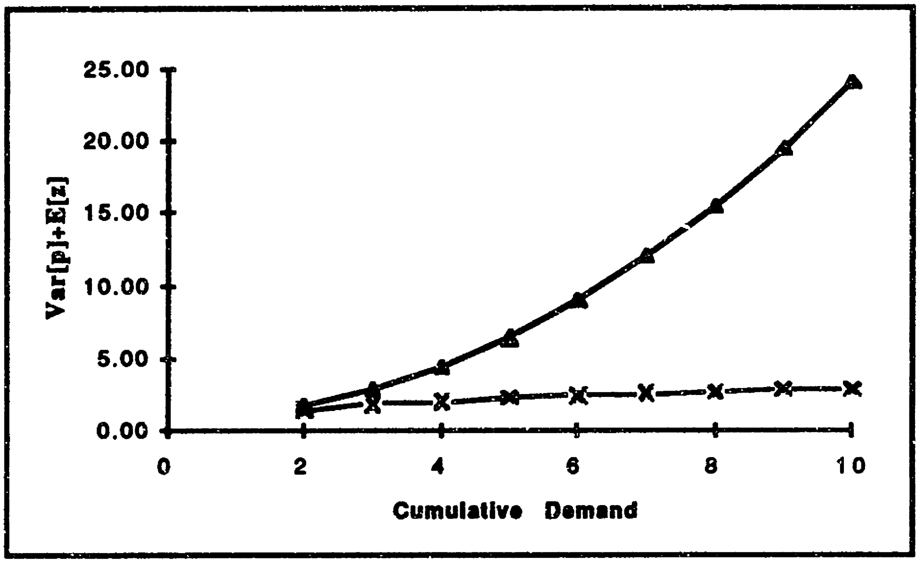

Figure 2-4: $\Pi(Var[p_t], E[z_t]) : \sigma_i^2 = \frac{1}{i}, \overline{D}^i = 1$

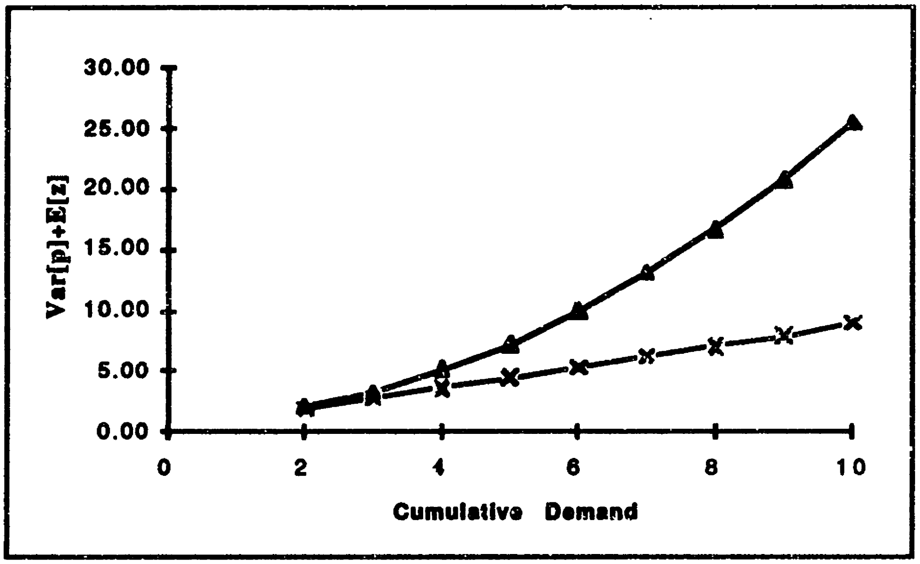

Figure 2-5: $\Pi(Var[p_t], E[z_t]) : \sigma_i^2 = 1, \overline{D}^i = 1$

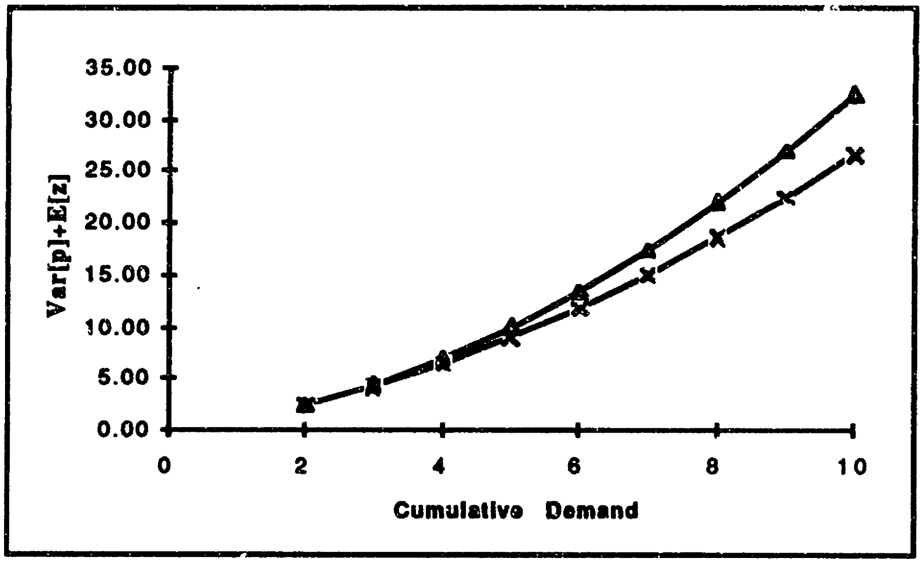

Figure 2-6: $\Pi(Var[p_t], E[z_t]) : \sigma_i^2 = i, \overline{D}^i = 1$

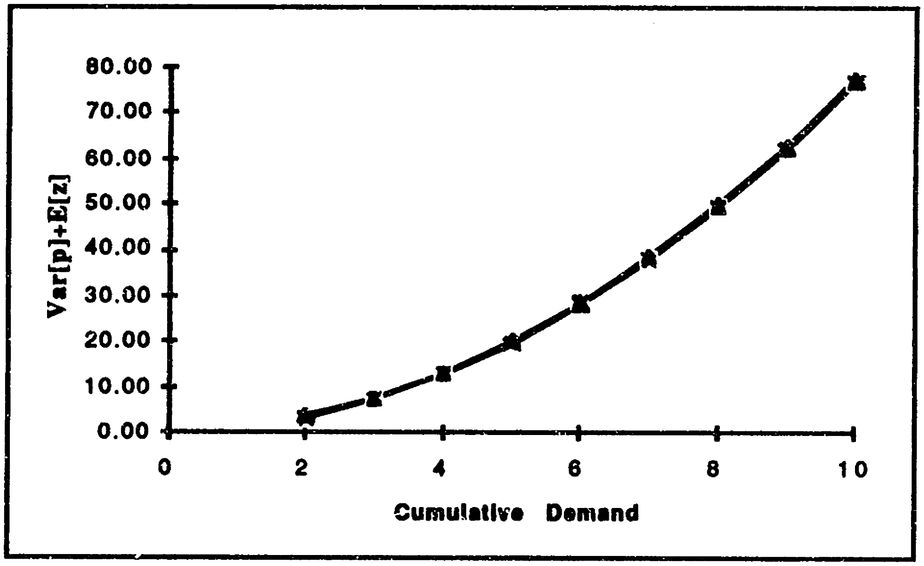

Figure 2-7: $\Pi(Var[p_t], E[z_t]) : \sigma_i^2 = i^2, \overline{D}^i = 1$

Next, we briefly consider a different type of policy which we refer to as an assignment policy. We show that the linear policy we have described here is essentially a fixed assignment of work upon entry to the system. This is counterintuitive to the construction that the production level is a function of the current work in queue.

An assignment policy

We now describe a policy in which production levels are set but *ignore* the current workload queue. The policy that we described above sets the production level as a linear function of the unfinished workload. Here, we propose a policy in which all of the incoming demand in period t, D_t^i, $i = 1, \ldots, N$ is permanently and feasibly scheduled for production at time $\underline{t}$.

We introduce some additional notation:

- X_t^i: Quantity slated for production in period $t+i$. Because work is permanently assigned to a period, a feasible work assignment consists of jobs that have due-dates that can vary from $t+i+1$ to $t+N-i$. The vector $X_t = \{X_t^i\}$ then represents the assignments over the planning horizon as of period t.
- Y_t^i is the assignment made to period $i+t$ in period t from the incoming demand. We have that $X_t^i = X_{t-1}^{i+1} + Y_t^i$. The vector of this assignment is given by $Y_t = \{Y_t^i\}$.

The motivation for this type of policy arises because there are often rigidities in the system, making it difficult to re-schedule work. We assume the assignment takes a particular form. To make an interesting basis for comparison, we consider a linear policy:

$$Y_t = BD_t \tag{2.47}$$

where $B = \{b_{ij}\}$ and b_{ij} is the quantity of D_t^i assigned to period $t+j$. More formally, this type of policy follows from a dynamic programming model similar to that used in the development of the workload policy of the previous section.

Feasibility of this type of policy is guaranteed by :

1. $b_{ij} \geq 0$
2. $\sum_{j=1}^{N} b_{ij} = 1$
3. B upper triangular, i.e., $b_{ij} = 0 \; i > j$

The first two items ensure that all work is assigned and the production level is non-negative. The last constraint in conjunction with the first two then satisfies due-date requirements. A matrix B that satisfies these constraints is a feasible assignment.

Theorem 1 *Every assignment policy of the form $Y_t = BD_t$ with B a feasible assignment matrix determines a unique independent workload policy of the form $p_t = \alpha^T W_t$.*

Initially, this seems surprising, since the first policy $p_t = \alpha^T W_t$ explicitly considers the cumulative work ordered by due-date, and the second policy, $Y_t = BD_t$ ignores the current state of the system. However, to see this, consider the following: Suppose we have a demand quantity D_t^i. According to the workload policy we will produce α_i of it in period t, leaving $(1-\alpha_i)D_t^i$ unfinished. In period $(t+1)$, α_{i-1} of the $(1-\alpha_i)D_t^i$ remainder is processed. This is equivalent to, in period t itself assigning $\alpha_i D_t^i$ for production in period t, $\alpha_{i-1}(1-\alpha_i)D_t^i$ for production in period $(t+1)$ etc. In other words, $b_{1,i} = \alpha_i, b_{2,i} = \alpha_{i-1}(1-\alpha_i), \ldots b_{i,i} = \alpha_1(1-\alpha_2)\ldots(1-\alpha_i)$. Thus, given a set of $\{\alpha_i\}$ we can generate the b_{ij}.

2.2.8 Computational Testing

In this chapter we have undertaken a modelling exercise and an extensive derivation which has suggested the use of a linear loading policy. In the literature, the linear policy is often used, justified for its analytic tractability and its ease in implementation. How well does this policy perform in practice? How do other dynamic policies compare? The optimal α vector for $\Pi(Var[p_t])$ was independent of σ_j^2 for all demand streams D_t^j. This is an appealing, robust policy since it requires no information about the variance of the input demand stream for setting parameter values. Is it possible that we could do better by explicitly looking at the queue? We proposed a sequential heuristic for solving $\Pi(Var[p_t], E[z_t])$. Unlike $\Pi(Var[p_t])$, the solution here is given as a function of the mean and variance of the input demand streams.

In this section we discuss some computational studies that might be worth pursuing for certain problem specific circumstances. Of special interest here is $\Pi(Var[p_t])$. For one, in this 1-machine context the presence of finished goods inventory may not be very crucial to the overall economic value of the system. Historically, as we have noted previously, the smoothing problem has been somewhat interchangeable with the planning problem for the reason that the costs associated with variation in production level are significantly greater than those for holding inventory. We defer computational work for $\Pi(Var[p_t], E[z_t])$ to the reader with a specific problem instance.

We propose a heuristic that has as its motivation the following: We have at our disposal two types of information. For one, we know the mean demand level of the input demand streams. This is a long-run type of quantity. However, in the short-run, we can look at the current workload level of the system as a factor in setting the production level. In the policy that we propose we want to maintain a balance between the long term information we know about the demand process and more timely information about the actual realization of the current queue length.

Assume that there is only one demand stream. We know that the inputs for D_t^j are "approximately" equal to the mean, although they may vary significantly from $\overline{D}^j$. In a more myopic sense suppose we want to exploit information about the current status of the queue. Intuitively, over a window of j periods we want to produce "on average" $\frac{\sum_{i=j}^{N} W_{it}^j}{j}$ during each of those periods. Cruickshanks et al [84] did a simulation study of this latter policy, in the case that there are no due-date restrictions.

To quantify these observations, we propose a parametric heuristic where the parameter weights the two types of information at hand to determine the production level.

We make the following assumptions in our simulation study:

- Dispatch policy is FIFO. We need this (or some dispatch rule in order to update the workload queue)
- The distribution function for the input demand stream is known and given

We denote the policy we have described here by $\mathcal{G}$. It has the form:

$$\gamma\frac{\sum_{i=j}^{N} W_{it}^j}{j} + (1-\gamma)\overline{D}^j$$

where $0 \leq \gamma \leq 1$ is a relative weight on the two types of information that we possess.

Additionally, we need to ensure that all work meets its due-date constraints. To satisfy this

$$P_t^j = \max\{\gamma\frac{\sum_{i=j}^{N} W_{it}^j}{j} + (1-\gamma)\overline{D}^j, W_t^1\} \qquad \mathcal{G}$$

The study here simulates the arrivals to the system of a particular demand process, and then sets the parameter γ at a level to minimize the variance in the actual production level when the production rule is given by $\mathcal{G}$. In particular, we briefly describe the form of the simulation:

```
Initialize:
        Set $\gamma = .05$
        Select demand stream $D_t^j$, i.e., the leadtime request on the input stream
        Select $\overline{D}^j$ and $\sigma_j^2$
Iteration
        Generate $D_t^j t = 1, \ldots, T$, $T$ =Length of the simulation horizon
        For $t > t_{min}$ calculate production variance
        Update minimum production variance
        Update $\gamma \rightarrow \gamma + .05$
        Update $\overline{D}^j$ and $\sigma_j^2$
```

This simulation essentially loops through the possible values for the mean and variance of a demand stream and then updates the γ factor. For each we calculate the production variance after time t_{min}, which is the time at which steady-state values can be computed.

The form of the distribution function for arrivals is a factor in the simulation. We have examined this policy when the demand process has either a Poisson or Normal distribution. In Figure (2.2.8) we depict the differences in the variance in production level in three scenarios for D_t^5. First, we have the minimum variance in production under the linear policy that is distribution independent. Then we have the minimum production variance values when the demand streams are Poisson and Normal. For the Poisson distribution, the mean and variance of the function are the same; we considered values for $\overline{D}^5 \in \{1, 2, \ldots, 10\}$. To be an interesting comparison, we specified the Normal distribution such that $\overline{D}^5 = \sigma_5^2$, and the simulation also runs over the same range of values for the parameters. In Figure (2.2.8) we depict a similar case, now with the demand stream D_t^7.

In particular, note that the linear policy falls somewhere between the minimum variance achieved. It performs better than the Poisson, but not as well as the Normal.

Comments on Computational Work

In modelling any dynamic physical system, there are a large number of possibilities for doing simulation studies. In the problem we have been discussing, to do any significant computational work we need to devise the simulation to be problem specific. The problem we have posed is a fairly general system with an objective that is relatively aggregate in scope. In using it, there needs to be detailed discussion about setting the parameters that weight the smoothing and inventory terms of the program $\Pi(Var[p_t], E[z_t])$. We have really only considered a simulation of a different type of "policy"(really just a rule that captured a trade-off between two perspectives on setting production level) for the program $\Pi(Var[p_t])$. The more general problem can be very context-specific, and we defer the study to later more computationally motivated projects. We did some preliminary computational work for $\Pi(Var[p_t])$ because of the unique feature that the optimal policy is independent of the moments of the demand process. We found then, in some cases it performs better than the simple parametric rule we devised, but that this is very much a function of the distribution function of the input stream.

2.2.9 Conclusions of the Basic Rigid Due-DateModel

In this section our objective was to describe and model the problem of developing a production loading policy in the presence of rigid due-date constraints. We did this by considering a restricted form of model in a finite horizon dynamic program. This

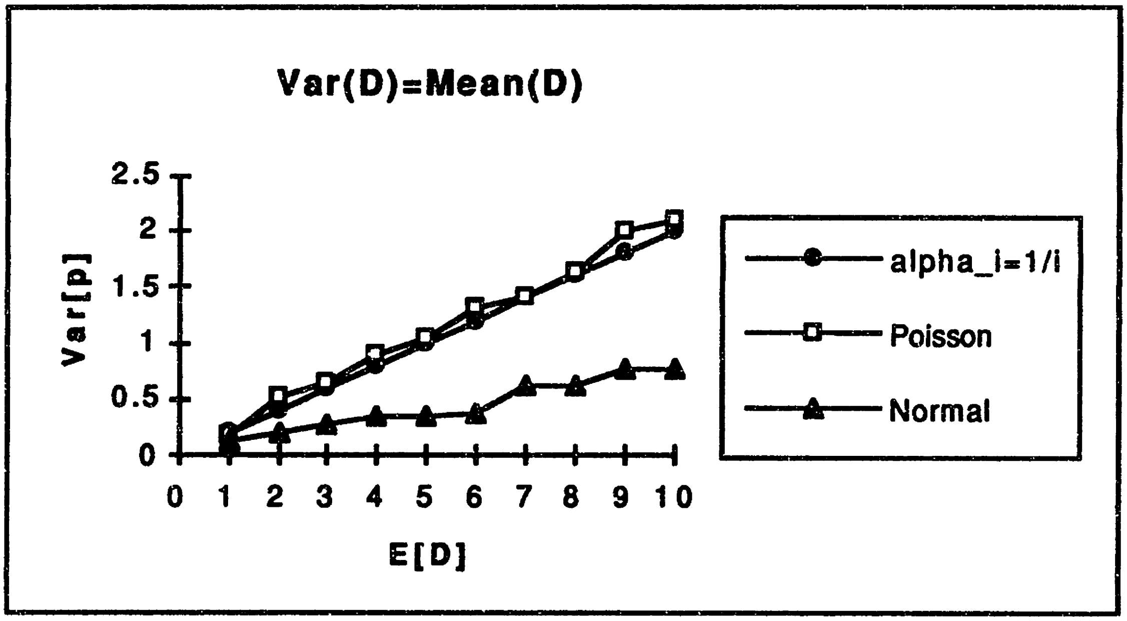

Figure 2-8: $Var[p_t]$ under $\alpha_i = \frac{1}{i}$, $\mathcal{G}$ for Poisson, Normal $\sim D_t^5$

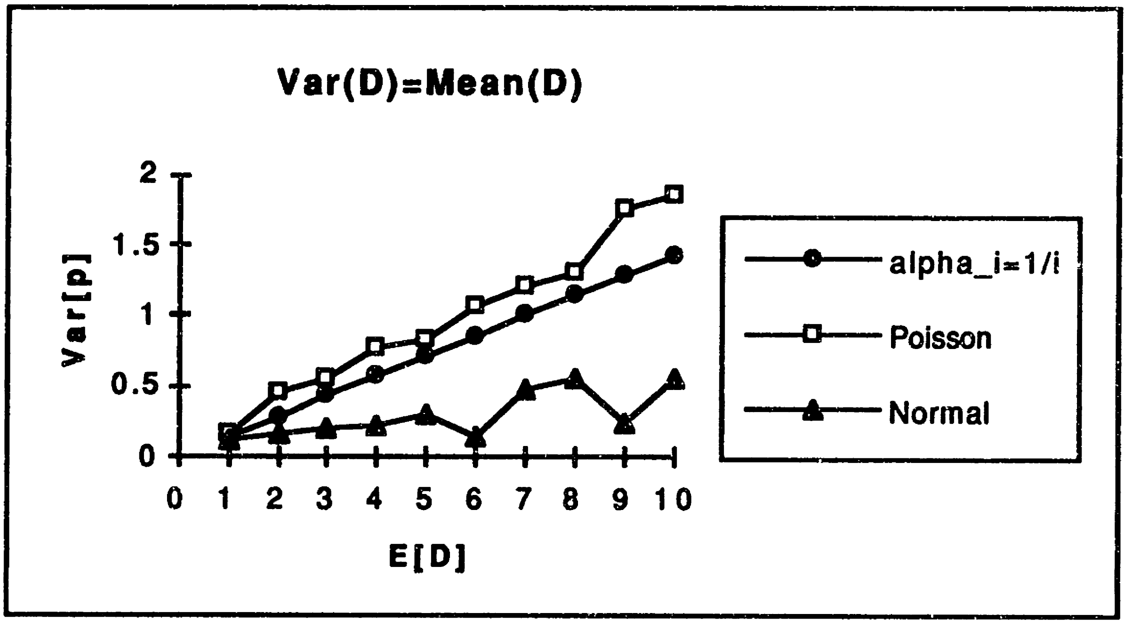

Figure 2-9: $Var[p_t]$ under $\alpha_i = \frac{1}{i}$, $\mathcal{G}$ for Poisson, Normal $\sim D_t^7$

suggested a linear production policy. We then did an analysis of the problem for the infinite horizon problem when this type of policy is implemented.

In the next sections, we will consider two types of extensions. First, we relax the rigidity of due-dates, and permit due-dates to be violated, with some associated cost for the late delivery of an order. Then, we see how to use the results of the one-machine system in the context of a larger type of manufacturing system with rigid due-date constraints. In both extensions we analyze the type of linear policy that we have introduced here.

Chapter 3

Non-Rigid Due-Dates

3.1 Problem Description

We have developed a loading policy for the one machine system that has a certain cost objective as a function of the machine's output process, and must satisfy rigid, exogenously-given due-dates. The assumption of rigidity reflects a market in which there is a relatively strong influence of the customer vis a vis the supplier. Consider now a situation in which the producer is relatively stronger than the customer in a market. This might arise as follows: The customer is smaller than the average customer, and as a result, exerts less influence. However, if the supplier is not a monopolist in this market, there is always a possibility that the customer will change to another supplier, albeit at a cost. In this scenario, satisfying the due-dates for the customer is not as crucial, but there is still incentive to meet the due-date requests in reasonable time. We refer to these non-rigid due-dates as being *violable* due-dates.

Previously, the production process was driven simultaneously by constraints to meet the demand due-dates, and the objective function to minimize costs. In the models we consider now, there is no longer a rigid due-date constraint. We incorporate an incentive for processing work via the objective function directly: There is no additional cost if the order is satisfied by its requested due-date; otherwise, the cost is increasing in the completion time.

We need to modify some of the assumptions and incorporate additional notation for the modelling of work with violable due-dates. We first present all of the notation that may be of interest in this section and then discuss several models.

3.1.1 Preliminaries

Notation

The major distinction between the model for the rigid and violable cases is that for the violable scenario we will be maintaining separate queues for the individual demand streams D_t^j. Previously the queues were indexed by the remaining due-date slack time.

Much of the notation we will use in this section is the same as that used for the rigid due-date model– in particular, those used to denote the demand processes and the production policy. However, we assume that the reader knows that the context here is one of non-rigid due-dates. As follows we detail the additional notation we will be using.

- We refer to D_t^j as *demand class* j, corresponding to the work that arrives in period t with a j-day due-date leadtime request. For example, $D_{t-1}^j, D_{t-2}^j, \ldots$ are also members of this class.
- $T_t^j \in \mathcal{Z}^+$ is a random variable for the length of time it takes for demand stream

D_t^j to complete processing. Specifically, suppose the quantity that enters in period t with a j-day leadtime is $\tilde{D}^j$. Here, T_t^j is the number of periods it takes for the entire quantity $\tilde{D}^j$ to be processed. If $T_t^j \leq j$, then the request is satisfied within the requested leadtime. If $T_t^j > j$, then the order is tardy. Incorporation of this tardiness component is the motivation of this chapter. $p_{T^j}(.)$ is the probability mass function associated with T_t^j.

- $U_t^j \in \mathcal{R}^+$ is the quantity of unfinished work belonging to demand class j at the beginning of period t. The vector of these values is $\{U_t^j\}$. Here we are not maintaining any ordering of work in terms of the remaining due-date slack as we did in the case of rigid due-dates; rather, we maintain a distinct queue for each demand class, with the idea of evaluating how long it takes work to make it through the queue, since it may be longer than its requested due-date.

  In this way, U_t^j differs fundamentally from W_t^j, which was used to keep track of unfinished work indexed by the slack remaining in the due-date window. As a consequence W_t^j was the amalgamation of several demand streams. U_t^j represents the unfinished work from one particular demand stream, each of which will be considered individually.

- $h_j(T_t^j) : \mathcal{Z}^+ \rightarrow \mathcal{R}$ is the lateness cost for members of D_t^j that took T_t^j periods to complete processing. Lateness can take on both negative and positive values.

Assumptions and Model

- Incoming demand streams are independent. We reiterate this assumption that we had previously because we will be implementing separate policies for the individual demand streams.

- We assume the lateness function has the form:

$$h_j(T_i^j) = \begin{cases} a_j(j - T_i^j) & T_i^j \leq j \\ b_j(T_i^j - j) & T_i^j > j \end{cases}$$
$$a_j, b_j \geq 0 \tag{3.1}$$

  According to this function, there are linear penalties for both early and late completion of work when there is a requested leadtime j. Thus, the penalty for producing work early, corresponding to a finished goods inventory cost, is decreasing and linear at rate a_j with respect to completion time $T_i^j = j$, at which point there is no penalty. Similarly, the costs associated with late production are linear at rate b_j.

  This form is very common in the literature for make-to-stock systems and seems appropriate for modelling the trade-off costs in our model. This exact form is not necessary for the development that follows. Penalties are assessed here on average behavior, whereas previously there was an infinite cost on late work which changed the structure of the problem.

As noted above, lateness can be either negative or positive(i.e., tardiness). In this way, we have modified the characterization of the inventory holding costs to be incorporated into this lateness function. Thus, early completion times incur holding costs as before.

3.1.2 Production Policy and Dynamics of System

As before, we will be analyzing a linear production policy. Previously, with the rigid due-date constraint, we set the production level as a linear function of the workload vector, indexed by due-date slack. Now, we are permitting work to be late, and so we modify the structure of the loading policy. In the case of rigid due-dates, we found that we could essentially aggregate demand streams by merging work according to the leadtime slack. This is not possible in this case, since one of the things we need to evaluate is the waiting time for each demand class. Thus, in our analysis, we propose a linear type of policy for an individual demand class; because we do not have the due-date restriction, we maintain a single queue of unfinished work, with the order of removal from the queue given by the dispatch policy. This method is also used in Cruickshanks et al[84] in which their primary objective is to assess the smoothing properties of a linear type of model. More detail of this is given in the Annotated Bibliography.

The linear policy for setting the production level is given by:

$$P_t^j = \beta_j U_t^j \tag{3.2}$$

where $0 \leq \beta_j \leq 1$ is a scalar valued parameter. As in the rigid model, we are still pursuing a production policy that is linear in the unfinised work.Previously, we constructed a rigid structure to ensure the satisfaction of due-dates. Now, without that restriction, we do not maintain information about the actual quantity of work with a particular due-date. Instead, we maintain information according to the individual demand classes.

Let $\beta = [\beta_1, \beta_2 \ldots, \beta_N]$ be the vector of the parameters for the N individual demand streams. Also, let $B = \ diag(\beta)$ and $\hat{B} = I - B$. This constraint reflects the "softness" of the due-dates; previously, we kept track of the due-date slack but in this type of rule we do not need the tight constraint.

The dynamics of the system under this type of linear rule are given by:

$$U_{t+1}^j = U_t^j - P_t^j + D_{t+1}^j \tag{3.3}$$

Substituting in the production policy, (3.2), this gives

$$\begin{aligned} U^j_{t+1} &= U^j_t - \beta^j U^j_t + D^j_{t+1} \\ &= (1-\beta^j)U^j_t + D^j_{t+1} \end{aligned}$$

In vector notation, this corresponds to

$$U_{t+1} = \hat{B}U_t + D_{t+1}$$

where $\hat{B} = diag((1-\beta^j))$.

We propose that our objective be to select a production policy from among the class of linear policies given by (3.2) that solves the following:

$$\min_{P_t} \sum_{j=1}^{N} fVar[P^j_t] + \sum_{j=1}^{N} E\{h_j(T^j_t)\}\overline{D}^j_t \tag{3.4}$$

s.t.

$$0 \le \beta_j \le 1 \quad \forall j$$

We consider two special cases of the program (3.4). In the first we assume that all work enters without any due date, but the cost function is increasing in its completion time. In the second model we consider a more general model in which work enters with non-identical due-date requests, but these dates are violable.

Assuming an infinite history of the process, we have the following lemmas:

Property 6 *Under a policy of the form given in (3.2)*

$$Var[P_t^j] = \frac{\beta_j}{2-\beta_j}\sigma_j^2 \ \forall j \tag{3.5}$$

Proof: Given

$$P_t^j = \beta_j U_t^j \tag{3.6}$$

we repeatedly substitute (3.4) into (3.6)

$$U_t^j = (1-\beta_j)U_{t-1}^j + D_t^j$$

to get

$$P_t^j = \beta_j \sum_{i=0}^{\infty}(1-\beta_j)^i D_{t-i}^j \tag{3.7}$$

Applying the definition of variance to this, and by independence of $D_t^j \ \forall t, j$, the result follows.□

Given the characterization of P_t^j above in (3.7), we can calculate some other quantities of interest:

Property 7

$$a.E[U_t^j] = \frac{\overline{D}^j}{\beta_j} \tag{3.8}$$

$$b.E[T_t^j] = \frac{1}{\beta_j} \tag{3.9}$$

Proof: a. By the same approach as above, we find that

$$U_t^j = \sum_{i=0}^{\infty}(1-\beta_j)^i D_{t-i}^j \tag{3.10}$$

Then,

$$\begin{aligned} E[U_t^j] &= E[\sum_{i=0}^{\infty}(1-\beta_j)^i D_{t-i}^j] \\ &= \sum_{i=0}^{\infty}(1-\beta_j)^i \overline{D}^j \\ &= \frac{\overline{D}^j}{\beta_j} \end{aligned}$$

b. When we apply Little's Law to this system, we get that

$$E[U_t^j] = \overline{D}^j E[T_t^j]$$

where $E[U_t^j]$ is the average queue length and $\overline{D}^j$ is the mean arrival rate for demand class j.□

Role of a dispatch rule

Although we have maintained our primary interest in loading policies, the violable due-date option forces us to consider the interaction of the loading scheme with the dispatch protocol. This was of peripheral interest in the case of the rigid due-dates because given a feasible loading policy there is no incentive for faster processing, although there may be interest in delay because of inventory holding costs.

The motivation for the specification of a dispatch policy is that we need to explicitly evaluate the distribution function of the waiting times for each demand class. The length of the wait then gives an indication of the costs that are incurred. Processing significantly before the due-date means that inventory costs are assessed, while processing after the requested due-date result in late delivery costs.

In this context, several different dispatch protocols may be of interest. We offer the following as examples of priority schemes which are a function of arrival time to the system:

- FIFO: Jobs are processed in order of earliest time of entry, i.e., " First-In-First-Out"
- LIFO: Most recent arrivals are processed first, i.e., " Last-In-First-Out"
- SIRO:"Service-In-Random-Order" – Jobs chosen at random according to a probability function

Dispatch protocols may also (and commonly) be given by other non-arrival time-based considerations. For example, it might be the case that there are economies to be gained by processing certain jobs concurrently. This might be because of savings in setups, maintenance, etc.

In this analysis we will be considering the SIRO and FIFO policies. FIFO is a very common type of dispatch policy. We show how to generate the mean waiting costs given the distribution function for the input demand stream, D_t^j. SIRO may be of somewhat general interest in the context that one of the non-arrival time based dispatch policies is being implemented. When one of these policies is being used, it may appear from an aggregate level that the work is being chosen arbitrarily. An additional benefit is that the policy constitutes an upper bound on the FIFO discipline, and requires no information about the distribution function for D_t^j. Then, we also discuss some lower bounds applicable to any dispatch policy.

3.1.3 Case 1: No due dates

Customers who do not quote a due date, but want the work done "as soon as possible", are implicitly quoting a soft due date of 1 period. Thus, in this model we have only one type of demand stream entering which does not have a specific due-date but seeks immediate processing. More concisely, we have exactly one incoming stream, D_t^1, the scalar demand entering in period t, with mean $\overline{D}^1$, and variance σ_1^2. No inventory is held since all work is delivered immediately. The trade-off is between production (smoothing) and waiting delays. Applying the expectation and variance operators, and making the simplifications corresponding to this special case, we obtain the optimization program (3.4):

$$\min_{\beta_1} fVar[\tilde{p}_t] + h_1 E[T_t^1]\overline{D}_t^1 \tag{3.11}$$

$$\text{s.t.}$$

$$0 \leq \beta_1 \leq 1 \tag{3.12}$$

Substituting (3.5) and (3.9) into (3.11), our optimization program becomes:

$$\min_{\beta_1} \; f\frac{\beta_1}{2-\beta_1}\sigma_1^2 + h_1\frac{1}{\beta_1}\overline{D}_t^1 \text{ s.t.}$$
$$0 \leq \beta_1 \leq 1$$

Result 1 *The optimal value of the parameter, β_1^* given by:*

$$\beta_1^* = \begin{cases} \frac{2\sqrt{h_1\overline{D}_t^1}}{\sqrt{2f\sigma_1+\sqrt{h_1\overline{D}_t^1}}} & \frac{h_1\overline{D}_t^1}{2f\sigma_1^2} \leq 1 \\ 1 & else \end{cases} \tag{3.13}$$

Proof: $\frac{\beta_1}{2-\beta_1}$ is convex and increasing on [0,1] as in Figure (3-1).

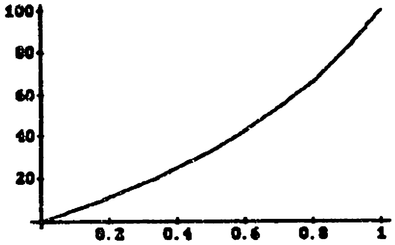

Figure 3-1: $\frac{\beta_1}{2-\beta_1}f = 50, h_1 = 5, \overline{D}_t^1 = 1, \sigma_1^2 = 1$

$\frac{1}{\beta_1}$ is convex and decreasing on [0,1] as in Figure (3-2)

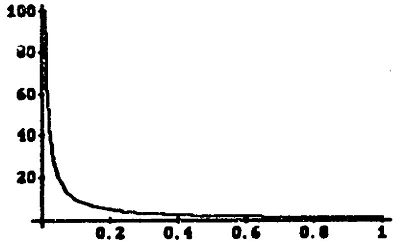

Figure 3-2: $h_1\frac{1}{\beta_1}\overline{D}_t^1$ $f = 50, h_1 = 5, \overline{D}_t^1 = 1, \sigma_1^2 = 1$

The minimum of the two terms will either occur in the interval or to the right of the interval, depending on the values of $f, h_1, \sigma_1^2, \overline{D}_t^1$.

Since the sum is convex the critical point yields the unconstrained minimum to (3.13). When we differentiate (3.13)

$$\frac{\partial}{\partial\beta_1}[f\frac{\beta_1}{2-\beta_1}\sigma_1^2 + h_1\frac{1}{\beta_1}\overline{D}_t^1] = f\sigma_1^2\frac{2}{(2-\beta_1)^2} - h_1\overline{D}_t^1\frac{1}{\beta_1^2} = 0$$

Solving for the positive root of β_1 we have

$$\begin{aligned}\beta_1 &= \frac{-4h_1\overline{D}_t^1 \pm \sqrt{16(h_1\overline{D}_t^1)^2 - 4(2f\sigma_1^2 - h_1\overline{D}_t^1)(-4h_1\overline{D}_t^1)}}{2(2f\sigma_1^2 - h_1\overline{D}_t^1)}\\ &= \frac{-2h_1\overline{D}_t^1+2\sqrt{2f\sigma_1^2h_1\overline{D}_t^1}}{2f\sigma_1^2 - h_1\overline{D}_t^1}\end{aligned}$$

$$= \frac{2\sqrt{h_1\overline{D}_t^1}(\sqrt{2f\sigma_1^2} - \sqrt{h_1\overline{D}_t^1})}{2f\sigma_1^2 - h_1\overline{D}_t^1}$$
$$= \frac{2\sqrt{h_1\overline{D}_t^1}}{\sqrt{2f\sigma_1^2} + \sqrt{h_1\overline{D}_t^1}}$$

If the minimum of the function occurs for $\beta_1 \geq 1$ then $\beta_1 = 1$ is the minimum on the interval. Equivalently,

$$\beta_1 \leq 1 \quad \Longleftrightarrow \quad \frac{h_1\overline{D}_t^1}{2f\sigma_1^2} \leq 1$$

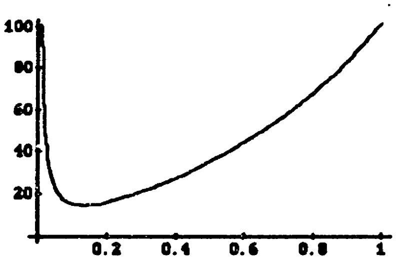

Figure 3-3: $f\frac{\beta_1}{2-\beta_1}\sigma_1^2 + h_1\frac{1}{\beta_1}\overline{D}_t^1$ $f = 50, h_1 = 5, \overline{D}_t^1 = 1, \sigma_1^2 = 1$

3.1.4 Case 2: D_t^j with Violable Due-date Requests

In this section we consider the general case with N demand classes. In Case 1 there were no holding costs. Now with N demand classes, i.e., due-date leadtimes $\in \{1, 2, \ldots, , N\}$, there are penalties for early production (holding costs) as well as for late production (tardiness costs).

For different values, the analysis will be the same. The assumption that $D_t^j \; \forall \; j \; t$ are independent, means that we can consider separately the optimization problem for each demand class. That is, the optimization problem for demand class j is:

$$\min_{\beta_j} fVar[\tilde{P}_t^j] + [\sum_{t=1}^{j} a(j-t)p_{T^j}(t) + \sum_{t=j+1}^{\infty} b(j-t)p_{T^j}(t)] \qquad \Pi_{original}$$

s.t.

$$0 \leq \beta_j \leq 1$$

In this case we know the variance term, but we still need to determine the distribution function for T_t^j. To characterize $p_{T^j}(t)$ we need to assume a form for the dispatch policy. Among the dispatch policies we have discussed, we consider two: FIFO and SIRO. As we will see, we need the entire distribution function of D_t^j for analysis of the FIFO policy. Alternatively, we can consider some bounds on this solution. The SIRO policy yields an upper bound on $\Pi_{original}$ and we also present an easily evaluated lower bound.

FIFO policy

In this section we want to evaluate the function $E[h_j(T_t^j)]$ under a FIFO policy, where $h_j()$ is given by the lateness function (3.1). To do this we proceed in the following way:

- Find the distribution functions for the queue length and for the "devolution" of the queue length
- Use the pdf of U_t^j to derive the probability mass function for T_t^j
- Evaluate $E[h_j(T_t^j)]$ and choose β_j to minimize Π_{FIFO}^j

This progression corresponds to the complete sequence of generating the cost contribution associated with T_t^j. However, it may be useful to bypass explicit derivation of one of these functions, instead assuming that it has a particular form and proceed through the remainder of the sequence. For example, we might want to *apriori* assume that the pdf of T_t^j is known and given; then we use this to evaluate $E[h_j(T_t^j)]$ directly and continue through the remainder of the sequence to choose the optimal β_j. We conclude with a numerical example of the full sequence of the derivation.

Distribution function for U_t^j

To find the distribution function for U_t^j we take the following approach: Suppose we observe the queue at time t and take a "snapshot" of its length. In successive periods we do not observe the queue itself, with respect to new arrivals. Instead, we track decrements in the snapshot queue, as work is processed each period. We then characterize the distribution function for this "devolution" of the queue and use it to characterize the distribution function for the queue length itself.

We begin with the basic equations of dynamics:

$$U_t^j = U_{t-1}^j - P_{t-1}^j + D_t^j$$

which as in (3.10) becomes

$$U_t^j = \sum_{k=0}^{\infty}(1-\beta_j)^k D_{t-k}^j \qquad (3.14)$$

Following the development in Graves[86], we introduce

$$U_t^j(r)$$

which represents the work remaining in the snapshot queue for demand class j at the beginning of period $t+r$ when the initial snapshot was taken at the beginning of period t.

$$
\begin{aligned}
U_t^j(r) &= U_{t-1}^j(r-1) - P_{t-1}^j && (3.15)\\
&= U_{t-r}^j(0) - \sum_{k=1}^{r} P_{t-k}^j && (3.16)
\end{aligned}
$$

with $U_t^j(0) = U_t^j$ from above.

In this way, $U_t^j(r) > 0$ corresponds to the queue associated with class j that has been there at least r periods. $U_t^j(r) < 0$ means that none of the current queue has been there for r periods and that a quantity of work equal to $|U_t^j(r)|$ has been processed in addition to the original $U_{t-r}^j(0) > 0$.

Substituting into (3.14) and (3.16) for the P_t^j terms and rearranging according to time indices we obtain

$$
U_{t-r}^j(0) = \sum_{k=0}^{\infty}(1-\beta_j)^k D_{t-k-r}^j \qquad (3.17)
$$

and

$$
\begin{aligned}
U_t^j(r) &= \sum_{k=0}^{\infty}(1-\beta_j)^k D_{t-k-r}^j - \sum_{s=1}^{r}\beta_j \sum_{k=0}^{\infty}(1-\beta_j)^k D_{t-s-k}^j \\
&= \sum_{s=0}^{\infty}(1-\beta_j)^{r+s} D_{t-r-s}^j - \sum_{k=1}^{r-1}(1-(1-\beta_j)^k) D_{t-k}^j && (3.18)
\end{aligned}
$$

Immediately from this, given $\overline{D}^j$ and σ_j^2, we can get the mean and variance of $U_t^j(r)$.

Property 8

$$
E[U_t^j(r)] = [\frac{1}{\beta_j} - r]\overline{D}^j \qquad (3.19)
$$

$$Var[U_t^j(r)] \;=\; [\frac{(1-\beta_j)^2}{2\beta_j - \beta_j^2} + (r-1)]\sigma_j^2 \tag{3.20}$$

To see these:

-

$$\begin{aligned}
&E[U_t^j(r)] \\
&\quad = E[\sum_{s=0}^{\infty}(1-\beta_j)^{r+s}D_{t-r-s}^j - \sum_{k=1}^{r-1}(1-(1-\beta_j)^k)D_{t-k}^j] \\
&\quad = [\sum_{s=0}^{\infty}(1-\beta_j)^{r+s} - \sum_{k=1}^{r-1}(1-(1-\beta_j)^k)]\overline{D}^j \\
&\quad = \{\sum_{k=1}^{\infty}(1-\beta_j)^k - \sum_{k=1}^{r-1}1\}\overline{D}^j \\
&\quad = [\frac{1-\beta_j}{\beta_j} - (r-1)]\overline{D}^j \\
&\quad = [\frac{1}{\beta_j} - r]\overline{D}^j \qquad \Box
\end{aligned}$$

-

$$\begin{aligned}
&Var[U_t^j(r)] \\
&\quad = Var[\sum_{s=0}^{\infty}(1-\beta_j)^{r+s}D_{t-r-s}^j - \sum_{k=1}^{r-1}(1-(1-\beta_j)^k)D_{t-k}^j] \\
&\quad = \{\sum_{s=0}^{\infty}[(1-\beta_j)^{r+s}]^2 + \sum_{k=1}^{r-1}1 + \sum_{k=1}^{r-1}[(1-\beta_j)^k)]^2\}\sigma_j^2 \\
&\quad = \{\sum_{k=1}^{\infty}[(1-\beta_j)^k]^2 + \sum_{k=1}^{r-1}1\}\sigma_j^2 \\
&\quad = [\frac{(1-\beta_j)^2}{1-[1-\beta_j]^2} + (r-1)]\sigma_j^2 \\
&\quad = [\frac{(1-\beta_j)^2}{2\beta_j - 2\beta_j^2} - (r-1)]\sigma_j^2 \qquad \Box
\end{aligned}$$

If we know the distribution function for D_t^j, then the characterization given in (3.18) allows us to determine the distribution of the queue length.

How difficult is it to find $f_{U^j(r)}(u)$? Consider the following: Let X be a random variable with probability density function $f_X(x)$, and finite mean μ and finite variance v^2. Given an infinite sequence of X_i's with $X_i \sim X$, let Y be a random variable defined by

$$Y = \gamma X_1 + \gamma^2 X_2 + \gamma^3 X_3 + \dots \tag{3.21}$$

$$0 \leq \gamma \leq 1 \tag{3.22}$$

Under what conditions can we apply a Central Limit type of theorem and conclude that (3.21) has a Normal distribution? The most traditional type of Central Limit Theorem holds for independent, identically distributed random variables. Here, the terms in (3.21) are independent but not identically distributed. If, however, each of the terms $\beta^i X_i$ satisfies Liapunov's condition, then the CLT may be employed. (A discussion of Liapunov condition is given in the Appendix. Loosely, we require that X_i, in addition to finite mean and variance, have some finite higher central moment, i.e., $E[|X_n - E[X_n]|^{2+\delta}], \exists\delta > 0$)

We use this result to characterize the distribution of $U_t^j(r), r = 0, 1, 2, \dots$ as given in (3.17) and (3.18). In particular we assume that Liapunov's condition is satisfied by the realizations of D_t^j. Applying CLT to (3.17) is straightforward. It follows that

$$U_t^j(0) \sim \mathcal{N}(\frac{\overline{D}^j}{\beta_j}, \frac{\sigma_j^2}{2\beta_j - \beta_j^2})$$

which holds regardless of the distribution of D_t^j. This characterization is useful because it yields the distribution function of the queue length.

Now, for $U_i^j(r)$ given as above,

$$U_i^j(r) = \sum_{s=0}^{\infty}(1-\beta_j)^{r+s}D_{i-r-s}^j - \sum_{k=1}^{r-1}(1-(1-\beta_j)^k)D_{i-k}^j$$

We can apply the CLT to the first term given here. What about the second term? There are two cases:

1. $D_i^j \sim \mathcal{N}(\overline{D}^j, \sigma_j^2)$
2. $D_i^j \sim f_{D^j}(d)$ where $E[D_i^j] = \overline{D}^j$ and $Var[D_i^j] = \sigma_j^2$.

Case 1: $D_i^j \sim \mathcal{N}(\overline{D}^j, \sigma_j^2)$ Because D_i^j is Normal, we can make use of the following:

Given two random variables X_1, X_2 where

$$\begin{aligned} X_1 &\sim \mathcal{N}(\mu_1, v_1^2) \\ X_2 &\sim \mathcal{N}(\mu_2, v_2^2) \end{aligned}$$

define

$$Y = X_1 - X_2$$

Then,

$$Y \sim \mathcal{N}(\mu_1 - \mu_2, v_1^2 + v_2^2)$$

In this way,

$$U_i^j(r) \sim \mathcal{N}([\frac{1}{\beta_j} - r]\overline{D}^j, [\frac{(1-\beta_j)^2}{2\beta_j - \beta_j^2} + (r-1)]\sigma_j^2)$$

Case 2: $D_i^j \sim f_{D^j}(d)$ *where* $E[D_i^j] = \overline{D}^j$ *and* $Var[D_i^j] = \sigma_j^2$

Again, evaluation of the first term in (3.18) is straightforward, with

$$\sum_{s=0}^{\infty}(1-\beta_j)^{r+s}D^j_{t-r-s} \sim \mathcal{N}([\frac{(1-\beta_j)^r}{\beta_j} - r]\overline{D}^j, [\frac{(1-\beta_j)^{2r}}{2\beta_j-\beta_j^2}]\sigma_j^2)$$

However, now that D^j_t is not Normal the evaluation becomes more involved. The pdf for the second term is derived by the $(r-1)$-fold convolution of each of the $(1-(1-\beta_j)^k)D^j_{t-k}k = 1, \ldots r-1$ elements. We require the exact form of $f_{D^j}(d)$, and then convolve the Normal first term with this latter function.

From this, then it would seem that it is preferable (given the option) to have $D^j_t \sim \mathcal{N}(\overline{D}^j, \sigma_j^2)$. In many cases it may be a reasonable assumption, and from Case 1, we see it is nicely tractable. However, there is an important caveat about $D^j_t \sim \mathcal{N}(\overline{D}^j, \sigma_j^2)$:

$$Pr\{D^j_t < 0\} > 0$$

i.e., there is positive probability for negative demand realizations, contradicting a preliminary assumption of the model. To resolve this one may need to do some other work in implementation.

Use the pdf of $U^j_t(r)$ to derive the probability mass function for T^j_t
Assume that $U^j_t(r)$ $r = 0, 1, \ldots$ has a known distribution function. Recall that T^j_t is a random variable for the length of time it takes for the entire quantity D^j_t to complete processing. If $T^j_t \leq j$, then the request is satisfied within the requested leadtime. If $T^j_t > j$, then the order is tardy.

The following holds:

Property 9

$$U^j_t(r) > 0 \iff T^j_t > r$$

yielding

$$Pr\{U_t^j(r) \geq 0\} = Pr\{T^j \geq r\}$$

$U_t^j(r)$ is the snapshot queue associated with class j at time t. In particular, at time t it includes the realization of D_t^j. We know that $U_t^j(r) > 0$ means that there is a queue of positive length associated with class j that has been there at least r periods. Since work is processed in a FIFO fashion, when $U_t^j(r) > 0$ this means that the entire quantity D_t^j cannot have been processed. Conversely, $T_t^j \geq r$ means that the snapshot queue at time t is not empty after r periods, i.e., $U_t^j(r) \geq 0$ □

Of particular interest, we note that $Pr\{U_t^j(r) \geq 0\}$ is decreasing in the value of r, so the probability of waiting a very long time becomes small. In Figure (3-4) we show this for $U_t^j(r) \sim \mathcal{N}(\mu_r, v_r^2)$.

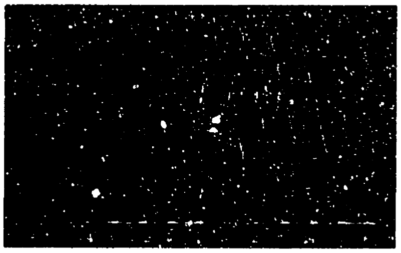

Figure 3-4: $Pr\{U_t^j(r) \geq 0\} \downarrow r$: Curve moves left with $r \downarrow$

Using this relation, we generate the complement of the distribution function for T_t^j:

$$\overline{F}_{T^j}(r) = Pr\{T^j \geq r\} = [1 - \sum_{t=1}^{r-1} p_{T^j}(t)] \tag{3.23}$$

Then, using elementary methods, we derive the probability mass function with

$$Pr[T_t^j = r] = \overline{F}_{T^j}(r) - \overline{F}_{T^j}(r+1)$$

Evaluate $E[h_j(T_t^j)]$ and choose β_j to minimize Π_{FIFO}^j

Given that the loading policy is given by $P_t^j = \beta_j U_t^j$, we expect that the $E[h_j(T_t^j)]$ term should be a function of β_j. To complete the analysis, then, we set the β_j level so that Π_{FIFO} is minimized. This latter, however, is difficult to do analytically, because of the analytic form of the Normal distribution. The optimization requires a computational analysis to set the β_j levels.

Example

We illustrate this derivation by considering the problem for demand stream D_t^5. For different values of β_5, we get different distribution functions for T_t^5, shown in Figures (3-5)-(3-8), with $\overline{D}^j = 1$, $\sigma_j^2 = 1$. In the case that β_5 is small, the waiting time distribution is Normal-looking, although it is truncated for values < 0. When β_5 is large $Pr\{T_t^5 > k\}$ gets small very quickly for values of $k > 2$.

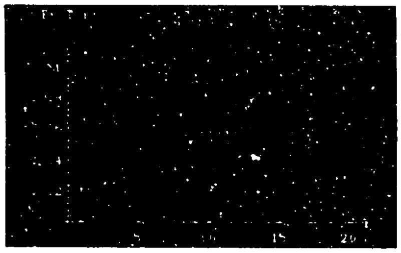

Figure 3-5: Distribution of T_t^5 $\beta_5 = .1$

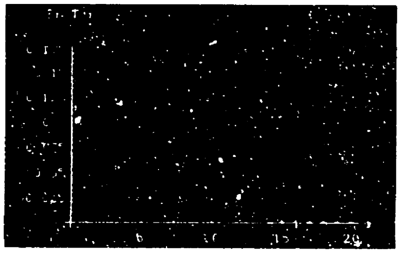

Figure 3-6: Distribution of T_t^5 $\beta_5 = .2$

Then, for a fixed value for β_5, we can calculate the expected value of Π_{FIFO}. To set the optimal value of the parameter, we must explicitly vary $\beta_5 \in [0,1]$. The

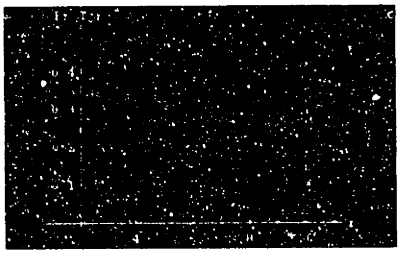

Figure 3-7: Distribution of T_t^5 $\beta_5 = .5$

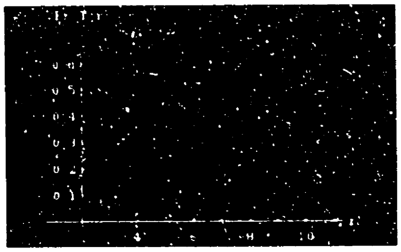

Figure 3-8: Distribution of T_t^5 $\beta_5 = .8$

resulting shape of the $\Pi_{FIFO}(\beta_5)$ curve is a function of the parameter values f, a, b. We illustrate these for some extreme parameter values in Figures (3-9)-(3-10).

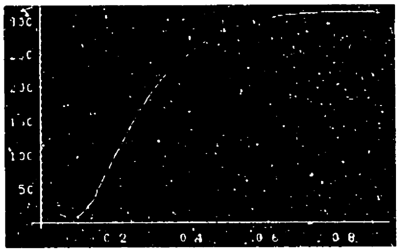

Figure 3-9: $\Pi_{FIFO} : f = 1, a = 100, b = 1$

Comments on FIFO In this section we have seen how to analyze the waiting time distribution for a demand class under a linear type of production policy. There were several limitations that we encountered:

Assumption of $D_t^j \sim \mathcal{N}(\overline{D}^j, \sigma_j^2)$ may not be appropriate if $\frac{\overline{D}^j}{\sigma_j} < 2$ or 3.

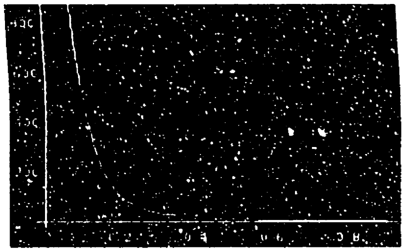

Figure 3-10: $\Pi_{FIFO} : f = 1, a = 1, b = 100$

When D_t^j is not Normal, then the evaluation of the distribution function for $U_t^j(r)$ may be difficult.

Optimal value for parameter β_j must be evaluated numerically instead of analytically.

Because the use of a FIFO policy may not always be appropriate or may be too difficult to evaluate, we consider another dispatch rule that serves as an upper bound on this solution and is more analytically tractable. Additionally, we offer a lower bound that is valid for any dispatch rule.

SIRO policy

Property 10 *When the dispatch policy is SIRO, the probability mass function for T_t^j is given by a geometric function of the form:*

$$p_{T^j}(t) = \beta_j(1-\beta_j)^{t-1} \quad t = 1,2,\ldots \tag{3.24}$$

Proof: Consider a unit of work in class j that enters in period t_0, denoted $\Delta_{t_o}^j$. We include it in the queue of unfinished work $U_{t_0}^j$. The loading policy specifies that β_j of $U_{t_0}^j$ is processed in period t_0. Since the dispatch policy is SIRO, work is chosen with uniform equal probability.

$$\begin{aligned} Pr\{\Delta^j_{t_o} chosen\} &= \int_{U^j} Pr\{\Delta^j_{t_o} chosen | U^j_{t_0} = u\} f_{U^j}(u) du \\ &= \int_{U^j} \frac{\beta_j u}{u} f_{U^j}(u) du \\ &= \beta_j \end{aligned}$$

Continuing in this way we see that this probability is independent of the value of U^j_t, $t = t_0, t_0 + 1, \ldots$, given that a particular unit of work is a member of U^j_t. In this way the probability of any unit of work being chosen is independent of the unfinished work in queue. Since

$$Pr\{\Delta^j_{t_o} chosen | \Delta^j_{t_o} \in U^j_t\} = \beta_j \ \forall t \tag{3.25}$$

we see that

$$\begin{aligned} Pr\{T^j_t = k\} &= Pr\{\Delta^j_t \text{chosen in period } t + k - 1\} \\ &= (1 - \beta_j)^{k-1} \beta_j \end{aligned}$$

Thus, T^j_t has a geometric distribution. Note that when the dispatch policy is SIRO, T^j_t is independent of D^j_t.

Given

$$p_{T^j}(t) = \beta_j (1 - \beta_j)^{t-1} \quad t = 1, 2, \ldots \ \forall j$$

we evaluate the $E\{h_j(T^j_t)\}$ as

$$\begin{aligned} E\{h_j(T^j_t)\} &= \sum_{t=1}^{j} a(j - t)\beta_j (1 - \beta_j)^{t-1} + \sum_{t=j+1}^{\infty} b(t - j)\beta_j (1 - \beta_j)^{t-1} \\ &= \frac{a[j\beta_j + (1 - \beta_j)^j - 1]}{\beta_j} + \frac{b(1 - \beta_j)^j}{\beta_j} \end{aligned}$$

Then the full optimization program for the jth stream is given by:

$$\min_{\beta_j} f\frac{\beta_j}{2-\beta_j} + \frac{a[j\beta_j + (1-\beta_j)^j - 1]}{\beta_j} + \frac{b(1-\beta_j)^j}{\beta_j}$$

s.t.

$$0 \leq \beta_1 \leq 1$$

This function is not convex for all values of f, a, b – See Figure(3-11) However, we can use numerical methods to find the solution in a straighforward way. Alternatively, we can find a lower bound that will hold in general as a function of these parameters. We discuss this in more detail next. This lower bound is valid under both FIFO and SIRO policies.

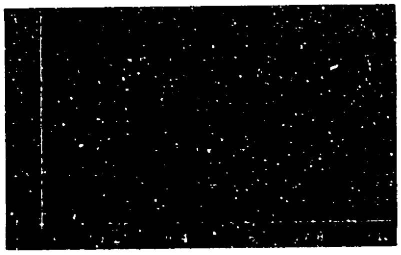

Figure 3-11: $\Pi^j_{SIRO} : f = 1, a = 100, b = 1, j = 5$

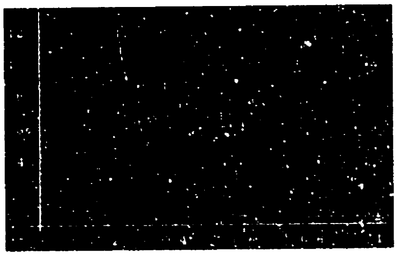

Figure 3-12: $\Pi^j_{SIRO} : f = 1, a = 1, b = 100, j = 5$

Upper Bound

To establish the upper bound on the $E[h_j(T_i^j)]$ under a FIFO policy we draw upon some stochastic order relations between the dispatch policies.

Property 11 •

$$E[T_i^j(SIRO)] = E[T_i^j(FIFO)]$$

•

$$Var[T_i^j(SIRO)] \geq Var[T_i^j(FIFO)]$$

We denote this by

$$T_i^j(SIRO) \geq_v T_i^j(FIFO)$$

See Kleinrock[69].

Property 12 *If X and Y are nonnegative random variables such that $E[X] = E[Y]$, then*

$$X \geq_v Y \iff E[h(X)] \geq E[h(Y)] \textit{ for all convex } h$$

Proof: See Ross[84]

Property 13 (Π_{SIRO}) *is an upper bound on* (Π_{FIFO})

Proof: Follows from above properties

Lower Bound

Property 14 *Jensen's Inequality: Given a random variable X and a convex function g,*

$$E[g(X)] \geq g(E[X])$$

Proof: Standard iterative convexity argument. See Ross[87]

Property 15 *The solution to*

$$\min_{\beta} \sum_{j=1}^{N} f \frac{\beta_j}{2-\beta_j} + a_j(j - \frac{1}{\beta_j})^+ + b_j(\frac{1}{\beta_j} - j)^+ \tag{3.26}$$

s.t.

$$0 \leq \beta_j \leq 1 \quad \forall j$$

is a lower bound on $\Pi_{original}$

Proof: This follows directly by Jensen's Inequality.

Again, we decompose the problem by demand class. The problem for demand class j:

$$\min_{\beta_j} f \frac{\beta_j}{2-\beta_j} + a_j(j - \frac{1}{\beta_j})^+ + b_j(\frac{1}{\beta_j} - j)^+ \qquad \Pi^j(\beta_j)$$

s.t.

$$0 \leq \beta_j \leq 1 \tag{3.27}$$

Because at most one of the terms $(j - \frac{1}{\beta_j})^+$, $(\frac{1}{\beta_j} - j)^+$ will be strictly positive, we consider individually the following two programs:

$$\min_{\beta_j} f\frac{\beta_j}{2-\beta_j} + a_j(j - \frac{1}{\beta_j}) \qquad \Pi_1^j(\beta_j)$$

s.t.

$$\frac{1}{j} \le \beta_j \le 1$$

$$\min_{\beta_j} f\frac{\beta_j}{2-\beta_j} + b_j(\frac{1}{\beta_j} - j) \qquad \Pi_2^j(\beta_j)$$

s.t.

$$0 \le \beta_j \le \frac{1}{j}$$

In $\Pi_1^j(\beta_j)$, both $\frac{\beta_j}{2-\beta_j}$ and $(j - \frac{1}{\beta_j})^+$ are increasing over $[\frac{1}{j}, 1]$. Clearly, then, the function is minimized at $\beta_j = \frac{1}{j}$, with $\Pi_1^j(\frac{1}{j}) = \frac{f}{2j-1}$.

In $\Pi_2^j(\beta_j)$, $\frac{\beta_j}{2-\beta_j}$ is increasing and convex on $[0, \frac{1}{j}]$ while $(\frac{1}{\beta_j} - j)$ is decreasing and convex on $[0, \frac{1}{j}]$ and so, the minimizing β_j is in $[0, \frac{1}{j}]$ $\forall j, f, b$. Since the sum of the terms is convex, solving

$$\frac{d}{d\beta_j}[f\frac{\beta_j}{2-\beta_j} + b(\frac{1}{\beta_j} - j)] = 0 \tag{3.28}$$

yields critical point $\tilde{\beta}_j$.

From this, we can characterize the solution to $\Pi^j(\beta_j)$ by:

$$\Pi^j(\beta_j) = min\{\frac{f}{2j-1}, \Pi_2^j(\tilde{\beta}_j)\} \tag{3.29}$$

where $\frac{f}{2j-1}$ is the solution to $\Pi_1^j(\frac{1}{j})$.

To get some intuition about the relative impact of the parameters, f, a, b in Figures (3-13)-(3-15) we show how the value of $\Pi^j(\beta_j)$ varies. In particular, when either a or b is a dominant term, we find that $\Pi^j(\beta_j)$ is minimized for $\beta_j = \frac{1}{j}$ which in Figure (3-13) and Figure (3-14) is given by $\beta_j = 0.2$. This is not a surprising result because $\beta_j = \frac{1}{j}$ is the value for which the function, bounded below by 0, essentially reaches the value 0. β_j takes on different values only in the presence of significant smoothing costs, which in particular, emphasize larger values of β_j to dampen out the variations.

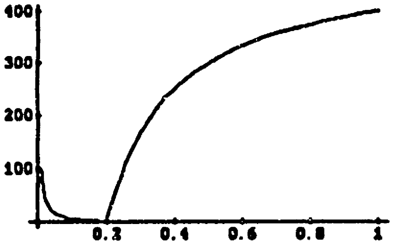

Figure 3-13: $\Pi^j(\beta_j) : f = 1, a = 100, b = 1, j = 5$

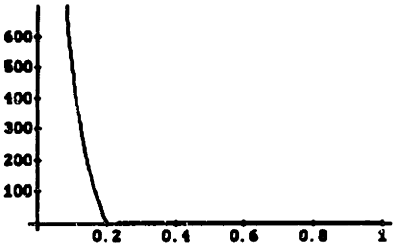

Figure 3-14: $\Pi^j(\beta_j) : f = 1, a = 1, b = 100, j = 5$

In the case that f is large, then $\Pi^j(\beta_j)$ is minimized for $\beta_j \leq \frac{1}{j}$ as in Figure (3-15), approximately the solution to the problem with only smoothing costs.

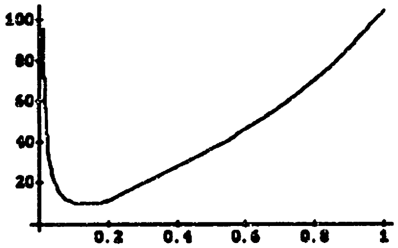

Figure 3-15: $\Pi^j(\beta_j) : f = 100, a = 1, b = 1, j = 5$

3.1.5 Comments

In this chapter we have examined the problem of setting appropriate parameter values for the program ($\Pi_{original}$) when the rigid due-date constraint is relaxed. In particular, we were able to distinguish between the situation in which all work demands immediate delivery and the situation in which there are several demand classes, each with a different leadtime request. Now, although work may be delivered late, there may be a significant penalty associated with it. We did an analysis of ($\Pi_{original}$) under two types of dispatch policies, and also offered a lower bound on the problem that is independent of the dispatch rule.

3.2 Appendix

Liapunov's Condition This is a Central Limit theorem for independent random variables each with a finite mean and $(2+\delta)th$ central moment, for some $\delta > 0$. For $n = 1, 2, \ldots$, let X_n be a random variable with finite mean $E[X_n]$ and finite $(2+\delta)th$ moment $\mu(2+\delta; n) = E[|X_n - E[X_n]|^{2+\delta}]$. Let the sequence $\{X_n\}$ be independent, and let Z_n be defined by

$$Z_n = \frac{S_n - E[S_n]}{\sigma[S_n]} \quad S_n = X_1 + X_2 + \ldots + X_n$$

Then,

$$\lim_{n\to\infty} \Phi_n(u) = e^{\frac{-1}{2u^2}}$$

holds if

$$\lim_{n\to\infty} \frac{1}{\sigma^{2+\delta}[S_n]} \sum_{k=1}^{n} \mu(2+\delta; k) = 0$$

in which $\sigma^2[S_n] = \sum_{k=1}^{n} Var[X_k]$

Chapter 4

Multi-Machine Systems

4.1 Rigid due-dates and multi-machine systems

4.1.1 Problem Description

In this chapter, we consider the due-date constrained production planning problem for multi-machine systems. Our ultimate objective is to indicate how to set production levels for a job shop system. A job shop system is a manufacturing facility which consists of a set of machines, and incoming jobs require processing on some finite subset of these machines, determined according to the characteristics of the individual order. The job shop facility is the most general type of manufacturing environment. In the previous section we have considered one of these machines in isolation, or have assumed that there is exactly one machine in the shop. In what follows we consider the problem for the special case of a serial line: given a set of ordered machines, all incoming demand follows the same fixed processing sequence. Serial lines represent the next higher degree of complexity from the 1-machine system. We then propose to cast the job shop environment as a set of these serial lines, and to apply the analysis of the serial lines to the general system.

We begin by developing a model for a serial production line. The analysis draws upon the results of the one-machine system. Then, as the next extension, we consider a job shop configuration which we interpret as a finite set of these serial lines.

Some of the questions that arise in the context of a larger system:

1. What is the progression of work through each of the machines in the line?
2. How do we ensure due-date feasibility?
3. What is an appropriate objective function?

- To determine how work is satisfied at each station, we need to assume a form for the production policy. Again, we examine the linear class of policies. In terms of the progression of work from machine to machine, we assume that a demand unit requires one unit of processing time at each machine on the line. Upon completion at machine i, it enters the queue of unfinished work at machine $i + 1$. After completion at machine m a job enters a finished goods inventory and waits until its due-date before being picked up by the customer who had placed the order. We discuss this in more detail shortly.
- In scheduling the workload at each machine in this system to maintain due-date feasibility, we have two options: First, we can maintain information about the workload at each machine in terms of the due-date slack, i.e., remaining time until the due-date, for completion at the end of the system. Alternatively, we can divide up the total leadtime into sub-leadtimes for the individual machines.

  To illustrate the distinction between the two perspectives, suppose we have a four machine serial line, and in period t we have a realization of the demand stream D_t^8. In the first case, we trace the course of the D_t^8 quantity at each station for each period until its due-date at the beginning of period $t+8$. In this case we need to keep track of feasibility requirements, i.e, the due-date and the processing requirements at all downstream stations. In particular, we set the parameters of the linear policy to guarantee job completion within the due-date.

In the second model we can choose to set rigid sub-leadtimes at the individual machines so that all work spends at least one unit of time at the machine, and that the overall system due-date is satisfied. With respect to the example, for instance we might have the following: machine 1 has a leadtime of 2 periods, machine 2 has a leadtime of 2 periods, machine 3 has a leadtime of 3 periods, and machine 4 has a leadtime of 1 period. This division is feasible for the original exogenous leadtime. Then we would schedule each machine independently, as we did previously for the one machine case.

We refer to the first approach as a *system* due-date scheme. In the second approach, we view the problem as one of setting *internal* due-dates for each machine. The systemic perspective is more appropriate to a higher-level manager. The second seems more relevant at an operational level since it is easier to give machine due-dates to individual operators.

- In extending the objective function to a larger system, we need to define what smoothness and inventory mean with respect to the larger system. The extension of the objective function for the one machine problem to the larger system is

$$\min \sum_{i=1}^{m} (u_i Var[p_{it}] + v_i E[z_{it}]) \tag{4.1}$$

where u_i and v_i are weight factors for the production variance and inventory costs, respectively, at stage i. p_{it} is the production level of machine i in period t, z_{it} is the inventory level at machine i in period t. We clarify these quantities within the context of each of the models that we will be developing.

We briefly present notation that is applicable for both of the models that we will be studying. We begin by presenting a uniform linear policy that is the logical extension of the optimal one machine policy. This policy will provide a benchmark against which we evaluate the performance of the two models we will be developing.

The remainder of the chapter is in two parts, considering in turn the system due-date scheme and the internal due-date scheme for a serial production line. For each model, we begin with a section of additional notation and assumptions. We present an analysis of each model and discuss the results. Then we directly compare the two models. Finally, we propose how the model can be extended to a more general job shop environment.

4.1.2 Notation and Modelling an m-Station Serial Line

Much of the notation in this section should look familiar, since we have only modified previous notation for the one machine rigid due-date model by adding another subscript to indicate the machine in reference. Here we present notation that is applicable for both models. We defer presentation of model-specific notation.

- m machines in the system, arranged in series
- N Maximum length of a requested due-date window, and without loss of generality, the length of the planning horizon
- D_t^j $j = 1, \dots, N$ is a random variable, denoting the demand in units of work that arrives at the beginning of period t and is due at the beginning of period $t + j$. D_t^j is independent of D_t^k $j \neq k$, and the mean and variance of D_t^j are $\overline{D}^j$, and σ_j^2, respectively. In vector notation we have D_t with elements D_t^j, $j = 1, \dots, N$. We assume $D_t, D_{t+1}, \dots$ are independent, identically distributed random vectors, with mean $\overline{D} = \{\overline{D}^j\}$ and covariance matrix, $Cov(D) = diag\{\sigma_j^2\}$.

Since each job will require exactly one period of processing at each of these stations, we restrict the due dates on incoming demand so that they are feasible. Thus, we require that all demand have a due date of at least m periods, i.e., $D_t^j = 0, j = 1, \ldots, m-1$.

We have not yet identified terms for queue length, inventory, etc.The meaning of these particular quantities differs slightly, and discuss the full definition within the context of the model.

4.2 A Uniform Linear Policy (L)

The structure of the rigid due-date problem has been getting progressively more difficult: We began with only smoothing costs for a 1-machine system, then we incorporated finished goods inventory costs, and now we are looking at larger systems which may have both smoothing and inventory costs associated with each machine. As a result, we have proposed heuristics to generate feasible solutions that we argue are "good" in some way. Typically, given a heuristic for a minimization problem, the traditional way to verify that the heuristic performs well is to find a good lower bound and demonstrate that the heuristic is "close". However, for the rigid due-date problem, it is difficult to relax the constraints, because they are all intertwined, having been recursively defined.

One lower bound that we can offer: Suppose we drop the machine subscripts and assess the performance of the entire system as a "black box". The solution of this is a lower bound on the full multi-machine problem. To accomodate this interpretation, we apply a slight modification of the analysis of the one-machine system. Consider the problem for D_t^k, $k > m$. By simply inputting this into the 1-machine problem, we find that many of the solutions will be infeasible to $(Q2)$, since it is possible for some quantity of work to be processed in fewer than m periods. Instead, we have to assume that D_t^k waits for m - 1 periods outside the system. With this restriction, we are effectively forcing the job to be processed immediately at the first $m - 1$ stations, with full freedom for the remaining $k - m$ days for choosing the period of production. However, this is a very weak bound on the multi-machine problem. We propose the following approach, which, although not as rigorous a comparison, has intuitive appeal, and may prove a more satisfying and realistic resolution.

Instead of worrying about finding a strong lower bound to the problem, suppose we compare the performance of one of our heuristics to the performance of a simple rule that we call a *Uniform Linear Policy:* (L)

In particular, this is defined by specifying that the set of parameters $\{\alpha_i^j\}$ is given by

$$\alpha_i^j = \frac{1}{j-(m-i)} \quad j = m - i + 1 \qquad\qquad (L)$$

Policy (L) corresponds to setting the parameters equal to the inverse of the remainder of the feasible due-date window remaining. For example, consider a specific $\{\alpha_i^j\}$ and the quantity W_{it}^j. The work at machine i at the beginning of period t has $j - (m - i)$ periods remaining at machine i in order to be feasible for completion at the last machine. Policy (L) indicates that we should just produce $\frac{1}{j-(m-i)}$ of the work remaining. This policy is actually a very simple, popular policy to set production levels; in spirit it is a multi-machine version of the 1-machine solution in which we spread work uniformly over the due-date window while maintaining feasible progress through the system. Here we have the additional requirement of taking into account the number of streams remaining to finish processing in the system. This seems to be an appropriate policy against which we can compare our heuristics because it is an extension of the 1-machine case.

4.3 System Due-Dates

4.3.1 Motivation

In modelling a system due-date accounting scheme we are interested in inter-machine effects, and control of the system as a unit. Clearly, there are a large variety of policies to consider. Again, we propose a linear rule as the extension of the policy we studied for the one machine problem. There is some additional basis for this in the literature, as in Graves[86] where it is used in the context of setting production levels for a job shop, albeit without the due-date restriction. For one, it is a reasonably tractable policy with a theoretical basis as described for the one machine case. Another is that it is fairly straightforward to implement.

In this model all of the machines are simply stages within a single production system. The primary information that we record is the workloads ordered by due-date slack. In this case, when we refer to the due-date slack on a job, it indicates the time remaining to finish processing at all m stations. Thus, at each machine, we have a vector of work similar to that of the 1-machine system, however we impose restrictions to maintain system feasibility. The details here follow after the notation.

This section begins with some of the preliminary features of the system due-date model. We then discuss techniques for evaluation of the model.

4.3.2 Modelling the System Due-Date Model

Here we offer some additional notation and assumptions that are specific to our modelling of the system due-date approach.

Notation

- S_{it}^j denotes the arrivals to machine i in period t with a system due-date slack of j periods. S_{it} is the vector of arrivals to machine i in period t, with elements S_{it}^j

- W_{it}^j $j = 1, \ldots, N$, $i = 1, \ldots, m$ is the total quantity of work awaiting machine i at the beginning of period t that is due at the beginning of period $t + j$. W_{it} is the vector of unprocessed or remaining demand at machine i, ordered by due date at the beginning of the t–th period with elements W_{it}^j.

- P_{it}^j $i = 1, \ldots, N$ is the control variable representing the quantity of W_{it}^j that is produced at machine i during period t. P_{it} is the production vector in period t with elements P_{it}^j. The actual quantity processed at machine i in period t is given by $\sum_{j=m}^{N} P_{it}^j$. In shorthand, this is eP_{it}, where e is the N-vector of 1's. For ease of notation, the total production at machine i during period t is given by $p_{it} = eP_{it}$.

- z_{it} is a scalar quantity denoting work that has finished processing at machine i but has not yet been processed at machine $i + 1$. For machines $1, \ldots, m - 1$ this corresponds to work-in-process inventory, still in the "pipeline," given by $z_{it} = \sum_{j=m}^{N} W_{i+1,t}^j$. For the mth station, this quantity is finished goods inventory, and these units remain here until their due-dates at which time they are picked up the customers.

To summarize the super- and sub-scripts for the vector-valued quantities S_{it}, W_{it}, P_{it}: Subscript t denotes the time of observation t, the subscript i denotes machine i, and the superscript j indicates that j periods remain in the due-date slack for the system.

Production Policy and Optimization Program

Given a workload level of $W^j_{i,t}$, the unfinished work at station i at the beginning of period t that has a system due-date slack of length j, the linear production rule corresponds to

$$P^j_{it} = \alpha^j_i W^j_{it}$$

From the definition above, P^j_{it} is the quantity of W^j_{it} that contributes to the cumulative production quantity of machine i during period t, P_{it}. α_i is an $N \times 1$ vector with elements α^j_i. α^j_i represents the fraction of W^j_{it} that will be produced in period t.

Given in vector form

$$P_{it} = A_i W_{it} \tag{4.2}$$

with $A_i = diag(\alpha_i)$. This is the multi-machine extension of the production policy, $P_t = AW_t$ that we studied for the 1-machine system.

As we have indicated, the objective function we are interested in is the minimization of

$$\sum_{i=1}^{m}(u_i Var[p_{it}] + v_i E[z_{it}])$$

Because we have a make-to-order system, we require that $0 \leq \alpha^j_i \leq 1$. Since we consider only demand vectors D_t that are satisfiable, α^j_i is a dummy parameter for $j < m - i + 1$ (we arbitrarily set these $\alpha^j_i = 0$). Now, in order to ensure that the due date constraint is satisfied, and given that it must complete processing at the *ith* station at least $m - i$ periods before its due date, we require that $\alpha^{m-i+1}_i = 1$.

In this way, the optimization program we consider is:

$$\min \sum_{i=1}^{m} (u_i Var[p_{it}] + v_i E[z_{it}]) \tag{4.3}$$

where

$$0 \leq \alpha_i^j \leq 1 \; \forall i = 1, \ldots, m$$

$$\alpha_i^{m-i+1} = 1$$

In this optimization program we will be seeking to set values for the parameters, $\{\alpha_i^j\}$ for each machine i and the elements of each of these vectors.

Dynamics of Input and Output Processes at Each Machine

The input process to machine $i+1$ in period $t+1$ is the production level of machine i in period t. We maintain information about this according to the due-date slack. To express the arrival stream to machine i:

$$S_{i+1,t+1}^j = P_{i,t}^{j+1} \quad i = 1, \ldots, m-1$$

with

$$S_{1,t}^j = D_t^j \quad i = 1, \ldots, m-1$$

Consider a unit of work that is at machine i at the beginning of period t with $j+1$ days in its due-date slack and is processed during this period; this unit becomes a part of the arrival stream to machine $i+1$ at the beginning of period $t+1$, now with j days remaining in its due-date slack. For the first machine, the input is given by the elements of the demand process realization D_t.

To express this in vector form:

$$S_{i+1,t+1} = CP_{it} \tag{4.4}$$

where C is an update matrix, given by the appropriately dimensioned matrix with 1's on the upper diagonal, and 0's elsewhere. Making the substitution of (4.2) into this expression

$$S_{i+1,t+1} = CA_i W_{it} \tag{4.5}$$

where W_{it} is station i's workload vector in period t, $A_i = diag(\alpha_i)$, and C is the update matrix.

The dyramics of the workload queue at station i are given by

$$\begin{aligned} W^j_{i,t+1} &= W^{j+1}_{i,t} - P^{j+1}_{i,t} + S^j_{i,t+1} \\ &= (1 - \alpha^{j+1}_i) W^{j+1}_{i,t} + S^j_{i,t+1} \end{aligned}$$

where the first equation expresses the basic equation of change for the variable $W^j_{i,t}$, and the second equation is the substitution of the production policy into the production level of period t. In particular, each period we have new additions to workload queue i, ordered by the system due-date slack; this is the input stream $S^j_{i,t}$. The work present in period t, with a $j+1$-day leadtime is decremented by the amount of the production $P^j_{i,t}$; the remainder has a j-day leadtime slack at the beginning of period $t+1$.

The vector representation of this is given by:

$$W_{i,t+1} = C\hat{A}_i W_{it} + S_{i,t+1} \tag{4.6}$$

where $\hat{A}_i = I - A_i$.

As we have defined it above, for the first $m-1$ machines, the inventory level at machine i is work that has finished processing at machine i but has not yet been processed at machine $i+1$. The dynamics of the inventory level evolve by:

$$z_{i,t+1} = z_{i,t} + p_{i,t} - p_{i+1,t} \quad i = 1, \ldots, m-1$$

Each period it is updated by additions that correspond to production at machine i, and subtractions as work is produced at machine $i+1$.

As we noted above, $z_{i,t}$ is actually the aggregation of the workload queue of machine $i+1$. In other words,

$$z_{i,t} = \sum_{j=1}^{N} W^{j}_{i+1,t} \quad i = 1, \ldots, m-1 \tag{4.7}$$

We use this relationship later to characterize the dynamics of the inventory level in more detail.

The representation for the inventory at the mth machine, $z_{m,t}$, differs in terms of the work removed each period. In this case, work is removed from finished goods inventory as jobs are delivered to the customer when they reach their respective due-dates. The work that is due at the beginning of period t is $\sum_{j=1}^{N} D^{j}_{t-j}$, given by the accumulation of each stream D^{j}_{t} that arrived j periods previous, i.e., D^{j}_{t-j}.

$$z_{m,t+1} = z_{m,t} + p_{m,t} - \sum_{j=1}^{N} D^{j}_{t-j+1}$$

Here we have limited ourselves to the basic dynamics of the system. We develop these further and use them in analyzing the optimization programs.

Special Cases

For the model that maintains system due-dates, we evaluate the objective function under three cases, in increasing order of complexity, the simpler cases yielding some intuition that may prove useful for the more complex cases.

(Q1) $v_i = 0 \; \forall i, \; u_i = 0 \; i \neq k \; u_k \neq 0$

(Q2) $v_i = 0 \; \forall i, \; u_i \neq 0 \; \exists i$

(Q3) Not all $v_i, \; u_i$ 0, i.e., the general case

In special case $(Q1)$ we are only interested in the variance in production level at machine k. This could represent a situation in which, relative to other machines, we have primary interest in maintaining tight control over the performance of the single machine. This may be the case if machine k is a bottleneck in the line, or its capacity is very close to the mean demand level and the system is sensitive to deviations in the production level at this machine. Under scenario $(Q2)$, we recognize that the production variance at several machines is of issue, to different but significant degrees. Thus, while we may identify one particular machine as being of primary importance, there may be other machines crucial to the overall performance of the system. For this, $(Q2)$ represents the minimization of the weighted sum of the production variance of all of the machines in the line. $(Q3)$ is the most general of the cases, representing the full problem for the serial line, minimization of the weighted sum of production variances and inventory levels.

We note that problems $(Q1)$ and $(Q2)$ are of great interest in their own right, not just as a special, simplified cases of $(Q3)$. This is the case because when we *impose a due-date constraint* then there is likely to be only minimal variation in the inventory level over the class of reasonable feasible policies.

4.3.3 Preliminary Results for the System Due-Date Problem

Before we move on to actually solving each of the optimization programs indicated in the three cases above, we offer some preliminary analysis of the problem that we use to characterize the objective functions $(Q1)$-$(Q3)$.

We recursively use the dynamic representation of $S_{i,t}$ and $W_{i,t}$ given above in (4.5) and (4.6), respectively to express the $W_{i,t}$ in terms of the original demand inputs to the system. In turn, we use this to express the dynamics of the production process, and the inventory levels. We then evaluate the mean and variance of the production process, and the mean of the inventory process for use in the optimization program (4.3).

From the 1-machine problem we have that

$$\begin{aligned} W_t &= C\hat{A}W_{i+1,t} + D_t \\ &= \sum_{h=0}^{\infty}[C\hat{A}]^h D_{t-h} \end{aligned}$$

In this m-machine problem, this corresponds to the characterization of the workload at machine 1:

$$W_{1,t} = \sum_{h_1=0}^{\infty}[C\hat{A}_1]^{h_1} D_{t-h_1}$$

In turn, $W_{1,t}$ is the initial condition upon which we compute the $W_{i,t}$ terms for downstream stations, $i > 1$.

The recursion for $W_{i,t}$ begins as:

$$W_{i,t+1} = \sum_{h_i=0}^{\infty}[C\hat{A}_i]^{h_i} S_{i,t-h_i}$$

$$= \sum_{h_i=0}^{\infty} [C\hat{A}_i]^{h_i} C A_{i-1} W_{i-1,t-h_i-1}$$

$$\vdots$$

with $[C\hat{A}_i]^{h_i} = 0$ for $h_i > N$.

In this expression, the first equation expresses that machine $i+1$ has inputs from machine i, as well as leftover work from previous periods. Additionally, we need to account for the number of periods that work has stayed at a particular machine. The input from machine i, in turn, has a particular relationship to inputs from machine $i-1$, and so on. To summarize: At each substituion that we do on the intial equation of dynamics, we are stepping back one unit of time, and adding in the impact of a machine one machine further back in the serial line. In this way, we generate all of the paths that a job may have taken by the time it exists as an element in the queue $W_{i+1,t+1}$

Continuing in this form, making substitutions, and using the characterization of $W_{1,t}$, we find that

$$W_{i+1,t+1} =$$

$$\sum_{h_{i+1}=0}^{\infty} [C\hat{A}_{i+1}]^{h_{i+1}} C A_i \sum_{h_i=0}^{\infty} [C\hat{A}_i]^{h_i} C A_{i-1} \ldots C A_1 \sum_{h_1=0}^{\infty} [C\hat{A}_1]^{h_1} D_{t-\sum_{j=1}^{i} h_j} \qquad (4.8)$$

In both production and inventory characterizations, we apply the vector α_i to the W_{it} above. In this representation, W_{it} contains all of the information about the paths that the jobs took to get to this position.

In what follows, we use the characterization of the workload dynamic to evaluate several quantities of interest:

Mean production level for station i, $E[p_{it}]$

Variance in production level for station i, $Var[p_{it}]$

Mean inventory level for station i, $E[z_{it}]$

We evaluate these quantities of the m-station serial production system when each of the machines operates under a policy of the form

$$p_{it} = \alpha_i W_{it}$$

We find that $E[p_{it}] = \sum_{j=1}^{N} \overline{D}^j$, which follows intuitively. The expressions for $Var[p_{it}]$ and $E[z_{it}]$ require more effort; in particular, we evaluate each one using recursive functions.

Property 16

$$E[p_{kt}] = \sum_{j=1}^{N} \overline{D}^j \tag{4.9}$$

We pursue an inductive argument here rather than computing directly from the expression above in (4.8).

1. Equation holds for $k = 1$ trivially.
2. Assume it holds for $k = n$.
3. Let $k = n + 1$:

$$\begin{aligned} E[p_{n+1}] = E[p_{n+1,t+1}] &= E[\alpha_{n+1}^T W_{n+1,t+1}] \\ &= \alpha_{n+1}^T E[W_{n+1,t+1}] \\ &= \alpha_{n+1}^T E[C\hat{A}_{n+1} W_{n+1,t} + CA_n W_{n,t}] \\ &\vdots \\ &= \alpha_{n+1}^T E[\sum_{k=0}^{\infty} [C\hat{A}_{n+1}]^k CA_n W_{n,t-k}] \end{aligned}$$

$$
\begin{aligned}
&= \alpha_{n+1}^T \sum_{k=0}^{\infty} [C\hat{A}_{n+1}]^k C A_n E[W_n] \\
&= \alpha_{n+1}^T [I - C\hat{A}_{n+1}]^{-1} C A_n E[W_n] \\
&= \sum_{j=m}^{N} \overline{D}^j
\end{aligned}
$$

To verify the last equality, we note that

- $$\alpha_{n+1}^T [I - C\hat{A}_{n+1}]^{-1} = e$$

  This follows because $[I - C\hat{A}_{n+1}]$ is the matrix with 1's on the diagonal, and $(1 - \alpha_{n+1}^j)$ on the upper diagonal. Inversion and then premulitplication of this quantity by α_{n+1}^T is a vector of 1's (See Chapter 2).

- $$eCA_nE[W_n] = E[p_n] = \sum_{j=m}^{N} \overline{D}^j$$

  corresponds to the inductive assumption □

Next, we take a different approach to establish the computational expressions for the terms $Var[p_{it}]$ and $E[z_{it}]$. The arguments we make are an accounting system for the different paths that a demand stream may take in proceeding through the system.

Suppose we have a serial line of length $\tilde{m}$. When we add a station to the end of the line, the input to the $\tilde{m} + 1$st machine is the smoothed output stream, ordered by system due date, of the $\tilde{m}$th machine. To evaluate the variance in the production policy of the $\tilde{m} + 1$st station under the linear production rule described above, we view the original $\tilde{m}$-machine system as a "black box," with appropriate input and output processes, and the $\tilde{m}+1$st machine as the "2nd" machine in series with the black box system.

Property 17

$$Var[p_{kt}] = \sum_{j=m}^{N} x_1^j(k)\sigma_j^2 \tag{4.10}$$

where

$$x_i^j(k) = (1-\alpha_i^j)^2 x_i^{(j-1)}(k) + (\alpha_i^j)^2 x_{i+1}^{(j-1)}(k) \; i = 1,\ldots,k-1$$

with initial conditions

$$x_i^j(k) = \begin{cases} 0 & j < m-i+1, \; i = 1,\ldots,k-1 \\ 1 & j = m-i+1 \end{cases}$$

and

$$x_k^j(k) = a_k^j$$

given by

$$a_k^j = (1-\alpha_k^j)^2 a_k^{j-1} + (\alpha_k^j)^2 \;\; j < m-k+1$$

$$a_k^{m-k+1} = 1$$

Using the representation for $W_{i+1,t+1}$ in the equation (4.8) to find the $Var[p_{i+1,t+1}]$,

$$Var[\alpha_{i+1}W_{i+1,t+1}] =$$

$$Var[\alpha_{i+1} \sum_{h_{i+1}=0}^{\infty} [C\hat{A}_{i+1}]^{h_{i+1}} CA_i \sum_{h_i=0}^{\infty} [C\hat{A}_i]^{h_i} \ldots CA_1 \sum_{h_1=0}^{\infty} [C\hat{A}_1]^{h_1} D_{t-\sum_{j=1}^{i} h_j}]$$

$$=$$

$$\alpha_{i+1} \sum_{h_{i+1}=0}^{\infty} [C\hat{A}_{i+1}]^{h_{i+1}} CA_i \ldots Cov[D][\alpha_{i+1} \sum_{h_{i+1}=0}^{\infty} [C\hat{A}_{i+1}]^{h_{i+1}} CA_i]^T$$

Clearly (or, rather, not so clearly) this is a difficult expression to evaluate. However, to calculate the actual terms we make use of properties of the parameters α_i^j, and of the structure of the production policy, whereby a fixed fraction is allocated for production from all of the feasible elements of the workload vector. In particular, there are certain terms that drop out of the calculation because they represent infeasible terms; e.g., we can discount work at machine i with due-date slack of less than $m - i$ periods. To actually calculate the value of $Var[p_{i+1,t+1}]$, we account for only the feasible terms and accumulate all of the values. We do this in a recursive way, as we detail next.

In the expression for variance, there is a telescoping dependence of the terms in the calculation. The x_i^j terms are an accounting for the "paths" that a demand stream may have taken. In particular:

$x_i^j(k)$ denotes the contribution to the variance term when a unit of work is at machine i with a due-date slack of length j for calculation of the variance term associated with machine k. The k argument is a marker for which set of initial conditions need to be applied.

$x_1^j(k)$ is used to calculate the variance of the production at machine k. It corresponds to the impact of demand stream D_t^j on machine k, given that it arrives first at machine 1.

To see this, consider

$$x_i^j(k) = (1-\alpha_i^j)^2 x_i^{(j-1)}(k) + (\alpha_i^j)^2 x_{i+1}^{(j-1)}(k) \tag{4.11}$$

Suppose we have a unit of work at machine i with a j-day due-date slack. (α_i^j) of it is processed in the current period, and moves to the next machine at the end of the period. The variance in the production level due to this is $(\alpha_i^j)^2 x_{i+1}^{(j-1)}(k)$. $(1-\alpha_i^j)$ of it remains at this machine and the due-date slack decreases by one unit in the next period, yielding the $(1-\alpha_i^j)^2 x_i^{(j-1)}(k)$ term.

We show that the expression for $Var[p_{mt}]$ accounts for all of the paths. Similar arguments hold for all of the other $Var[p_{it}]$ terms. Each term follows by generating the terms given in the representation of W_{it} in the equation (4.8). First, we have that $\sigma_j^2 = 0$ for $j < m$, and thus, $x_1^j(m)\sigma_j^2 = 0, j < m$, since only feasible demands are included. Consider the term $x_1^m(m)$. In our model, D_t^m, i.e., demand entering in period t and due at the beginning of period $t+m$, must be processed immediately by each machine in the line. The relevant release coefficient at each station i is $\alpha_i^{m-i+1} = 1$. Thus the contribution to the variance in production at station m in period $t+m-1$ due to D_t^m is $\prod_{i=1}^m (\alpha_i^{m-i+1})^2\sigma_m^2 = \sigma_m^2$.

Demand D_t^{m+1} has the flexibility to be delayed (i.e., not processed immediately) at at most one station. In period $(t+m)$, D_{t-1}^{m+1} and D_t^{m+1} will both contribute to p_{t+m}, and thus the parameters associated with the routings of these demands will contribute to the x_1^{m+1} term. A portion of D_t^{m+1} is processed immediately at each of the m stations, equal to $\prod_{i=1}^m \alpha_i^{m-i+2} D_t^{m+1}$, and thus requiring a $\prod_{i=1}^m (\alpha_i^{m-i+2})^2 \sigma_{m+1}^2$ term in the variance. For the portion of D_{t-1}^{m+1} that is delayed, there are m choices for the site of the delay. The quantity delayed at station i is $\{1-\alpha_i^{m-i+1}) \prod_{j=1}^{i-1} \alpha_j^{m-j+2}\} D_{t-1}^{m+1}$, with a contribution of $\{(1-\alpha_i^{m-i+1})^2 \prod_{j=1}^{i-1} (\alpha_j^{m-j+2})^2\}\sigma_{m+1}^2$ to the variance.

Calculation of the $x_1^{m+1}(m)$ accounts for all of these terms. For the case in which

$m = 3$, we look at the contribution by D_t^4:

$$x_1^4 = (1 - \alpha_1^4)^2 + (\alpha_1^4)^2(1 - \alpha_2^3)^2 + (\alpha_1^4)^2(\alpha_2^3)^2(1 - \alpha_3^2)^2 + (\alpha_1^4)^2(\alpha_2^3)^2(\alpha_3^2)^2$$

The first term accounts for one period of delay at machine 1. The second term indicates immediate processing at machine 1, and delay at machine 2. The third term has immediate processing at machine 1 and 2, with delay at machine 3. The last term has immediate processing at all of the machines for a fraction of D_t^4 equal to $(\alpha_1^4)(\alpha_2^3)(\alpha_3^2)D_t^4$ and a contribution to the variance of $(\alpha_1^4)^2(\alpha_2^3)^2(\alpha_3^2)^2\sigma_4^2$. Rearranging the terms, we get

$$x_1^4 = (1 - \alpha_1^4)^2 + (\alpha_1^4)^2\{(1 - \alpha_2^3)^2 + (\alpha_2^3)^2[(1 - \alpha_3^2)^2 + (\alpha_3^2)^2]\}$$

which corresponds to the statement above. The argument continues in this fashion, accounting for all the "paths". □

The significance of this representation is that for specific parameter values we can systematically evaluate the variance term.

Property 18

$$E[z_{kt}] = \sum_{j=m}^{N} y_1^j(k)\overline{D}^j \qquad (4.12)$$

where

$$y_i^j(k) = \alpha_i^j y_{i+1}^{j-1}(k) + (1 - \alpha_i^j)y_i^{j-1}(k)$$

with initializations:

- $$y_k^j(k) = \alpha_k^j T_k^j + (1 - \alpha_k^j)y_k^{j-1}(k) \quad k = 1, \ldots, m - 1$$

where

$$T_k^j = 1 + (1 - \alpha_k^{j-1})T_k^{j-1}$$

and

$$T_k^{m-k+1} = 1$$

- $$y_k^{m-k+1}(k) = \begin{cases} 0 & k = m \\ 1 & k < m \end{cases}$$

- $$y_m^j(m) = (j-1)\alpha_k^j + (1 - \alpha_k^j)y_m^{j-1}(m) \quad k = 1, \ldots, m-1$$

with

$$y_m^1(m) = 0$$

Here, again, we use the representation of $W_{i,t}$ given in (4.8) to express $E[z_{it}]$. We have that $E[z_{it}] = E[\sum_{j=1}^{N} W_{i+1,t}^j]$ $i = 1, \ldots, m-1$. Applying the expectation operator,

$$\begin{aligned} E[W_{i+1,t+1}] \quad = & \\ & E[\sum_{h_{i+1}=0}^{\infty} [C\hat{A}_{i+1}]^{h_{i+1}} CA_i \ldots CA_1 \sum_{h_1=0}^{\infty} [C\hat{A}_1]^{h_1} D_{t-\sum_{j=1}^{i} h_j}] \end{aligned}$$

Again, we know that this expression is difficult to evaluate. Rather, we use the recursive form of this expression in order to actually calculate the value of this term.

The recursive argument for the correctness of the $E[z_{it}]$ term is similar to that given for the variance expression. We account for all of the demand streams and the quantity of work, ordered by due-date slack that comes from an individual demand stream. Consider the dynamics for $y_i^j(k)$:

$$y_i^j(k) = y_i^{j-1}(k)(1-\alpha_i^j) + \alpha_i^j y_{i+1}^{j-1}(k)$$

$\alpha_i^j W_{it}^j$ is produced at machine i in period t. This quantity then enters $z_{i,t+1}$. This corresponds to the $\alpha_i^j y_{i+1}^{j-1}(k)$ term. $(1-\alpha_i^j)$ of it remains at this machine and the due-date slack decreases by one unit in the next period, yielding the $(1-\alpha_i^j)^2 y_i^{(j-1)}$ term. With respect to the calculation of $y_k^j(k)$, $(1-\alpha_k^j)W_{kt}^j$, $j > m-k+1$ is left behind in the queue, and awaits another period for the decision of production at machine k. What about the quantity $\alpha_k^j W_{kt}^j$ that is produced? How long does it remain in the inventory of machine k? It stays as long as it is not produced at machine $k+1$. This quantity is captured in T_k^j. In a sense it represents the expected time that work entering z_{kt} with a j-day leadtime remains in this inventory. Thus, it has the terms corresponding to the linear parameters of the machine immediately succeeding it.

In this recursive way, we can account for all of the paths by which a demand stream arrives to machine k. Without going through the enumerative argument to prove the form of $E[z_{it}]$, we argue that we account for all of the demand streams: Work enters W_{kt} with the length of the due-date slack varying from $m-k+1$ to $N-k$, and then appropriate fractions of these enter the inventory for $(j-1)$ periods, where j is the due-date slack. □

Recursive form for calculating $x_i^j(k)$ and $y_i^j(k)$

We summarize how to generate the $x_i^j(k)$ and $y_i^j(k)$ values as follows:

$$\begin{aligned} &\text{Given } \{\alpha_i^j\} && \forall i,j \\ &\quad\text{with } \alpha_i^{m-i+1} && = 1 \\ &&& \alpha_i^j = 0 \quad j < m-i+1 \end{aligned}$$

Generate $x_i^j(k)$:

```
Initialize
        $x_k^{m-k+1}(k) = 1$
        $x_k^j(k) = a_k^j$
    where
        $a_k^j = (1 - \alpha_k^j)^2 a_k^{j-1} + (\alpha_k^j)^2$
        $a_k^{m-k+1} = 1$
Recursion
        $i = k$
    begin
        $i = i - 1$
    begin
        $j = 2, 3, \ldots, N - (m - i)$
        $x_i^j(k) = \frac{x_i^{j-1}(k) x_{i+1}^{j-1}(k)}{x_i^{j-1}(k) + x_{i+1}^{j-1}(k)}$
    end
    end
```

Generate $y_i^j(k)$: with initializations:

- $$y_m^j(m) = (j-1)\alpha_k^j + (1-\alpha_k^j)y_m^{j-1}(m) \quad k = 1, \ldots, m-1$$

  with

  $$y_m^1(m) = 0$$

- $$y_k^{m-k+1}(k) = \begin{cases} 0 & k = m \\ 1 & k < m \end{cases}$$

- $$T_k^j = 1 + (1-\alpha_{k+1}^{j-1})T_k^{j-1}$$

  and

  $$T_k^{m-k+1} = 1$$

  $$y_k^j(k) = \alpha_k^j T_k^j + (1-\alpha_k^j)y_k^{j-1}(k) \quad k = 1, \ldots, m-1$$

Recursion

```
    $i = k$
  begin
    $i = i - 1$
```

```
begin
    $j = 2, 3, \ldots, N - (m - i)$
    $y_i^j(k) = y_i^{j-1}(k)(1 - \alpha_i^j) + \alpha_i^j y_{i+1}^{j-1}(k)$
end
end
```

4.3.4 Analysis of System Due-Date Problems $(Q1)$-$(Q3)$

In each of the cases that we will be discussing for the system due-date model, we offer a heuristic that takes into account some local information about the cost function and uses as a subroutine results from the 1-machine model. The solutions that we get via the heuristic we compare to the solution offered by policy (L).

(Q1) System Due-Dates: $v_i = 0 \; \forall i, \;\; u_i = 0 \; i \neq k, \;\; u_k \neq 0$

The optimization program for this case is given as:

$$\min u_k Var[p_{kt}] \qquad (Q1)$$

where

$$0 \leq \alpha_i^j \leq 1 \; \forall i = 1, \ldots, m$$

$$\alpha_i^{m-i+1} = 1$$

The expression for the variance of the kth machine is given above, which we repeat here:

$$Var[p_{kt}] \;=\; \sum_{j=m}^{N} x_1^j(k)\sigma_j^2$$

where

$$x_i^j(k) \;=\; (1-\alpha_i^j)^2 x_i^{(j-1)}(k) + (\alpha_i^j)^2 x_{i+1}^{(j-1)}(k) \; i = 1, \ldots, k-1$$

with initial conditions as detailed above.

$$x_i^j(k) = \begin{cases} 0 & j < m - i + 1, \; i = 1, \ldots, k-1 \\ 1 & j = m - i + 1 \end{cases}$$

and

$$x_k^j(k) = a_k^j$$

given by

$$a_k^j \;=\; (1 - \alpha_k^j)^2 a_k^{j-1} + (\alpha_k^j)^2 \;\; j < m - k + 1$$

$$a_k^{m-k+1} \;=\; 1$$

Consider the recursion function

$$x_i^j(k) \;=\; (1 - \alpha_i^j)^2 x_i^{(j-1)}(k) + (\alpha_i^j)^2 x_{i+1}^{(j-1)}(k) \; i = 1, \ldots, k-1$$

Given any set of $\{\alpha_i^j\}$, we have that

- $x_i^j(k)$ is monotonic in $x_i^{j-1}(k), x_i^{j-2}(k), \ldots$
- $x_i^j(k)$ is monotonic in $x_{i+1}^{j-1}(k), x_{i+1}^{j-1}(k), \ldots$

Thus, to minimize

$$\sum_{j=m}^{N} x_1^j(k) \sigma_j^2,$$

We begin by minimizing ("from left-to-right ")

$$\{x_k^{m-k+1}(k), x_k^{m-k+2}(k), \ldots, x_k^N(k)\}$$

then

$$\{x_{k-1}^{m-(k-1)+1}(k), x_{k-1}^{m-(k-1)+2}(k), \ldots, x_{k-1}^{N}(k)\}$$

$$\vdots$$

and finally

$$\{x_1^m(k), x_1^{m+1}(k), \ldots, x_1^N(k)\}$$

How then, do we minimize a particular $x_i^j(k)$ term? Suppose we have the set of optimal $\{x_r^s(k)\}, r = k, k-1\ldots, i$ and $s = m - r + 1, \ldots, N$ and additionally, we have $\{x_i^s(k)\}$ $s = m - i + 1, m - i + 2, \ldots, j - 1$. To find the optimal $x_i^j(k)$ we need to find the optimal α_i^j for the recursion

$$x_i^j(k) = (1 - \alpha_i^j)^2 x_i^{(j-1)}(k) + (\alpha_i^j)^2 x_{i+1}^{(j-1)}(k) \tag{4.13}$$

Note that for the fixed $x_i^{(j-1)}(k)$ and $x_{i+1}^{(j-1)}(k)$ that we are given, $x_i^j(k)$ is convex in (α_i^j). Differentiating (4.13) and solving for (α_i^j) yields

$$\alpha_i^{j*} = \frac{x_i^{j-1}(k)}{x_i^{j-1}(k) + x_{i+1}^{j-1}(k)}$$

and making the substitution to (4.13)

$$x_i^{j*}(k) = \frac{x_i^{j-1}(k) x_{i+1}^{j-1}(k)}{x_i^{j-1}(k) + x_{i+1}^{j-1}(k)} \forall j = k, \ldots N - 1$$

In the initialization portion, the parameters for the kth machine are set according to the same recursion that we used for the 1-machine model. And so it follows that under objective function $(Q1)$, when the objective function is $\min u_k Var[p_{kt}]$, the optimal policy at the kth machine itself is the same as the optimal policy for the one machine system. What about the value of the upstream parameters? $\{\alpha_i^{j*}\}$ is decreasing but $\alpha_i^{j*} \geq \frac{1}{j-(m-i)}$, $j = m - i + 1, \ldots, N$ where the $\alpha_i^{j*} = \frac{1}{j-(m-i)}$ is the policy under (L) and corresponds to the previous one machine optimal control. Thus, more work is processed early at upstream machines, giving longer due date windows over which to smooth production at machine k. In this way, although for machine k there is positive benefit to having a reasonably smooth output stream at an upstream station, the crucial thing is to do so while moving the workload through the series of machines $1, 2, \ldots, k-1$ to maximize the length of the due date windows which are input to the kth station.

Again, we find that the optimal parameter values are independent of both the σ_j^2 and the parameter u_k. Given the one machine result this is not surprising. Essentially, we want the production level to be as uniform as possible. This holds regardless of the original input stream. This follows because of the monotonicity properties of the recursion, as in the 1-machine case.

Solution of $(Q1)$:

To summarize how to generate the $x_1^j(k)$ terms that we require:

1. Initialize according to

$$x_k^{m-k+1}(k) = 1$$

$$x_k^j(k) = a_k^j$$

where

$$a_k^j = \frac{1}{j-(m-k)}$$

2. *Recursion*

$$x_i^j(k) = \frac{x_i^{j-1}(k)x_{i+1}^{j-1}(k)}{x_i^{j-1}(k) + x_{i+1}^{j-1}(k)}$$

Example:(Q1)

We consider a specific instance of the problem $(Q1)$ that we have identified above. In this example, we assume that

- The number of stations $m = 4$
- $N = 8$, with arrival processes given by $D_t^4, D_t^5, D_t^6, D_t^7, D_t^8$
- The weight factor $u_k = 1$ for each of the machines, $k = 1, \ldots, m$

We evaluate each of the functions

$$x_i^j(k)$$

for each demand stream $\{D_i^j\}$ $j = 4, \ldots, 8$ and for each of the machines in the line $k = 1, \ldots, 4$. The $\{\alpha_i^j\}$ values that we use are generated by the specification of the optimal $\{\alpha_i^j\}$ values that we derived via solution of $(Q1)$.

In figures (4-1)-(4-4) we compare the solutions to $(Q1)$ above and the solution that the policy (L) affords. Note that the $x_1^j(1)$ values are identical; policy (L) is the optimal 1-machine solution, and so the parameters under this objective function are the same. As k gets larger, we find that the solution given for $(Q1)$ does successively better than policy (L). We aggregate these results in Table (4.2).

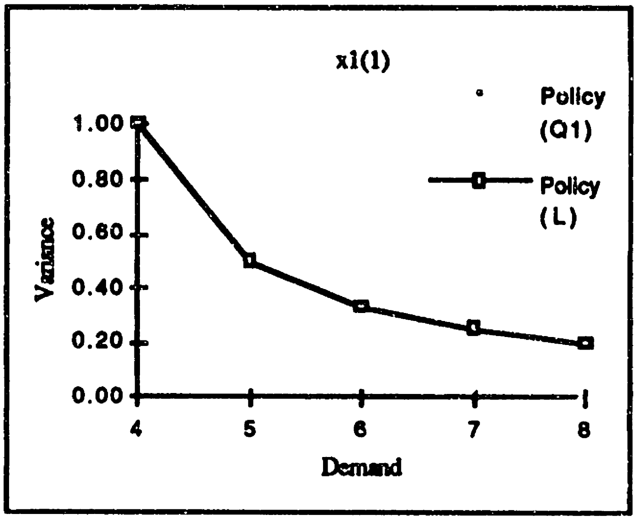

Figure 4-1: $x_1^j(1)$: Policies $(Q1)$ and (L) with $m = 4$, $N = 8$

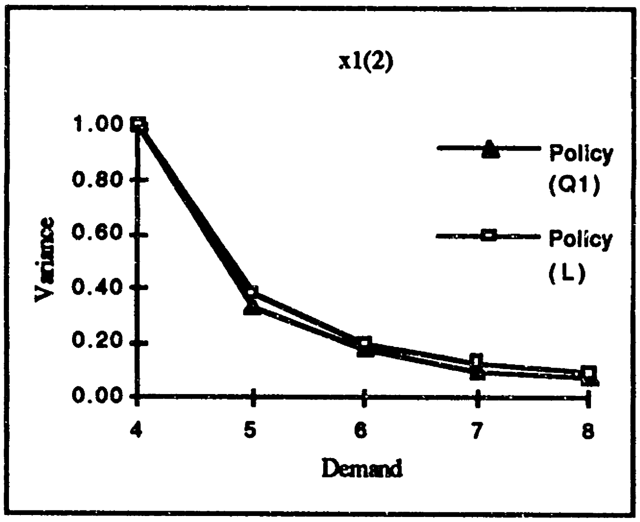

Figure 4-2: $x_1^j(2)$: Policies $(Q1)$ and (L) with $m = 4$, $N = 8$

k	$Q1(k)$	L	$\%\Delta$
1	2.28	2.28	–
2	1.67	1.80	7
3	1.43	1.70	16
4	1.31	1.68	22

Table 4.1: $x_1(k)$: % improvement of $Q1(k)$ vs L

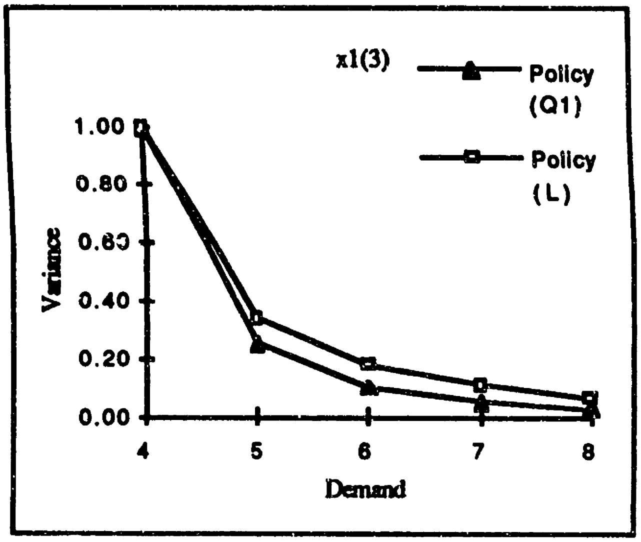

Figure 4-3: $x_1^j(3)$: Policies $(Q1)$ and (L) with $m = 4$, $N = 8$

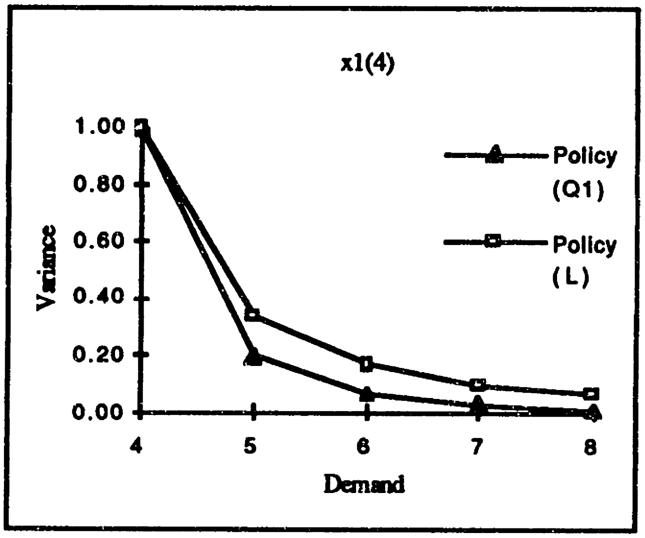

Figure 4-4: $x_1^j(4)$: Policies $(Q1)$ and (L) with $m = 4$, $N = 8$

(Q2) System Due-Dates: $v_i = 0\ \forall i,\ u_i \neq 0\ \exists i$

The program $(Q2)$ in this case minimizes the weighted sum of the production variances of the m stations in the line.

$$\min \sum_{i=1}^{m} u_i Var[p_{it}] \qquad\qquad (Q2)$$

$$0 \leq \alpha_i^j \leq 1\ \forall i = 1, \ldots, m$$

$$\alpha_i^{m-i+1} = 1$$

where the variance for machine k is given by the recursive function above. Although each of the $Var[p_{it}]$ terms is convex in an individual $[\alpha_i^j]\ \forall j$, $Var[p_{it}]$ is not convex in the other $\{\alpha_i\}$. This makes the problem very difficult, with no special structure to exploit as we did in the 1-machine case and in (Q1). In this section we present a heuristic for finding a good solution to (Q2). We will use some local information about the function to generate a good feasible solution.

From (Q1), given an objective function $\min u_k Var[p_{kt}]$, we can recursively compute the optimal parameters for upstream machines, $\{\alpha_i^j\}_{i=1}^{k}$ with $j = m-i+1, \ldots, N$. The downstream values can be set arbitrarily as long as the solution is feasible; for example, we can have $\alpha_i^j = 1\ j = m - i + 1, \ldots, N$ with $i = k + 1, \ldots, N$. We use this result to suggest a heuristic to set parameter values for $(Q2)$.

The heuristic we suggest here uses the result of (Q1) to do some local optimization. Consider a small version of the problem. Suppose we are interested in setting the parameters at two machines k and l, so that we have

$$\min u_k Var[p_{kt}] + u_l Var[p_{lt}] \tag{4.14}$$

where $k \leq l$, with $u_l \geq u_k$.

Heuristic Q2:(2-machine optimization in m-station line)

1. Solve $\min u_l Var[p_{lt}]$. Apply the recursive function to set $\alpha_l, \alpha_{l-1}, \ldots, \alpha_1$. For $\alpha_i, i > l$ set $\alpha_i^j = 1, j = m - i + 1, \ldots, N$. Denote this solution by $\hat{\alpha}$.
2. Solve $\min u_k Var[p_{kt}]$ and set vectors $\alpha_k, \alpha_{k-1}, \ldots, \alpha_1$. From the solution to $\min u_l Var[p_{lt}]$ above, use the recursive function to set vectors upstream from machine k between machines $k+1$ to machine l, to get $\alpha_l, \alpha_{l-1}, \ldots, \alpha_{k+1}$. Denote these values by $\tilde{\alpha} = \{\tilde{\alpha}_i\}$ $i = 1, \ldots, k$.
3. Now, let

$$\alpha^* = arg \min_{\tilde{\alpha}, \hat{\alpha}} u_k Var[p_{kt}] + u_l Var[p_{lt}] \tag{4.15}$$

The decision about which of these vectors to choose is a function of the weight parameters.

Heuristic Q2:(m-machine case) We indicate the generalization of the 2-machine heuristic of $(Q2)$ to a k-machine heuristic. Essentially, we begin by ranking all of the machines and then undertake a pairwise comparison of the machines, using the 2-machine criterion to update a current "best" vector of α parameters.

1. Rank order the u_i terms.
2. Solve $\min u_k Var[p_{it}]$. Apply the recursive function to set $\alpha_i, \alpha_{i-1}, \ldots, \alpha_1$, and set the values of the upstream parameters to $\alpha_k^j = 1$. Denote these values by $\hat{\alpha} = \{\hat{\alpha}_i\}$ $i = 1, \ldots, k$

3. Begin with the machines with the two largest weights, u_k and u_l. Update the vectors according to $Q2(2 - machine)$. Output here is a vector. Take this vector in conjunction with the vector with the next greatest weight and repeat heuristic $Q2(2 - machine)$ for these two.

Examples: Q2 In what follows we consider in turn the following policies

1. $Q2A(2 - machine)$
2. $Q2B(2 - machine)$
3. Linear rule (L)

We use the 4-machine serial line that we described earlier. Following Heuristic $Q2(2 -- machine)$, we solve

$$\min u_3 Var[p_{3t}]$$

From this we get

$$\begin{aligned}
\tilde{\alpha}_3 &= \{0.00, 1.00, 0.50, 0.33, 0.25, 0.20, 0.17, 0.14\} \\
\tilde{\alpha}_2 &= \{0.00, 0.00, 1.00, 0.67, 0.50, 0.40, 0.33, 0.29\} \\
\tilde{\alpha}_1 &= \{0.00, 0.00, 0.00, 1.00, 0.75, 0.60, 0.50, 0.43\}
\end{aligned}$$

To complete the solution so that it is feasible, we set

$$\tilde{\alpha}_4 = \{1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00\}$$

Denote this solution by $\tilde{\alpha}$ where

$$\tilde{\alpha} = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4\}$$

	$Q(2)A-(2machine)$		$Q(2)B-(2machine)$		(L)	
j	$x_1^j(1)$	$x_1^j(3)$	$x_1^j(1)$	$x_1^j(3)$	$x_1^j(1)$	$x_1^j(3)$
4	1.00	1.00	1.00	1.00	1.00	1.00
5	0.63	0.25	0.50	0.33	0.50	0.38
6	0.46	0.10	0.33	0.17	0.33	0.20
7	0.37	0.05	0.25	0.10	0.25	0.13
8	0.30	0.03	0.20	0.07	0.20	0.09

Table 4.2: $Var(p_t)$ at machines 1 and 3 under different policies

We use these values of the parameters $\tilde{\alpha}$ to generate the $x_1^j(k)$ values; we summarize these values in Table (4.2).

We use these values of $\tilde{\alpha}$ to then evaluate the function

$$\sum_{j=1}^{N} x_1^j(3)(\tilde{\alpha}) = 1.43$$

and

$$\sum_{j=1}^{N} x_1^j(1)(\tilde{\alpha}) = 2.75$$

And so the value of the objective function becomes

$$u_1 2.75 + u_3 1.43$$

We compare this to the value of the solution given for case B.

In solution (B), we use the terms $\{\tilde{\alpha}_2, \tilde{\alpha}_3, \tilde{\alpha}_4\}$ since $\hat{\alpha}_i = \tilde{\alpha}_i$, for $i = 2, 3, 4$. Only the value for the parameters at machine 1 change, since we re-solve to minimize $Var[p_{1t}]$. Thus,

$$\hat{\alpha}_1 = \{0.00, 0.00, 0.00, 1.00, 0.50, 0.33, 0.25, 0.20\}$$

(u_1, u_3)	$(Q2)$
(4,1)	10.79
(1,4)	8.47

Table 4.3: Evaluation of $(Q2)$

From this we find that

$$\sum_{j=1}^{N} x_1^j(3)(\hat{\alpha}) = 1.67$$

and

$$\sum_{j=1}^{N} x_1^j(1)(\hat{\alpha}) = 2.28$$

And so the value of the objective function becomes

$$u_1 2.28 + u_3 1.67$$

The choice of whether to select $\tilde{\alpha}$ or $\hat{\alpha}$ is then a function of the relative values of u_1 and u_3. Thus, choose $\tilde{\alpha}$ when

$$u_1 2.75 + u_3 1.43 < u_1 2.28 + u_3 1.67$$

else choose $\hat{\alpha}$. Equivalently, choose $\tilde{\alpha}$ when

$$u_1 < .51 u_3$$

Example:$(Q2)$

In Table (4.3) we make assume values for (u_1, u_3) and evaluate the corresponding values of $(Q2)$:

	machine			
	1	2	3	4
$x_1(i)$	2.28	1.80	1.70	1.68
$y_1(i)$	7.50	6.25	5.62	0.62

Table 4.4: Variance and inventory terms for machine i

($Q3$) System Due-Dates: v_i u_i not all 0

The objective function we consider now is:

$$\min u_i Var[p_{it}] + v_i E[z_{it}] \qquad (Q3)$$

This is the most general form of the problem that we have been considering. The complexity of ($Q2$) was significantly greater than that of ($Q1$), but we were still able to generate a reasonable heuristic. The inclusion of the inventory terms greatly increases the difficulty of the type of optimization we are attempting. The type of local optimization that we did for the 1-machine problem does not work here because of the form of the functions. However, because we have a computationally straightforward way in which to evaluate the function, we can make some observations, "rules of thumb", and verify them numerically.

There is ample opportunity for computational testing here. In particular, consider the uniform linear (L) that we have repeatedly used as the benchmark against which to evaluate other policies. It is a good starting point for this problem: For one, it sets production levels to be relatively smooth, although we have seen that these values are not optimal in the sense of either ($Q1$) or ($Q2$). The benefit of the policy not being optimally smooth is that the inventory levels may not have reached the same levels.

From Table (4.4) we see how the process becomes progressively smoother and the quantity of inventory held decreases.

The type of testing that we envision here is with respect to the values of the

	Ex. 1	Ex. 2	Ex. 3	Ex. 4
u_1	1	4	4	1
v_1	4	1	4	1
u_3	1	4	1	4
v_3	4	1	1	4

Table 4.5: Matrix of test examples: Specify u_i, v_i values

parameters. In particular, we consider situations in which we might be interested in approximating the program $(Q3)$ by either $(Q1)$ or $(Q2)$ in the case that the smoothing terms dominate, or offer the lower bound in the case that the inventory terms dominate.

Example:$(Q3)$ The optimization we are looking for is:

$$\min u_1 Var[p_{1,t}] + v_1 E[z_{1,t}] + u_3 Var[p_{3,t}] + v_3 E[z_{3,t}]$$

However, we have difficulty solving this, so we evaluate the objective function at certain values of the paramters. In particular, we evaluate the objective function at the solutions given by the problems $(Q2)$ and policy (L). The function we are looking at is:

$$u_1 x_1(1) + v_1 y_1(1) + u_3 x_1(3) + v_3 y_1(3) \tag{4.16}$$

In Table (4.5) we make assumptions on values for u_1, v_1, u_3, v_3 to create four test examples. Our idea here in making assumptions on the values of the parameters u_1, v_1, u_3, v_3 is to indicate the relative importance of the parameters. In Example 1, the inventory terms dominate, with $u_1 = u_3 = 1, v_1 = v_3 = 4$. The situation is reversed in Example 2. Examples 3 and 4 indicate which of the machines dominate.

In Table (4.6) we evaluate the function given in equation (4.16), under the different policies available. In particular, evaluating the function at the values indicated by Heuristic (Q2) perform better than the linear policy. We return to this example again

	Ex. 1	Ex. 2	Ex. 3	Ex. 4
$\alpha(L)$	56.46	29.04	46.44	39.06
$\alpha(Q2A)$	50.18	28.22	46.43	31.97
$\alpha(Q2B)$	46.59	26.46	41.43	31.62

Table 4.6: Evaluate ($Q3$) at values for u_i, v_i

in the context of the internal due-date model.

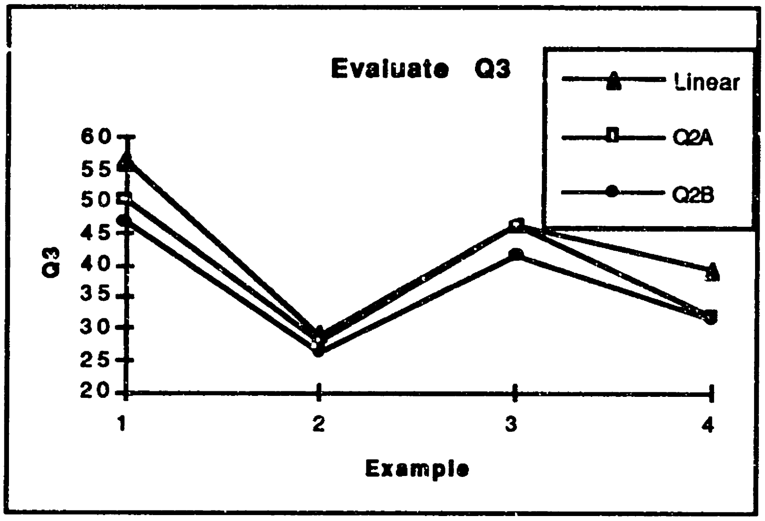

Figure 4-5: $Q3$ example evaluated under policies $Q2A, Q2B, L$

4.4 Internal Due-Dates

In the case of an internal due-date accounting scheme our primary interest is not in the inter-machine effects of a production policy, as in the first model. Rather, we recognize that there may be difficulties in implementing such a policy, especially for larger systems. In response, we consider a version of the problem in which the machines are de-coupled to function independently. They are linked by the feasibility of the due-date window assignment made by the manager, but the operator of each machine is ignorant of the constraints facing other machines in the line.

We de-couple the system via the individual machine due-dates so that the problem decomposes into m 1-machine problems, each of which we can analyze in a 1-machine framework. In particular, we do this by setting feasible internal leadtimes at each machine for each input stream. Here feasibility means that a realization of demand stream D_t^k spends at least one period at each machine and completes processing at all of the machines within the k period leadtime. We require that a given demand stream finish processing at the particular machine within the assigned internal leadtime. After completing processing at machine i, work waits until its internal due-date (i.e, is held within an intermediate "finished goods inventory") and then proceeds to machine $i+1$.

This section begins with some of the preliminary features of the internal due-date model. We then propose techniques for evaluation of the model.

4.4.1 Modelling the Internal Due-Date Model

We begin our discussion of this model with some additional notation that we use in describing the internal due-date model. We then offer some assumptions and preliminary results. Then we proceed to present the model in more detail.

Notation

- $d_i(D_t^k)$ is a decision variable that is the length of the internal due-date leadtime assignment at machine i for demand stream D_t^k. For example, suppose that $m = 2$, and for the demand stream D_t^6, the assignment is given by $\{d_1(D_t^6) = 4, d_2(D_t^6) = 2\}$. By this, we require that the demand stream D_t^6 must complete processing at machine 1 by the beginning of period $t + 4$, at which time it becomes an input to the queue at machine 2. Between the time that work completes processing at machine 1 and becomes input to machine 2 it is held in the "finished goods inventory" of machine 1. Similarly, D_t^6 is processed at machine 2 sometime within the interval $[t + 4, t + 5]$. It is ready for delivery to the customer at the beginning of period $t + 6$. In general, D_t^k must have completed processing at machine i at the beginning of period $t + \sum_{l=1}^{i} d_l(D_t^k)$ for each i.

- $Q_{it}^j(k)$ $j = 1, \ldots, N$, $i = 1, \ldots, m$ denotes the queue length of unfinished work at machine i in period t of demand stream D_t^k with j days remaining in its internal due-date slack. The additional subscript k distinguishes it from the familiar W_{it}^j. Previously, we were able to aggregate all of the demand streams and order the queue by the remaining slack for the system. Here we need to make the distinction for individual demand streams because we may assign different internal leadtimes to the different machines. $Q_{it}(k)$ is the vector representation of unprocessed or remaining demand at machine i, ordered by due date at the beginning of the t–th period with elements $Q_{it}^j(k)$.

- $P_{it}^j(k)$ $i = 1, \ldots, N$ is the control variable denoting the quantity of $Q_{it}^j(k)$ that is produced during period t. $P_{it}(k)$ is the production vector in period t with elements $P_{it}^j(k)$. The quantity processed at machine i in period t of demand class k is given by $\sum_{j=1}^{N} P_{it}^j(k)$. For ease of notation, the total production at machine i during period t of demand stream k is given by $p_{it}(k) = eP_{it}(k)$, where e is the N-vector of 1's. The total overall production at machine i during period t is $\sum_{k=1}^{N} p_{it}(k)$.

- z_{it} is a scalar quantity denoting finished goods inventory at machine i from demand stream k at the beginning of period t that has not yet been "delivered" to machine $i+1$. Note that this is a quantity that we keep note of at machine i, whereas in the system due-date model, the inventory level comprises all work that is held downstream from machine i, either in queue at another machine or in the finished goods inventory after machine m.

- $S_{it}(k)$ is the arrival of work to machine i in period t of elements from the demand stream D_t^k. S_{it} is the accumulation of all of these inputs, given by $S_{it} = \sum_{k=1}^{N} S_{it}(k)$.

Production Policy and Optimization Program

Because we maintain a queue for each demand stream at each machine, we also include the additional demand stream identifier within the production policy. The production policy at each station is of the form:

$$p_{it}(k) = \alpha_i Q_{it}(k)$$

The crucial problem for this scheme is how to determine the $d_i(D_t^j)$ values. This issue does not affect the production policy.

As in the system model the objective function is:

$$\min \sum_{i=1}^{m} (u_i Var[p_{it}] + v_i E[z_{it}])$$

The constraints here are slightly different than in the system model, since the leadtime at a machine is fixed and assigned; in the system model we had to keep track of the slack for the system and set the α values of each machine to maintain system feasibility. The constraints follow directly once the $\{d_i(D_t^k)\}$ values are given:

$$\alpha_i^1 = 1 \quad i = 1, \ldots, m$$

and

$$\begin{aligned} 0 \leq \alpha_i^j \leq 1 \quad j &= \{d_i(D_t^k)\} \\ i &= 1, \ldots, m \\ k &= 1, \ldots, N \end{aligned}$$

Suppose that the leadtime for D_t^k at machine i is given by $d_i(D_t^k)$. The $\alpha_i^1 = 1$ constraint is familiar from the 1-machine case. The range on the $0 \leq \alpha_i^j \leq 1$ constraint is given for values of j up to $d_i(D_t^k)$. For $j > d_i(D_t^k)$, we can set $\alpha_i^j = 0$ since these terms will never be used. Note that as in the 1-machine system, the values of the parameters are independent of the individual demand streams, and are only a function of the due-date slack remaining for the machine. This follows because we are modelling each unit as a 1-machine system in isolation.

As we will detail in a later section, once the $d_i(D_t^k)$ values have been set, the problem essentially reduces to m 1-machine problems. However, we need to set these values for each of the N incoming demand streams. Since the streams are independent, we will develop the model for one demand stream, with the understanding that the full model is the aggregation of these terms. The optimization program for demand stream D_t^k is denoted $\mathcal{I}_k$:

$$\min \sum_{i=1}^{m} (u_i Var[p_{it}] + v_i E[z_{it}]) \qquad (\mathcal{I}_k)$$

$$\alpha_i^1 = 1 \quad i = 1, \ldots, m$$

and

$$0 \leq \alpha_i^j \leq 1 \; j = \{d_i(D_t^k)\}, \; i = 1, \ldots, m$$

As the reader may be anticipating, we will be changing the form of the program to be explicitly in terms of the variables $d_i(D_t^k)$. We make the transformation in a later section.

Dynamics of Input and Output Processes

In the system model we took some care in formalizing the dynamics of the arrival process, workload, production, and inventory levels. In this model, except for the arrival process, the representation of the other quantities is identical to that expressed in the 1-machine model. We summarize the dynamics of these quantities without further justification. The reader can refer to the 1-machine model for further details.

To characterize the arrival process at machine i in period t, S_{it}:

$$S_{i+1,t+1}(k) = D^k_{t-\sum_{l=1}^{i} d_l(D_t^k)} \quad i = 1, \ldots, m-1 \tag{4.17}$$

where

$$S_{1,t}(k) = D_t^k$$

This follows because having set the values for $d_i(D_t^k)$, each realization of D_t^k moves from machine i to machine $i+1$ at the intervals indicated by the internal leadtimes given by $d_i(D_t^k)$.

We can use this characterization of the arrival process and representations from the 1-machine system to characterize the workload, production and inventory levels in a given period.

For the workload level,

$$\begin{aligned} Q_{i,t+1}(k) &= C\hat{A}_i Q_{i,t}(k) + S_{i,t+1} \\ &= \sum_{k=0}^{\infty} [C\hat{A}_i]^k S_{i,t-k+1} \end{aligned}$$

where C is the familiar update matrix, $\hat{A}_i = I - A_i$.

$$\begin{aligned} p_{i,t+1}(k) &= \alpha_i^T Q_{i,t+1}(k) \\ &= \alpha_i^T \sum_{k=0}^{\infty} [C\hat{A}_i]^k S_{i,t-k} \end{aligned}$$

The dynamics of the inventory level at machine i vary as follows

$$\begin{aligned} z_{i,t+1}(k) &= z_{i,t}(k) + p_{i,t}(k) - \hat{D}_{i,t+1}(k) \\ &= \sum_{j=0}^{\infty} (\alpha^T \sum_{i=0}^{N-1} (C\hat{A})^i D_{t+1-i-j}(k) - \hat{D}_{t+1-j}(k)) \end{aligned}$$

where $\hat{D}_{i,t+1}(k)$ is the quantity of work from demand class k due for completion at machine i at the beginning of period $t+1$.

Special Cases

Again, we approach the problem by first considering a restricted version of the full problem and then use the analysis of this in the evaluation of the full model. In the internal due-date model we consider two cases:

(R1) $v_i = 0 \; \forall i, \;\; u_i \neq 0 \; \exists i$

(R2) The general case with non-zero u_i, v_i

(R1) corresponds to an objective function that is the weighted sum of the production variances of the m machines in the line. The objective function given by (R2) appends a term to (R1) that is a weighted sum of inventory levels of the m machines.

Previous Results from the 1-Machine Case

Since each of the machines then operates as if it were a 1-machine system, we remind the reader of some of the results from the 1-machine model that we will be using in our analysis. From the 1-machine N-demand stream case, we can characterize the variation in production and the mean inventory when using a linear production rule.

$$Var[p_t] = \sum_{i=1}^{N} a_i \sigma_i^2$$

$$\begin{aligned} where \quad a_i &= (1-\alpha_i)^2 a_{i-1} + \alpha_i^2 \\ a_1 &= 1 \end{aligned}$$

and

$$E[z_t] = \sum_{j=2}^{N} b_j \overline{D}^j$$

where

$$b_j = b_{j-1}(1-\alpha_j) + (j-1)\alpha_j$$

and

$$b_1 = 0$$

For the 1-machine case we also identified the smoothing problem as an interesting, non-trivial special case of the full problem. Suppose now that we have only one input demand stream, D_t^k. We summarize the 1-machine results for the case in which there are only smoothing costs, and the case in which there are costs for both variance in production and for finished goods inventory.

Case 1: 1-machine, Minimize $Var[p_t]$

The solution to the smoothing problem,

$$\min_{\alpha} Var[p_t]$$

s.t.

$$0 \leq \alpha_j \leq 1 j = 1, \ldots, k$$

$$\alpha_1 = 1$$

For this program, the solution is given by

$$\alpha_j = \frac{1}{j} \Rightarrow Var[p_t] = \frac{1}{k}\sigma_k^2$$

Case 2: 1-machine, Minimize $Var[p_t] + E[z_t]$

$$\min_{\alpha} Var[p_t] + E[z_t]$$

s.t.

$$0 \leq \alpha_j \leq 1$$

$$\alpha_1 = 1$$

To solve this, we proposed a "sequential" heuristic

- Input: $\sigma_i^2, \overline{D}^i$

- Initialize: $\alpha_1 = 1, a_1 = 1, b_1 = 0$
- Given fixed $[\alpha_1, \alpha_2, \ldots, \alpha_i]$

$$\alpha^*_{i+1} = \max\{0, \frac{2a_i\sigma^2_{i+1} + (b_i - i)\overline{D}^{i+1}}{2(a_i+1)\sigma^2_{i+1}}\}$$

- Successive terms of the recursion are generated by:

$$\begin{aligned} a_{i+1} &= a_i(1-\alpha_{i+1})^2 + \alpha^2_{i+1} \\ b_{i+1} &= b_i(1-\alpha_{i+1}) + i\alpha_{i+1} \end{aligned}$$

We refer the reader to Chapter 2 for further details on the development of these results.

We use these results extensively in the models that we will be discussing. We append subscripts to these terms to indicate the machine in which we are interested.

4.4.2 Analysis of Internal Due-Date Problems: $R1 - R2$

The analysis of the internal due-date problem is not as interesting as that for the system model. We will be directly applying the results of the 1-machine problem, and the problem becomes a due-date assignment problem. The assignment problem is an integer program which has a form that is known to be NP-complete. Thus, we can only make definitive statements about the optimal solution by enumerating all of the feasible solutions. However, it is of interest since we can use it to see how much we lose in de-coupling the system.

(R1) Internal Due-Dates: $v_i = 0 \ \forall i, \ u_i \neq 0 \ \exists i$

In this section, we are looking to solve:

$$\min \sum_{i=1}^{m} (u_i Var[p_{it}]) \qquad (R1)$$

s.t.

$$0 \leq \alpha_j \leq 1$$

$$\alpha_1 = 1$$

Strictly, we should include a subscript on $(R1)$ to indicate that a single demand stream is being considered; we assume that this is understood so that we can reduce the notation in the development that follows.

Suppose we are given a set of internal leadtimes of the form: $d(D_t^k) = \{d_1, d_2, \ldots, d_m\}$. Using the result for the one machine case, we have that the solution to (R1) is:

$$\sum_{i=1}^{m} u_i \frac{1}{d_i} \sigma_k^2$$

Equivalently, we can consider the program $\mathcal{I}_k(Var[p_t])$; here the decision variables are the d_i values.

$$\min \sum_{i=1}^{m} u_i \frac{1}{d_i} \sigma_k^2 \qquad \mathcal{I}_k(Var[p_t])$$

s.t.

$$\sum_{i=1}^{m} d_i = k$$

$$d_i \geq 1, integer$$

As we have noted the problem is one of setting the due-date leadtimes rather than that of setting the work level parameters. Problems of this form have been considered in the literature. The problem can be interpreted as a *discrete, convex knapsack* problem. Zipkin [80] and Bitran and Hax [81] have considered variants of this particular form. Bitran and Hax present a recursive procedure to solve the integer relaxation of $\mathcal{I}_k(Var[p_t])$, which is a lower bound for our problem.

From the lower bound we can construct a "good" feasible solution to $\mathcal{I}_k(Var[p_t])$ by re-arranging the fractional portions. Suppose that a solution to the relaxation is given by $\overline{d} = \{\overline{d}_1, \overline{d}_2, \ldots \overline{d}_m\}$. Let

$$\overline{d}_i = z_i + f_i$$

$$\begin{aligned} \text{where } z_i &= \lfloor \overline{d}_i \rfloor \text{ is the integer portion of } \overline{d}_i \\ f_i &= \overline{d}_i - \lfloor \overline{d}_i \rfloor \text{ is the fractional portion of } \overline{d}_i \end{aligned}$$

We know

$$\sum_{i=1}^{m} f_i \in \mathcal{Z}^+$$

because $\overline{d}$ is feasible for $\mathcal{I}_k^{relax}(Var[p_t])$ Also, $\sum_{i=1}^{m} f_i \leq m$ regardless of the problem instance. This is straighforward because we are only adding up at most m fractional components to yield this $\sum_{i=1}^{m} f_i$ quantity

In this way, z_i periods have been assigned to the ith machine, and $\sum_{i=1}^{m} f_i$ periods are remaining to be assigned. Any distribution of these $\sum_{i=1}^{m} f_i$ periods is feasible for $\mathcal{I}_k(Var[p_t])$. An easy assignment is $d = \{d_1, d_2, \ldots, d_m\}$ to $(R1)$:

$$\begin{aligned} d_i &= z_i \; i = 1, 2, \ldots, m-1 \\ d_m &= z_m + \sum_{i=1}^{m} f_i \end{aligned}$$

To find a local optimal solution, we do an interchange of the current assignment that evaluates the net marginal benefit of a reduction in the leadtime of machine i from d_i to $d_i - 1$ and the increase of the leadtime of some machine l from d_l to $d_l + 1$.

First, we introduce the variables Δ_i^+ and Δ_i^- to capture the impact on $(R1)$ of increasing/decreasing, respectively, the due-date leadtime assignment.

Δ_i^+ represents the marginal change in the variation cost of the change $d_i \rightarrow d_i + 1$

Δ_i^- represents the marginal change in the variation cost of $d_i \rightarrow d_i - 1$.

To do this, for each i, calculate:

$$\begin{aligned}\Delta_i^+ &= u_i[\frac{1}{d_i} - \frac{1}{d_i+1}] = u_i\frac{1}{d_i(d_i+1)} \\ \Delta_i^- &= u_i[\frac{1}{d_i-1} - \frac{1}{d_i}] = u_i\frac{1}{d_i(d_i-1)}\end{aligned}$$

In an interchange operation we start with a feasible solution $\hat{d} = \{\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_m\}$ and then, selecting two machines i, l, reassign the length of the internal due-date between the two to maintain feasibility:

$$\begin{aligned}&d_i \rightarrow d_i - 1 \\ &d_l \rightarrow d_l + 1\end{aligned}$$

To get a locally optimal solution we can then do a "greedy" interchange. For this,

$$\Delta\mathcal{I}_k = \max_{i,l}(\Delta_l^+ - \Delta_i^-) \tag{4.18}$$

where the current leadtime for machine l is increased by one unit and that for machine i is decreased by one unit. In this way we can construct a local optimum.

Example

If the number of machines in this system is small, then enumeration of the solutions

k	$u_1 = 4, u_3 = 1$
4	5.00
5	3.00
6	2.33
7	1.83
8	1.50
Total	13.66

Table 4.7: $\mathcal{I}_k(Var[p_t])$ via enumeration

may be very easy to do. We return to the example for which there are four machines and five demand streams, $D_t^4, D_t^5, D_t^6, D_t^7, D_t^8$. We enumerate the values of $\mathcal{I}_k(Var[p_t])$ in Table (4.7), for each stream k when there are two non-zero parameters $(u_1, u_3) = (4, 1)$. The solution is the same when we reverse the paramters so that $(u_1, u_3) = (1, 4)$ because the machines operate independently.

4.4.3 (R2) Internal Due-Dates: v_i, u_i not all 0

For objective function

$$\min \sum_{i=1}^{m} u_i Var[p_{it}] + v_i E[z_{it}] \qquad\qquad \mathcal{I}_k(Var[p_{it}], E[z_{it}])$$

The introduction of the inventory costs means that we cannot use the very concise representation that we used in evaluating $(R1)$. We have the general representation of the smoothing and inventory costs given above. As before this type of objective function requires an explicit assessment of the relative importance of the smoothing versus holding costs over all of the machines. As in the case of problem (R1), we propose that in order to make the representation and analysis tractable, that if we are given a feasible vector of internal due-date leadtime assignments, that we solve the problem for each machine in the way that we did for the 1-machine case. In this way, the real decision variable is the length of the due-date window; in a sense this is like being able to control the due-date of an "input stream" to a machine i, although limited by the actual exogenous due-date request.

Intuitively, the tradeoffs we expect to see:

$$u_i >> v_i \rightarrow \text{Longer window important}$$
$$v_i >> u_i \rightarrow \text{Longer window not as important}$$

The reasoning here is that when the variance is significant, then a longer window permits more smoothing. When the variance is not as important, then the tendency is to delay production in any case, and so the longer window yields little benefit.

Example (R2) : $u_i = 0, v_i = 0,\ i = 2, 4$
In this example we would like to solve

$$u_1 Var[p_{1t}] + v_1 E[z_{1t}] + u_3 Var[p_{3t}] + v_3 E[z_{3t}] \tag{4.19}$$

where there are five input demand streams, $D_t^4, D_t^5, D_t^6, D_t^7, D_t^8$. However, even for the 1-machine case we do not have an easily obtainable solution, and so we cannot solve Equation (4.19) very readily. As an alternative, we examine several special cases of this problem to get some insight that may be useful in a more general context. We proceed as follows:

- We assume several different sets of values for the parameters, i.e, values for $\{(u_1, v_1), (u_3, v_3)\}$
- For each set of $\{(u_1, v_1), (u_3, v_3)\}$ we apply the 1-machine sequential search heuristic for each of the machines $\{1, 3\}$ to come up with values for the parameters at each station i, α_i
- We enumerate the feasible due-date assignments and the corresponding value of the function (4.19)

	Ex. 1	Ex. 2	Ex. 3	Ex. 4
(u_1, v_1)	(1,4)	(4,1)	(4,4)	(1,1)
(u_3, v_3)	(1,4)	(4,1)	(1,1)	(4,4)

Table 4.8: Matrix of (u_i, v_i) values: Four examples

We consider four examples, given by varying the parameters $\{(u_1, v_1), (u_3, v_3)\}$. We summarize these in Table (4.8)

Without loss of generality, we can assume that

$$u_i = 0 \Rightarrow d_i(D_t^k) = 1, \ \forall k$$

in any optimal solution. This follows because the only reason we are interested in having a due-date window of length > 2 is to achieve smoothing benefits. With $u_i = 0$ there is no cost to variation in production at machine i.

Next, given values for the parameters (u_i, v_i) for machine i we apply the sequential heuristic of the 1-machine model. The input of the pair (u_i, v_i) results in a vector α for the machine, denoted by:

$$(u_i, v_i) \rightarrow \alpha$$

Using the examples that we have presented in Table (4.8) we find the corresponding α to be:

$$(1, 4) \rightarrow \alpha = \{1, 0, 0, \ldots\}$$

$$(4, 1) \rightarrow \alpha = \{1, 0.44, 0.21, 0.06, 0, 0, \ldots\}$$

$$(4, 4) \rightarrow \alpha = \{1, 0.25, 0, 0, \ldots\}$$

Machine	(u_i, v_i)			
i	(1,4)	(4,1)	(4,4)	(1,1)
1	1.00	4.00	4.00	1.00
2	1.00	2.47	3.50	0.88
3	1.00	2.21	3.50	0.88
4	1.00	2.19	3.50	0.88

Table 4.9: $u_i Var[p_{it}] + v_i E[z_{it}]$ using sequential heuristic

k	$d_1(D_t^k)$	$d_2(D_t^k)$	$d_3(D_t^k)$	$d_4(D_t^k)$	Ex1	Ex2	Ex3	Ex4
4	1	1	1	1	1.00	8.00	5.00	5.00
5	2	1	1	1	1.00	6.47	4.50	4.50
6	2	1	2	1	1.00	4.94	4.38	4.38
7	3	1	2	1	1.00	4.68	4.38	4.38
8	3	1	3	1	1.00	4.42	4.38	4.38

Table 4.10: $\mathcal{I}_k(Var[p_{it}], E[z_{it}])$ via enumeration of $d_i(D_t^k)$ values

$(1,1) \rightarrow \alpha = \{1, 0.25, 0, 0, \ldots\}$

We use these values of α to determine the value of $u_i Var[p_{it}] + v_i E[z_{it}]$ given in Table (4.9). We use the values of α that resulted from the sequential heuristic.

To return to the initial example in which we have a serial line with 4 machines, and five input streams, $D_t^4, D_t^5, D_t^6, D_t^7, D_t^8$, we enumerate the solutions:

- For a set of due-date assignments, $\{d_i(D_t^k)\}$, we use the sequential heuristic to set the values for alpha
- Enumerate through the values of the due-date assignments

In Table (4.10) we present the best due-date assignment for each incoming demand stream according to the sequential heuristic via enumeration. The value of $d_2(D_t^k) = 1, d_4(D_t^k) = 1$ in all of these cases, as we had assumed above.

In using Table (4.10) we choose the minimum values from the enumeration to correspond to the due-date assignment. In this case we can then evaluate the objective function (4.19), using the values given in Table (4.10), and add across demand streams.

A few things to note: The optimal solutions to Examples 3 and 4 were the same. This is because they had identical, reversed costs. This solution is independent of the machine at which it is taking place. We discuss some of the other features of these solutions when we compare the system and internal due-date model.

4.5 Discussion of System and Internal Due-Date Scenarios

We began this chapter by discussing why one might be interested in approaching the multi-machine version of the rigid due-date problem in two different ways. Now, we compare the analyses of the two models. *Apriori*, we expect that when the objective function is specified in terms of the system performance the system model should perform better. When we have only smoothing costs this is true. However, when there are inventory costs the two models are not directly comparable. Specifically, in the internal due-date model, we do not incur work-in-process costs at a machine when work is produced immediately before its due-date; in the system model, there is still an inventory cost that is being incurred with respect to the preceding machine.

In Table (4.11) we summarize the values of the objective function

$$u_1 Var[p_{1t}] + u_3 Var[p_{3t}] \tag{4.20}$$

where for the internal due-date, these are given by the enumeration solution of (R1), and the solution for the system model corresponds to the (Q2) solution. We see in

(u_1, u_3)	(Q2)	(R1)
(4,1)	10.79	13.66
(1,4)	8.47	13.66

Table 4.11: Compare (Q2) vs. (R1)

	Ex. 1	Ex. 2	Ex. 3	Ex. 4
$\alpha(L)$	56.46	29.04	46.44	39.06
$\alpha(Q2A)$	50.18	28.22	46.43	31.97
$\alpha(Q2B)$	46.59	26.46	41.43	31.62
$\alpha(R3)$	5.00	28.51	22.64	22.64

Table 4.12: Compare all policies at values for u_i, v_i

Table (4.11) that the system model substantially outperforms the internal model. In the second example with $(u_1, u_3) = (1, 4)$ (Q2) is especially good because the internal model cannot incorporate smoothing that may have occurred upstream.

In Table (4.12) we summarize the values of the objective function

$$u_1 Var[p_{1t}] + v_1 E[z_{1t}] + u_3 Var[p_{3t}] + v_3 E[z_{3t}] \qquad (4.21)$$

where for the internal due-date, these are given as a function of the due-date assignment. For the policies (L) and the policy generated via $(Q2)$ we evaluate the terms according to the functions $x^j(i), y^j(i)$. As we indicated above, to make the appropriate comparison of the terms in Table (4.12) we need to add back in the costs of intermediate inventory holding. The example that comes closest to being a reasonable comparison is Example 2, where the inventory costs are small. In this case, even without adding back anything to account for the inventory cost terms, the system solutions outperform the internal model.

Additionally, in explicitly comparing the two models, it may be appropriate to extend the objective function being considered. Specifically, there may be other types of "maintenance costs" that may apply for each model. For example, in the system model, there may be costs of gathering, maintaining and distributing large quantities of data. In the internal due-date model, we do not have any costs that represent either costs of decoupling or of integration.

4.6 Extension to a Job-Shop Model

Up until now we have only considered systems that are flow lines: At each station there is exactly one type of input stream, characterized by a certain mean and variance, and there is exactly one type of output stream, generated by application of a fixed linear control to the inputs. Our study of this type of system has proven to be very useful and we have been able to determine the optimal policy for minimizing certain measures which are a function of the variance of output at the individual production sites.

Will a similar type of analysis be possible when there are multiple types of inputs and outputs such as in a job shop? In what manner should we maintain information about routings and due dates of the different jobs? What type of objective function is most appropriate? Our previous analysis takes into account that there is one "end" station in the line. The information about this is used for setting the parameters to be feasible, and for setting the initialization terms of the recursion.

The type of policy that we will propose in this note is motivated by the observation that each station i can be thought of as a station in several serial lines, with inclusion in a particular line when a demand type has station i on its routing. As a result, it is not sufficient to maintain a single workload vector– all due dates are not "equal", since it is equally important to know the actual routing of a demand type. Suppose that at station i there are J_i classes of jobs that have station i as a part of the routing. J_i can, for example, correspond to the actual set of routings that include station i.

On what basis should the classes be distinguished? The complication of multiple routings in the system suggests that explicit consideration of each routing entering station i should define each class. As one would expect, the notation here gets pretty messy, but conceptually it is no more difficult.

There is a lot of scope here for looking at more complicated systems, using our analysis of the serial line as a building block.

Chapter 5

Conclusions

5.1 Overview

In this thesis we have developed several models for production planning for a make-to-order system which must satisfy exogenously-given rigid due-dates. The costs of primary interest at an individual machine in the system were those of variation in the production level, and the quantity of inventory being held. We then considered objective function given in terms of these costs.

The first model we proposed was the basic model from which we began to gain intuition about these make-to-order, due-date constrained systems. In a very high-level sense, we could view a large system as a "1-machine" unit and use this to get an aggregate characterization of the system. Another case for which the 1-machine case may be of interest is when we have a system with one basic operation, which we might simplify to the 1-machine case.

As one extension of this model, we considered the planning problem facing a a producer who operates simultaneously in several markets. In particular, the producer differs in its response to each class of customers. For the customers that are considered less important, the producer may exceed the due-date request given by the customer. We offer some interpretations and models about a 1-machine production system with non-rigid due-dates.

As another extension to the one-machine problem with rigid due-dates we considered a multi-machine version of the problem. We offer two interpretations of this models, one corresponding to a system view and the other to the machine perspective.

5.2 Contributions and Observations

We began this thesis with the motivating problem faced by the printing facility at International Paper. How can our models be of interest to the issues at IP? The management there had begun making some elementary ventures into the type of aggregate models that we have been discussing. The models in this thesis should be useful in terms of formalizing the planning concept, rather then scheduling work on a more haphazard fashion, allocating work on an "as need" reactive basis.

We feel that these models may be very useful in the context of production planning. As always, we need to be careful in how the model is constructed, with respect to parameters, etc. However, the models we have developed are not highly problem specific and may find a wide range of application.

Our primary contribution to this area has been in developing models to analyze the planning problem for a make-to-order system. The make-to-order system with due-dates had not been considered before. In today's manufacturing environment, however this type of scenario is developing. In particular, in a JIT environment, there is an increasing emphasis on lowering inventory levels, which lessens the need for traditional make-to-stock models. Make-to-order systems may be more representative of the type of production systems that are evolving. There have been few models of such systems.

5.3 Directions for Future Research

We see that there may be several interesting extensions to the models we have presented. We briefly indicate these:

Due-date quotation: How can we use our models so that when a customer enters the system that the producer can either accept the request, or counter with a different deadline, where there is incentive for the customer to do so.

We need to work more with actual production systems and see how this can be incorporated into the real planning schemes. Also, how does this work with other aspects of manufacturing?

Computational work: Now we have a reasonable framework for studying systems that must accomodate rigid due-date requests. For specific problem instances we can test many different heuristics.

Chapter 6

Annotated Bibliography

This bibliography is grouped into three sections. The scenario we described is a dynamic, make-to-order production system with stochastic demand inputs and due-date restrictions. We are interested in developing production loading policies to satisfy the due-date constraints while maintaining smooth production levels. These problems in this context has not been addresses in the literature. However, there is a large body of research considering similar problems in a different context. For the sake of continuity, we have separated the literature review from the main body of the thesis.

In the bibliography that follows we give representative sources for the particular variation of the problem at hand. More extensive bibliographies are available in the individual references.

6.1 Production Smoothing

1. Cruickshanks, Allan B., Robert D. Drescher, and Stephen C. Graves, 1984,

"A Study of Production Smoothing in a Job Shop Environment," *Management Science*, 30, 368-380.

2. Graves, Stephen C., 1986, 'A Tactical Planning Model for a Job Shop,' *Operations Research*, 34, 522-533.
3. Hax, Arnoldo C., 1978, "Aggregate Production Planning," in *Handbook of Operations Research*, J, Moders and S. Elmaghraby (Eds.), Van Nostrand Reinhold, New York.
4. Holt, Charles C., Franco Modigliani, and Herbert Simon, "A Linear Decision Rule for Production and Employment Scheduling," *Management Science*, 2, 1-30.
5. Kleindorfer, Paul R., and K. Glover, 1973, 'Linear Convex Stochastic Optimal Control with Applications in Production Planning," *IEEE Transactions on Automatic Control*, AC-18, 56-59.
6. Kleindorfer, Paul R., et al, 1975, "Discrete Optimal Control of Production Plans," *Management Science*, 22, 261-273.
7. Modigliani, Franco and Franz E. Hohn, 1955, "Production Planning Over Time," *Econometrica*, 23, 46-66.
8. Silver, Edward A., 1967, "A Tutorial on Production Smoothing and Work Force Balancing," *Operations Research*, 16, 195-207.
9. Sobel, Matthew J., 1969, "Production Smoothing with Stochastic Demand I: Finite Horizon Case," *Management Science*, 16, 195-207.
10. Sobel, Matthew J., 1969, "Production Smoothing with Stochastic Demand II: Infinite Horizon Case," *Management Science*, 17, 724-735.

Silver[67] gives the definitive tutorial on the production smoothing problem in a general make-to-stock environment. He identifies the primary costs of interest:

- Basic production costs
- Costs of changes in production rate
- Inventory costs
- Shortage costs.

The cost of changes in the production rate is typically represented as a convex function. The other three cost are given by linear functions. Note that the basic production cost for an individual item is incurred regardless of the other three types of costs, for a fixed type of production process. For medium-range planning, then, it assumes a minor role. Additionally, the inventory and shortage costs when combined result in a convex function. Thus, as a simplification, the costs are often given as a general convex cost function.

In this paper, Silver goes on to examine the problem of production smoothing. The problem arises in an environment of stochastic demand. There are traditionally three ways in which to react to fluctuations in demand:

- change the size of the labor force
- work overtime or idle
- let inventories increase or decrease

The references to production smoothing date back to Holt, Modigliani and Simon [1955] in their classic model of production planning. They were the first to propose the linear decision rule, arising from the quadratic cost function. Sobel [1969, 1971] considers a more general form of the problem: Costs related to inventory and changes in production are convex, and the costs and demands may have non-stationary distributions. He considers the problem in both the finite and infinite horizon. Kleindorfer and Glover[1973] apply the methodology of stochastic optimal control to production smoothing. Kleindorfer et al[1975] present the application of discrete optimal control theory to deterministic production planning. Hax[1978] gives an updated overview.

In all of these latter papers due-date requirements are not explicitly considered, because of backlogging assumptions. Only in Cruickshanks et al[1984] is the make-to-order scenario considered.

6.2 Due-Dates

1. Bertrand, J. W. M., 1983, "The Effect of Workload Dependent Due-Dates on Job Shop Performance," *Management Science*, 29, 799-816.

2. Conway, R. W., W. L. Maxwell and L. W. Miller, 1967, *Theory of Scheduling*, Addison-Wesley, Reading, MA

3. Eilon S., and I. G. Chowdhury, 1976, "Due-Dates in Job Shop Scheduling," *International Journal of Production Research*, 14, 223-237.

4. Rinooy Kan, A. H. G., 1976, *Machine Scheduling Problems: Classification, Complexity and Computation*, Nijhoff, The Hague, Netherlands.

5. Seidmann, A. and M. L. Smith, 1981, "Due-Date Assignment for Production Systems," *Management Science*, 27, 571-581.

6. Wein, Lawrence M., to appear 1991, "Due-Date Setting and Priority Sequencing in a Multiclass M/G/1 Queue," *Management Science.*

The research here falls into two very different categories. The first is the classical combinatorial analysis. The typical problem here is deterministic, with objectives of minimizing a function of the lateness of the jobs to be processed (where lateness can be positive or negative). Rinooy Kan[1976] gives a good overview of this area.

In the second area of research known as due-date quoting, the due-date or delivery date to a customer is generated by the producer, based on given anticipated production plans. This is almost a converse of the problem in which we are interested. Conway, Maxwell and Miller[1967] discuss this briefly. A lot of simulation work has been done in this area, e.g., Eilon and Chowdhury[1976]; they find that measures of system performance improve when the due-date quotes take into account work in the system, Bertrand[1983] gives an analytical procedure to come to a similar conclusion. Seidmann and Smith[1981] formulate a cost function quantifying the tradeoffs between operational conerns (meeting due-dates) and strategic (desire to quote attractive leadtimes). From this they derive a constant due-date assignment.

We are not primarily interested in due-date quotation, although some of this methodolgy may have bearing in our study of flexible due-dates. In some very recent work, Wein[1991] has begun simultaneously considering due-date quotations and scheduling policies, and this is interesting because of the dual problems considered.

6.3 Other

1. Ahuja, Ravindra., Thomas L. Magnanti, and James B. Orlin, 1992, *Network Flows: Theory, Algorithms, and Applications*, (text in preparation) Prentice-Hall, New York.

2. Bertsekas, Dimitri P., 1987, Dynamic Programming: Deterministic and Stochastic Models, Academic Press, New York.

3. Bitran, Gabriel, and Arnoldo Hax, 1981, "Disaggregation and Resource Allocation Using Convex Knapsack Problems," *Management Science*, 27, 431-441.

4. Kleinrock, Leonard, 1975, *Queueing Systems Vol I and II: Theory and Applications*, John Wiley and Sons, New York.

5. Loeve, Michel, 1977, *Probability Theory I*, Springer-Verlag, New York.

6. Luenberger, David, G., 1984, *Linear and Non-linear Programming*, Addison-Wesley, Reading, MA.

7. Ross, Sheldon, 1983, *Stochastic Processes*, John Wiley and Sons, New York.

8. Stalk, George, 1988, "Time– The Next Source of Competitive Advantage," Harvard Business Review, July-August, 41-51.

9. Zipkin, Paul, 1980, "Simple Ranking Methods for Allocation of One Resource," *Management Science*, 26, 34-43.

Blinded by the Light

– Bruce Springsteen

Mad man drummers bummers and Indians in the summer
with a Teenage diplomat
In the dumps withe the mumps as the adolescent humps
his way into his hat
With a boulder on my shoulder feeling kind of older:
I tripped the merry-go-round
With an unpleasing sneezing and wheezing
the calliope crashed to the ground

Some all hot half shot was headed for the hot spot
Snapping his fingers, clapping his hands
And some fleshpot mascot was tied into a lovers knot
with a whatnot in her hand
And now young Scott with a sling shot finally found a tender spot
and throws his lover in the sand
And some bloodshot forget-me-not whispers Daddys within earshot
Saved the buckshot turn up the band

And she was blinded by the light
Cut loose like a deuce
Another runner in the night
Blinded by the light
She got down, but she never got tight
Oh, shell make it alright

Some brimstone baritone anticyclone rolling stone

preacher from the east
He says dethrone the dictaphone hit it in its funny bone
thats where they expect it least
And some new mown chaperone was standing in the corner all alone
watching the young girls dance
And some fresh sown moonstone was messing with his frozen zone
to remind him of the dealing of romance

And she was blinded by the light
Cut loose like a deuce
Another runner in the night
Blinded by the light
He got down, but he never got tight
Hes gonna make it tonight

Some silicone sister with her manager mister
told me I got what it takes
She said I turn you on sonny to something strong
if you play the song with the funky break
And gocart Mozart was checking out the weather chart
to see if it was safe to go outside
And little early pearly came by in her curly whirly
and asked me if I needed a ride

And some hazard from Harvard was skunked on beer
playing backyard bombadier
Yes, and Scotland Yard was trying hard they sent some dude with a calling card
he said do what you like but dont do it here

While I jumped up turned around spit in the air fell on the ground
asked him which was the way back home
He said take a right at the light keep going straight until night
and then boy youre on your own

And now in Zanzibar a shooting star was riding in a sidecar
humming a lunar tune
Yes, and the avatar blow the bar but first remove the cookie jar
were going to teach these guys to laugh too soon
And some kidnapped handicap was complaining that he caught the clap
from some mousetrap he bought last night
Well I snapped his skullcap and between his ears I saw a gap
but figured he would be alright

And she was blinded by the light
Cut loose like a deuce
Another runner in the night
Blinded by the light
She got down, but she never got tight
Oh, shell make it alright

Mama always told me not to look into the sights of the sun
Oh, but mama, thats where the fun is