SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK 14853

TECHNICAL REPORT 697

June 1986

# THE LOT SCHEDULING PROBLEM IN THE HIERARCHY OF DECISION MODELS

By

Robert Sheldon

This research was supported by National Science Foundation Grant No. DMC-8509131.

THE LOT SCHEDULING PROBLEM

IN THE HIERARCHY OF DECISION MODELS

Robert Sheldon, Ph.D.

Cornell University 1986

The economic lot scheduling problem (ELSP) surfaces from competition among products for a scarce resource, usually machine time. Standard approaches to the ELSP look only at scheduling multiple products on a single machine. To put the scheduling problem in proper context, we examine how scheduling decisions on a machine affect and are affected by other decisions and the physical structure of the system. This thesis addresses three important issues that put the scheduling problem in the context of its physical setting and range of parameters: idle time, the zero switch rule, and stochastic input to a bottleneck machine.

In most scheduling heuristics, the reason for idle time is to balance the cyclic production patterns. Idle time is also optimal in solutions to problems with high setup costs. We show that the condition for inducing idle time, given zero setup costs, is when one product has dominant holding costs and the remaining products have low machine utilization.

A common policy in scheduling is to start production only after the inventory reaches zero. This policy is called the zero switch rule

(ZSR) and is regarded as a good scheduling policy. We show that the condition when the ZSR is not optimal is when the ZSR solution yields lumpy production patterns for a product with dominant holding costs.

The standard approach to scheduling considers the input process to be deterministic and ignores delivery of raw parts. This thesis examines scheduling a bottleneck machine with stochastic inputs under a variety of situations. First, we isolate the issue of scheduling deliveries to a machine. We look at using state information to schedule the pre-bottleneck machines. Next, we consider an aggregate planning model to evaluate both lot scheduling and delivery and develop an algorithm for solving this problem. We show that the conditions when we should consider the delivery issue in conjunction with the lot sizing issue are when the holding cost of raw parts is high and when the variance of delivery times is high. Then we examine a dynamic programming formulation and consider a variation of the assembly model.

TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

The lot scheduling problem surfaces from competition among products for a single scarce resource, usually machine time. Batch production often is a natural consequence of manufacturing several products on the same machine, unless setup times are negligible. When scheduling production of batches in an environment having a single constrained resource, two issues must be resolved: the size of the production batches and the start times for production of each batch. The solution to this problem is computationally difficult (Hsu, 20) and therefore practical solutions must be obtained using heuristic techniques. This scheduling problem occurs so frequently that it has led to a standardized characterization as the Economic Lot Scheduling Problem (ELSP).

Standard approaches to the ELSP look only at the issue of scheduling multiple products on a single machine. However, in a more realistic setting, we must ask what is the real problem we are trying to solve? Couched in this setting is a hierarchy of decisions that must be made at different levels of management at different points in time and at different locations within the manufacturing system. Hence, to put the scheduling problem in the proper context, we must examine how the scheduling decisions on that machine affect and are affected by other decisions and the physical structure of the system.

The hierarchy of decisions for most manufacturing environments can be broken into the following four levels:

Manufacturing Systems Planning (or Capacity Planning)

Production Planning

Flow Planning

and Scheduling.

The decisions made at these various levels range from design of facilities down to real time detailed scheduling. Any modelling approach to a manufacturing system should therefore address how the model fits into the framework or structure of the overall decision process as well as how the problems, issues, and decisions from that model interact with the rest of the system.

The standard approach to lot scheduling considers the input process to a machine to be deterministic. Therefore, the issue of delivery of these raw parts is ignored when scheduling the machine. If the delivery of the parts is indeed deterministic, then deliveries can be scheduled to arrive just-in-time, and hence, the scheduling of a machine can be looked at independently of its predecessors. However, if the inputs to a machine are stochastic, that is, processing times on the predecessor machines are variable or the overall delivery times are inconsistent, then the schedule of the machine should be developed considering the issue of delivery of raw parts to the machine.

A traditional ELSP assumption is that the demand process is constant and continuous. An important issue in the physical context of the machine environment which deviates from this assumption arises if all different parts produced on a machine are subsequently assembled. In the assembly model, the work-in-process (WIP) inventory is held until all parts are available for assembly in addition to the normal inventory accumulated due to batch production. In this case, we get a different inventory pattern from the traditional ELSP sawtooth inventory pattern. This requires a different approach for the assembly model from the tradtional ELSP approach. Another variation occurs if the demand process is dynamic and backlogging of demand and machine capacity is permitted. These problems can be addressed in a general way for arbitrary ranges of parameters, however for restricted ranges of the parameters, a different method may work better.

Traditional approaches to the ELSP use the same solution technique for all scheduling problems regardless of the range of the parameters involved. If we place the problem in the context of the parameters involved, an important concept to look at before addressing any given scheduling problem is the notion of a dominant product. By this is meant that one of the products has its parameters such that any solution to the problem will always be dominated by that product. If this is the case for a given problem, we can focus in on solution approaches that take this into consideration. Using this notion, we can get better solutions for the restricted class of problems without the added effort of a generalized solution technique.

The concept of a dominant product plays heavily in looking at whether to induce idle time into the lot scheduling problem. In most heuristics, the reason for idle time is to balance the cyclic production pattern to accomodate non-rotation cycles. Idle time is also optimal in solutions to problems with artificially high setup costs, that is, the cost of setup is higher than the imputed value of lost machine capacity due to the setup time. Other than these cases, if we assume zero setup costs, then an optimal schedule would rarely have idle time unless one product is dominant.

A common policy in scheduling is to start production of a particular product only after the inventory of that product reaches zero. This policy is called the zero switch rule (ZSR) (Maxwell, 24) and has generally been regarded as a good scheduling policy. The ZSR seems intuitively to be sound if we are trying to minimize inventory costs. However, if we find an optimal solution restricted to the ZSR policy, there are cases where we can improve the solution by incorporating a non-zero switch, that is, for some product, start production before its inventory reaches zero.

The remainder of this dissertation examines the aforementioned issues of idle time, the ZSR, and stochastic input to a bottleneck machine. In chapter 2, a review of relevent literature is provided. This includes a discussion of the traditional ELSP and a review of hierarchical models which have lot scheduling embedded in their structure.

Chapter 3 looks at two important issues which have surfaced in the traditional ELSP literature, idle time and the zero switch rule (ZSR). In particular, we find that the conditions where inserting idle time improves a solution to the ELSP, given zero setup costs, are very restrictive. In addition, we verify that the ZSR is a good scheduling policy for most problems and give explicit conditions to show instances when we can do better with a non-zero switch.

In chapter 4, we examine scheduling a bottleneck machine with stochastic inputs under a variety of situations. First, we isolate the issue of scheduling deliveries to a bottleneck machine. We look at using current state information to schedule the pre-bottleneck machines. Next, we consider an aggregate planning model to evaluate both lot scheduling and delivery to the bottleneck machine. Then we examine a dynamic programming formulation under varying demand and relax the constraint that all demand be satisfied during each period. Finally, we consider a variation of the assembly model.

Chapter 5 develops an algorithm for solving the aggregate planning model of chapter 4 and shows when the delivery issue is important as well as the impact on the system of variability in the delivery process.

Chapter 6 presents conclusions regarding scheduling idle time, the ZSR, and scheduling a bottleneck machine with stochastic inputs.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 The Traditional Economic Lot Scheduling Problem (ELSP)

### 2.1.1 Background

The traditional ELSP addresses scheduling production of several products on a single machine. Elmaghraby (11) provides an excellent review of the traditional ELSP literature through 1977. The following are common notation and assumptions (with i being the index referring to a particular product):

$r_i$ demand rate in parts per unit time (constant, continuous)

$p_i$ production rate in parts per unit time (constant)

$\rho_i = r_i / p_i$ relative utilization of the machine by product i

( $\Sigma \rho_i \leq 1$ for feasibility)

$s_i$ setup time per production lot of product i

(assumed independent of sequence)

$A_i$ setup cost per production lot

(assumed independent of sequence)

$h_i$ holding cost per part per unit time

$T_i$ length of repeatable production cycle for product i

$H_i = 1/2 \; h_i r_i (1 - \rho_i)$ scaled holding cost when $T_i = 1$

The average cost for product i is

$$C_i = A_i / T_i + h_i r_i (1 - \rho_i) T_i / 2$$

$$= A_i / T_i + H_i T_i .$$

The objective of the ELSP is to find $T_i$ and start times for each product which give a feasible schedule at minimum possible cost. A feasible schedule is one which can be defined on a Gantt chart over any given time horizon such that the demand for each part is satisfied throughout the time horizon and the machine is never scheduled for more than one activity at any point in the time horizon. A schedule is infeasible if the machine is required to work on more than one product at any time.

### 2.1.2 Independent Solution (IS) to the ELSP

Suppose each product can be produced on independent machines with the constraint that total machine time used is equivalent to one machine. Assume inventory of each product follows the pattern

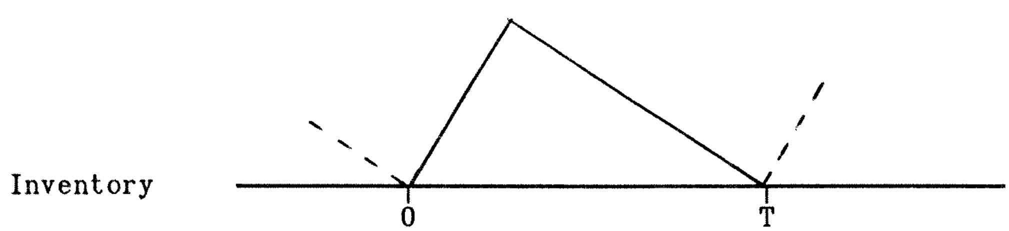



FIGURE 2-1

One Product Inventory Pattern

Average inventory cost $= h_i r_i T_i (1-\rho_i)/2 = H_i T_i$ .

We can formulate this problem as

$$\text{(2-1)} \qquad \text{minimize } \sum H_i T_i + \sum A_i/T_i$$

$$\text{(2-2)} \qquad \text{subject to } \sum s_i/T_i + \sum \rho_i \leq 1 .$$

First assume that (2-2) is not binding and solve (2-1) to get

$$T_i^* = (A_i/H_i)^{1/2} .$$

If this solution is not binding in constraint (2-2), then the theoretical optimum solution does not fully utilize machine capacity. The machine capacity utilized in this solution is

$$U = \sum s_i (H_i/A_i)^{1/2} + \sum \rho_i .$$

Observe that the higher we set the values of $A_i$, the less machine capacity we use in the theoretical optimum solution. The theoretical lower bound in this case is

$$C(T_i^*) = 2 \sum (A_i H_i)^{1/2} .$$

<u>Zero setup costs.</u>

In the case of zero setup costs, use Lagrangian relaxation on (2-1) and (2-2) to get

$$T_i^* = (s_i/H_i) \frac{\left\{ \sum (H_i s_i)^{1/2} \right\}}{\left\{ 1 - \sum \rho_i \right\}} .$$

Which gives the lower bound on solutions of this form with zero setup costs

$$LLB = \frac{\left\{ \sum (H_i s_i)^{1/2} \right\}^2}{\left\{ 1 - \sum \rho_i \right\}} .$$

### 2.1.3 Basic Period (BP) Approach

The BP approach restricts each $T_i$ to be an integer multiple of some basic period w, that is, $T_i = K w$, where K is an integer. Several authors have developed iterative algorithms using this approach, for example, Elmaghraby (11), Doll and Whybark (10), Fijita (12), and Madigan (23). These iterative techniques involve computing a basic period w and rounding cycle times $T_i$ to integer multiples of w, then checking for feasibility. If feasibility is not achieved, they iterate to a new basic period or modify the solution. A BP solution is feasible if a cyclic production schedule can be found such that the requirements in any given period, including setup and production time, do not exceed the length of the base period w. Hsu (11) showed that the check for feasibility is NP-hard.

### 2.1.4 Power-of-Two Restriction

Doll and Whybark (10), Delporte and Thomas (7), and Fujita (12) recommended restricting cycle times to power-of-two times some base period to simplify the search for a feasible schedule. Maxwell and

Singh (28) showed that restricting cycle times in the ELSP to power-of-two times some base period would yield solutions within 6% of unrestricted optimal cycle time solutions. Roundy (29) developed an algorithm to perform the power-of-two roundoff of order intervals which produced solutions within 6% of optimal solutions. Hence, the power-of-two roundoff technique provides a simple way to get solutions which are very close to optimality.

## 2.2 ELSP Imbedded in the Hierarchy of Models

The key feature of hierarchical models is that decisions are made at one level under consideration of their impact at other levels. The linking of levels of decision making then leads to better overall solutions to the system being modeled. In many hierarchical models involving mathematical programming, this linking takes the form of Lagrangian multipliers or changes to the constraints (see, for example, Graves, 15). Maxwell, Muckstadt, Thomas, and VanderEecken (27) propose a general modeling framework for production control consisting of the following three phases: creating a master production plan, planning for uncertainty, and real-time resource allocation. Although they don't present any detailed models, their approach shows how various models and algorithms can effectively be placed in a hierarchical decision-making structure. Hax and Meal (18) give reasons for using a hierarchical approach and develop a model where decisions at the aggregate level provide constraints for the lower levels. In this model, they link capacity planning decisions with detailed scheduling decisions. Grave's model (15) decomposes a large scale production planning problem into two subproblems corresponding to the Hax-Meal model and also provides feedback between detailed scheduling decisions and capacity planning decisions. Dempster, et al., (8), point out that the two fundamental reasons for using a hierarchical approach are to reduce the complexity of the solution process and to cope with uncertainty. Bitran, Haas and Hax (2) present a hierarchical production planning model where aggregate planning is done first followed by sequential levels of disaggregation in the production planning process. Maxwell and Muckstadt (26) develop a model that coordinates production decisions, including capacity planning and detailed scheduling, with transportation decisions. Hence, we see that the hierarchical approach provides a link among various levels of decisions in a manufacturing system.

# CHAPTER 3

# FURTHER CONSIDERATIONS ON THE TRADITIONAL ECONOMIC LOT SCHEDULING (ELSP) PROBLEM

## 3.1 When to Induce Idle Time in the ELSP with Zero Setup Costs

### 3.1.1 Introduction

In the ELSP with zero setup costs, intuition suggests that machine capacity should be fully utilized to minimize inventory holding costs. Dobson (9) presented a counterexample in which an optimal solution has idle time. This section addresses the conditions under which it is desirable to induce idle time into solutions with guaranteed feasibility. Analysis shows that these conditions can generally be characterized as one product having dominant holding costs and the other products having low machine utilization. We first develop the two product case and then examine two production patterns of the three product case. These lead to a generalization for multi-products.

### 3.1.2 Rationale for using zero setup costs.

Since the fundamental idea of the ELSP is scheduling multiple products through a single machine, it seems only natural that the dominant feature of the problem is the constrained resource of machine capacity (see, for example, Karmarkar, 21). Therefore, the most important impact of setup arises through the value of lost machine capacity. Hence, in this section setup costs are set to zero and the constraint on machine capacity is handled explicitly.

### 3.1.3 An Aside on the Baker and Bomberger Problems.

The classical Baker and Bomberger problems (1,3) have high setup costs, that is, the setup costs are higher than the value of machine capacity lost in setup time. Hence the theoretical optimum solutions to these problems have substantial idle time (see Appendix 2). Because of this idle time, several authors (7,10,11,13,14,23) have made significant improvements in solving these problems with heuristic algorithms. Clearly, the more idle time you induce by charging high setup costs, the easier it is to construct feasible schedules that approach the theoretical lower bound.

### 3.1.4 Two product case.

Consider two products produced on the same machine where one product (call it product 1) has a higher holding cost. Suppose for a length of time $t_0$ of the cycle T, we produce product 1 exactly to meet demand. That is, idle time is induced after producing each unit of product 1 so that no inventory cost is incurred on product 1 during the interval $(0,t_0)$ (see figure 3-1). Conceptually, you could also view this as slowing down the machine during $t_0$ so that product 1 precisely meets demand.

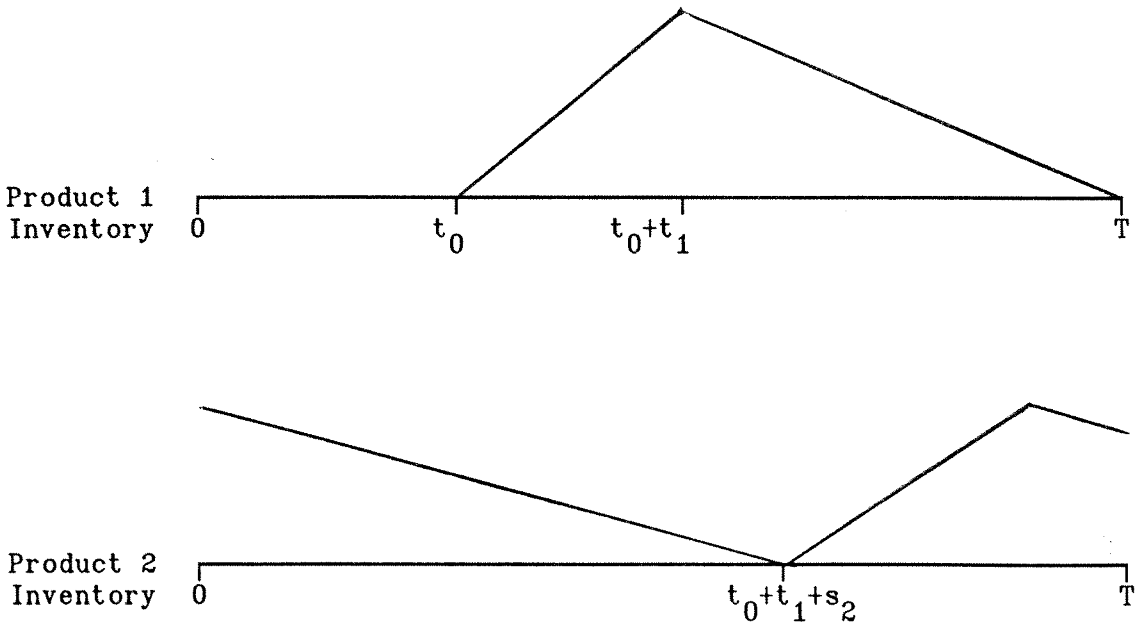



FIGURE 3-1

Two Product Inventory Pattern

The time parameters $t_0$, $t_1$, $t_2$, and T are related as follows:

$$t_2 = \rho_2 T \qquad \{\text{production time for product 2}\}$$

$$t_1 = [\rho_1/(1-\rho_1)](\rho_2 T+s_1+s_2) \qquad \{\text{production time for inventory cycle of product 1}\}$$

$$t_1+s_2+t_2+s_1 = (\rho_2 T+s_1+s_2) \;/\; (1-\rho_1) \qquad \{\text{length of inventory cycle of product 1}\}$$

$$t_0 = T - (t_1+s_2+t_2+s_1)$$

$$= T(1-\rho_1-\rho_2)/(1-\rho_1) - (s_1+s_2)/(1-\rho_1) \;.$$

Since $t_0$ must be non-negative, the cycle length T is constrained by:

$$\text{(1)} \qquad T \geq (s_1+s_2)/(1-\rho_1-\rho_2) \;.$$

This leads to the overall cost expression we want to minimize:

Average inventory cost $= H_1(t_1+s_2+t_2+s_1)^2/T + H_2T$ or

$$(2)\quad C(T) = H_1\left\{\rho_2{}^2T+2\rho_2(s_1+s_2)+(s_1+s_2)^2/T\right\}/(1-\rho_1)^2 + H_2T\ .$$

To find the minimum of (2) with respect to T,

$$(3)\quad dC/dT = H_1\rho_2{}^2/(1-\rho_1)^2 - H_1(s_1+s_2)^2/(1-\rho_1)^2T^2 + H_2\ .$$

Observe that $d^2C/dT^2 > 0$, hence we can set equation (3) equal to 0 and solve for $T^*$ to find the minimum of C(T).

$$T^* = (s_1+s_2) \;/\; \left\{(1-\rho_1)^2H_2/H_1 + \rho_2{}^2\right\}^{1/2}$$

Then constraint (1) is not binding and $t_0 > 0$ in the optimal solution of (2) only if

$$(s_1+s_2) \;/\; \left\{(1-\rho_1)^2H_2/H_1 + \rho_2{}^2\right\}^{1/2} > (s_1+s_2) \;/\; (1-\rho_1-\rho_2)\ .$$

This gives the condition for inducing idle time

$$H_2/H_1 < (1-\rho_1-2\rho_2)/(1-\rho_1)\ .$$

For example, if $\rho_1 = 0.6$ and $\rho_2 = 0.15$, induce idle time only if $H_2/H_1 < 1/4$. Observe that since the right hand side of this inequality is less than or equal to 1, a necessary condition for inducing idle time for product 1 is

$$H_2 < H_1\ .$$

Observe also that since the left hand side of this inequality is always greater than or equal to 0, another <u>necessary condition</u> is

$$\rho_2 < (1-\rho_1)/2 .$$

Hence, the general condition for inducing idle time in the two product case is when the product with lower holding costs also has low machine utilization.

Observe that the inventory costs using induced idle time may be substantially lower than the theoretical lower bound calculated under the assumption of full saw-tooth production inventory patterns. Consider the two product case with $H_2 < H_1$, $\rho_2 < (1-\rho_1)/2$, and $H_2/H_1 < (1-\rho_1-2\rho_2)/(1-\rho_1)$. Let $H_2 = H/k$ , $H_1 = kH$ , therefore, $H_2/H_1 = 1/k^2$ where $k > 1$ , $s_1 = s_2 = s$ , $\rho_1 = 0.5$ , and $\rho_2 < 0.25$. For this range of parameters, the lower bound from the independent solution is

$$\text{LLB} = \frac{\left\{(Hs/k)^{1/2} + (kHs)^{1/2}\right\}}{(1-.5-\rho_2)}$$

$$= \frac{2Hs(k+1)^2}{k(1-\rho_2)} .$$

For the model with induced idle time,

$$T^* = 2s \Big/ \left\{(1-.5)^2/k^2 + \rho_2{}^2\right\}^{1/2}$$

$$= 4ks \Big/ \left\{1+4k^2\rho_2{}^2\right\}^{1/2}$$

and

$$C(T^*) = 8Hs\left\{(1+4k^2\rho_2{}^2)^{1/2} + 2k\rho_2\right\} .$$

Then we can compare C(T*) to LLB.

$$\frac{C(T^*)}{LLB} = \frac{4k(1-\rho_2)}{(k+1)^2}\left\{(1+4k^2\rho_2{}^2)^{1/2} + 2k\rho_2\right\}$$

If we take the case when $\rho_2$ is small, we get

$$\lim_{\rho_2 \to 0} [C(T^*)/LLB] = 4k / (k+1)^2 .$$

Hence in this case, when $k > 1$ (that is, $H_2 < H_1$), then the idle time solution is less than the independent solution lower bound. If we take the case when k is large (that is, $H_2 << H_1$), we get

$$\lim_{k \to \infty} [C(T^*)/LLB] = 16\rho_2(1-\rho_2) .$$

Hence in this case, when $\rho_2 < .07$, the idle time solution is less than the independent solution lower bound. If we take the example where $\rho_2 = .05$ and $k = 5$, we get $C(T^*)/LLB = .85$, that is, the average cost per year for the idle time solution is less than the corresponding cost of the independent solution lower bound.

In summary, we see that in the two product case, for a restricted range of the parameters, we can get better solutions by inserting idle time into the schedule. The restriction on the range of the parameters can be expressed as one dominant product having higher holding costs and the other product having low machine utilization.

### 3.1.5 Three product case.

We have seen for the two product case conditions under which it is desirable to induce idle time into the schedule. Now consider three products produced on the same machine where one product (product 1) has higher holding costs. There are an infinite number of possible production patterns we could consider. However, two examples of production configurations will illustrate that the conditions for inducing idle time in the three product case are similar to the two product case.

#### 3.1.5.1 Production Configuration 1.

For the first example of the three product case, we consider a production pattern with two setups per cycle for the dominant product. That is, we build up inventory of product 1 prior to the production of each of the other products (see Figure 3-2).

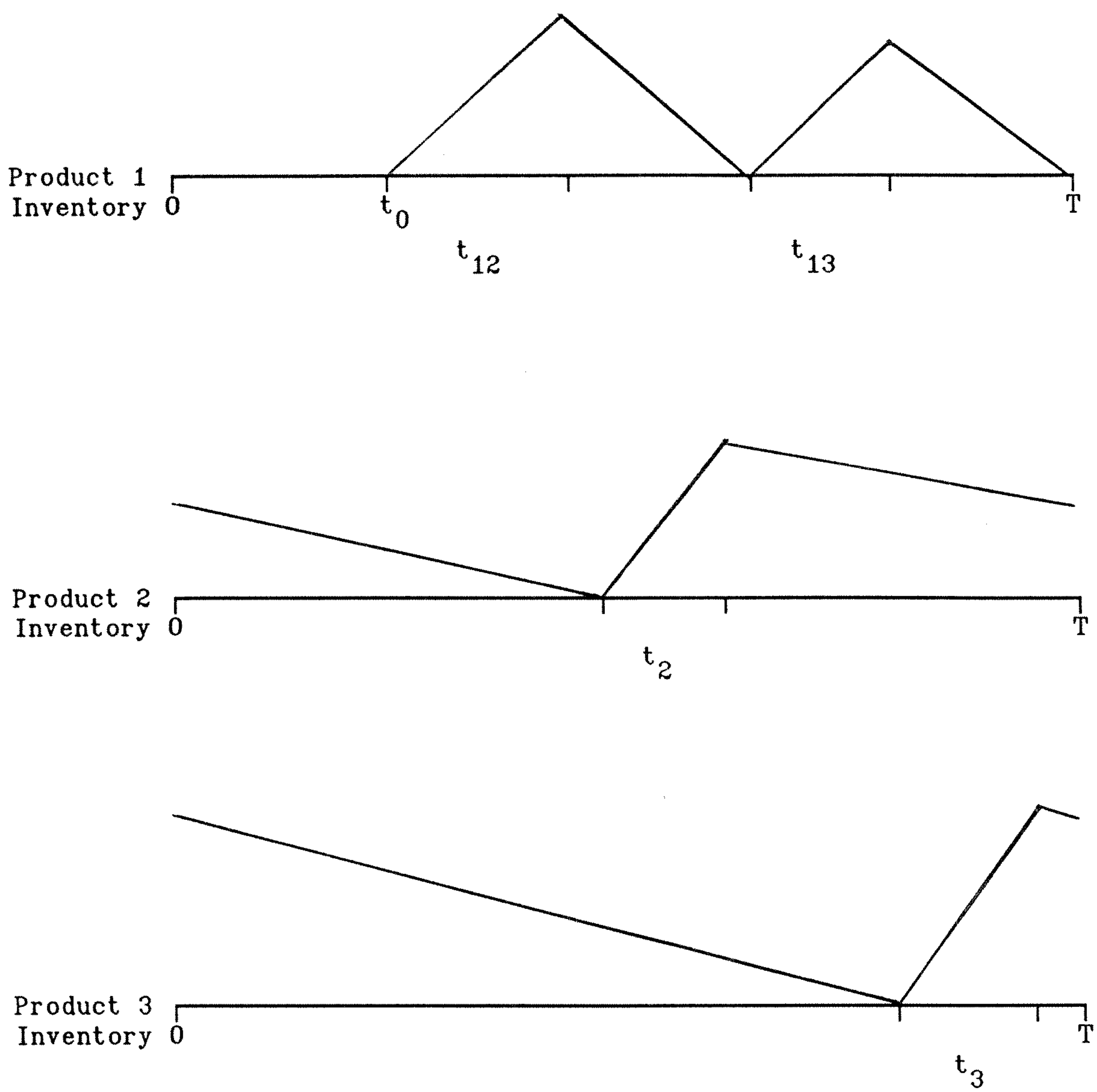



FIGURE 3-2

Three Product Inventory Pattern

The time parameters $t_0$, $t_{12}$, $t_{13}$, $t_2$, $t_3$, and T are related as follows:

$$t_2 = \rho_2 T$$

$$t_3 = \rho_3 T$$

$$t_{12} = \rho_1(\rho_2 T+s_1+s_2) \;/\; (1-\rho_1)$$

$$t_{12}+s_2+t_2+s_1 = (\rho_2 T+s_1+s_2) \;/\; (1-\rho_1)$$

$$t_{13} = \rho_1(\rho_3 T+s_1+s_3) \;/\; (1-\rho_1)$$

$$t_{13}+s_3+t_3+s_1 = (\rho_3 T+s_1+s_3) \;/\; (1-\rho_1)$$

$$t_0 = T(1-\rho_1-\rho_2-\rho_3)/(1-\rho_1) - (2s_1+s_2+s_3)/(1-\rho_1) \;.$$

Since $t_0$ must be non-negative, the cycle length T is constrained by

$$\text{(4)} \qquad T \geq (2s_1+s_2+s_3) \;/\; (1-\rho_1-\rho_2-\rho_3) \;.$$

<u>Average inventory cost.</u>

$$C(T) = \frac{H_1}{(1-\rho_1)^2}\left\{T(\rho_2^2+\rho_3^2)+2\rho_2(s_1+s_2)+2\rho_3(s_1+s_3) + \frac{(s\;+s_2)^2+(s_1+s_3)^2}{T}\right\}$$

$$+ \; H_2 T \; + \; H_3 T$$

To find the minimum of C with respect to T,

$$\text{(5)} \quad dC/dT = H_1\left\{\rho_2{}^2+\rho_3{}^2-(s_1+s_2)^2/T^2-(s_1+s_3)^2/T^2\right\} + H_2 + H_3 \;.$$

Observe that $d^2 C/dT^2 \geq 0$. Hence we can set equation (5) equal to 0 and solve for $T^*$ to find the minimum of C(T).

$$T^* = \frac{\left\{(s_1+s_2)^2+(s_1+s_3)^2\right\}^{1/2}}{\left\{(H_2+H_3)(1-\rho_1)^2/H_1 + \rho_2{}^2+\rho_3{}^2\right\}^{1/2}}$$

Condition for inducing idle time.

Then constraint (4) is not binding and $t_0 > 0$ in the optimal solution only if

$$(6) \qquad \frac{\left\{(s_1+s_2)^2+(s_1+s_3)^2\right\}^{1/2}}{\left\{(H_2+H_3)(1-\rho_1)^2/H_1 +\rho_2{}^2+\rho_3{}^2\right\}^{1/2}} > \frac{2s_1+s_2+s_3}{1 - \rho_1 - \rho_2 - \rho_3} .$$

Observe that $(s_1+s_2) + (s_1+s_3) \geq \left\{(s_1+s_2)^2 + (s_1+s_3)^2\right\}^{1/2}$.

Furthermore, note that

$$\left\{(H_2+H_3)(1-\rho_1)^2/H_1 + \rho_2{}^2+\rho_3{}^2\right\}^{1/2} < 1-\rho_1-\rho_2-\rho_3$$

only if $(H_2+H_3)/H_1 < 1$. Hence a <u>necessary condition for inducing idle time</u> is

$$(7) \qquad H_2+H_3 < H_1 .$$

Since $(H_2+H_3)/H_1 \geq 0$, another <u>necessary condition</u> is

$$\frac{\left\{\rho_2{}^2+\rho_3{}^2\right\}^{1/2}}{1-\rho_1-\rho_2-\rho_3} < \frac{\left\{(s_1+s_2)^2 + (s_1+s_3)^2\right\}^{1/2}}{(s_1+s_2)+(s_1+s_3)} .$$

If we take the simple case where $\rho_2 = \rho_3$ and $s_2 = s_3$, this reduces to

$$\rho_2+\rho_3 < (1-\rho_1)/2 \quad .$$

Note that this is similar to the two product case. The general condition for inducing idle time is when the products with lower holding costs also have low machine utilization.

3.1.5.2 Dobson's Counterexample

Dobson presented the following counterexample to show that under certain conditons, it was optimal to have a schedule with idle time (4, page 18):

$h_1 = 1$, $h_2 = h_3 = 0$, $r_1 = r_2 = r_3 = 1$, $p_1 = 4$, $p_2 = p_3 = M$, $s_1 = s_3 = 1$, $s_2 = M$, $\rho_1 = 1/4$, $\rho_2 = \rho_3 = 1/M$, $H_1 = 3/8$, and $H_2 = H_3 = 0$.

These parameters clearly satisfy the necessary condition (7), that is, $0 + 0 < 1$. Idle time should be induced if constraint (6) is satisfied. This holds for values of M as small as 5. Hence, we don't need to use extreme pathological cases to show the advantages of inserting idle time.

3.1.5.3 Production Configuration 2

For the next example of the three product case, we consider a production pattern where the dominant product is only set up once per cycle. That is, we build up enough inventory in that one production run to carry through the production time of the remaining products (see figure 3-3).

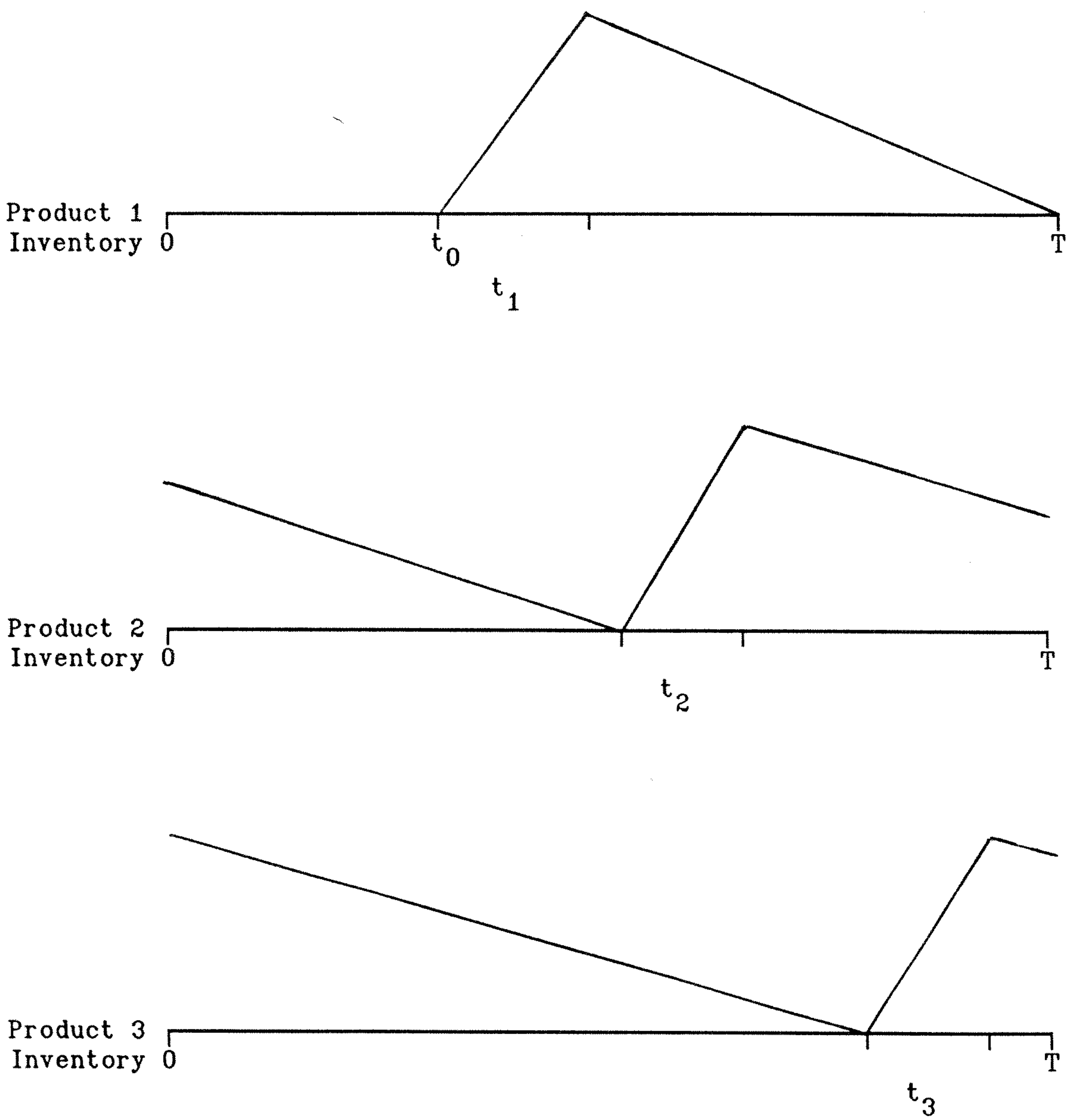



FIGURE 3-3

Three Product Inventory Pattern

The time parameters $t_0$, $t_1$, $t_2$, $t_3$, and T are related as follows:

$$t_2 = \rho_2 T$$

$$t_3 = \rho_3 T$$

$$t_1 = \rho_1[(\rho_2+\rho_3)T+(s_1+s_2+s_3)] \;/\; (1-\rho_1)$$

$$t_1+s_2+t_2+s_3+t_3+s_1 = [(\rho_2+\rho_3)T+(s_1+s_2+s_3)] \;/\; (1-\rho_1)$$

$$t_0 = T(1-\rho_1-\rho_2-\rho_3)/(1-\rho_1) - (s_1+s_2+s_3)/(1-\rho_1) \,.$$

Since $t_0$ must be non-negative, the cycle length T is constrained by

$$T \geq (s_1+s_2+s_3) \;/\; (1-\rho_1-\rho_2-\rho_3) \,. \tag{8}$$

<u>Average inventory cost.</u>

$$C(T) = H_1\left\{T(\rho_2+\rho_3)^2+2(\rho_2+\rho_3)+(s_1+s_2+s_3)^2/T\right\} \;/\; (1-\rho_1)^2 + H_2 T + H_3 T$$

To find the minimum of C with respect to T,

$$dC/dT = H_1\left\{(\rho_2+\rho_3)^2-(s_1+s_2+s_3)^2/T^2\right\} \;/\; (1-\rho_1)^2 + H_2 + H_3 \,.$$

Observe that $d^2\, C/d\, T^2 \geq 0$, hence we can solve for $T^*$ to find the minimum of C(T).

$$T^* = \frac{(s_1+s_2+s_3)}{\left\{(H_2+H_3)(1-\rho_1)^2/H_1 + (\rho_2+\rho_3)^2\right\}^{1/2}}$$

<u>Condition for inducing idle time.</u>

Constraint (8) is not binding and $t_0 > 0$ in the optimal solution only if

$$(H_2+H_3)/H_1 \quad < \quad 1 - 2(\rho_2+\rho_3)/(1-\rho_1) \ .$$

Observe that since the right hand side of this constraint is less than or equal to 1, a <u>necessary condition for inducing idle time</u> is

$$H_2+H_3 \quad < \quad H_1 \ .$$

Since $(H_2+H_3)/H_1 \geq 0$, another <u>necessary condition</u> is

$$\rho_2+\rho_3 \quad < \quad (1-\rho_1)/2 \ .$$

Note that this is the same as in the two product case. Hence, we see that for general production configurations of the three product case, the two necessary conditions for inserting idle time are that one product has dominant holding costs and that the remaining products have low machine utilization.

<u>3.1.6 General criteria for inducing idle time.</u>

Using the results derived in the two product case and the two production configurations considered in the three product case, we can formulate the following approach to determine if idle time should be inserted into the schedule for a particular problem:

1. One product (product 1) has dominant holding costs.

$$H_1 \quad > \quad \sum_{i > 1} H_i \ .$$

2. Remaining products have low machine utilization.

$$\sum_{i > 1} \rho_i < (1-\rho_1)/2 .$$

If a problem meets these necessary conditions, use the following procedure to determine if idle time should be induced:

3. Formulate a 'good' production pattern (for example, see reference 6 for a heuristic technique to develop production sequences).

4. Define the relationships between time parameters $t_i$ and T (similar to those defined in the two product and three product cases).

5. Formulate the average inventory cost as a function of T subject to the constraint that $t_0$ is non-negative.

6. Solve for $T^*$ to minimize C(T).

7. If the parameters of the problem make $t_0 > 0$, then idle time should be induced.

## 3.2 When the Zero Switch Rule (ZSR) is Not Optimal

### 3.2.1 Introduction.

In the Economic Lot Scheduling Problem (ELSP), the ZSR has generally been regarded as a good policy for keeping average inventory levels low (Maxwell, 24). This is supported by the fact that ZSR is optimal with respect to inventory of a single product as will be verified in the following section. Pathological cases have been developed to prove the non-optimality of the ZSR (Delporte, 6). This section addresses specific conditions under which the ZSR is non-optimal. These conditions generally occur when the optimal solution with respect to the ZSR gives uneven production patterns to high cost products.

### 3.2.2 General Approach.

The general approach we use to improve the ZSR solution is to look at marginal adjustments to the current production pattern where we can decrease the overall inventory costs.

#### 3.2.2.1 Cost Savings from Balancing Production.

Suppose we have a solution to the ELSP that is optimal with respect to the ZSR policy. Consider two adjacent production runs of one of the higher cost products such that adjacent runs are not balanced.

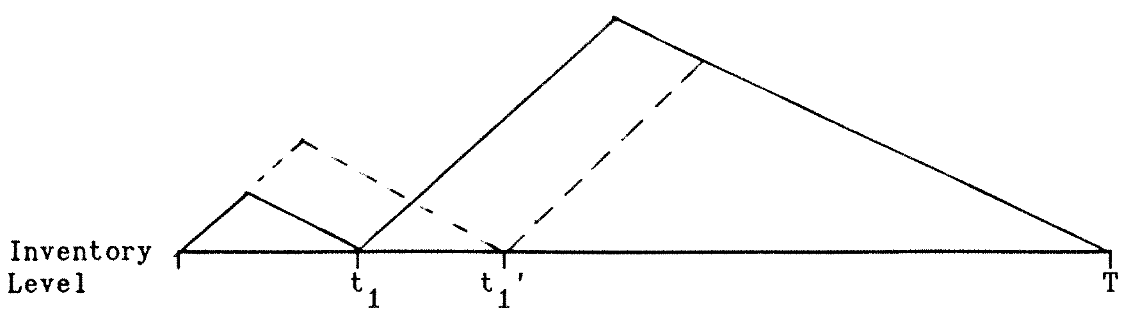



FIGURE 3-4

One Product Inventory Pattern

If we slightly adjust these production cycles, retaining the same overall length T for both cycles, we can show the marginal savings in average inventory costs. Consider the inventory of a given product under the ZSR for two non-identical production runs.

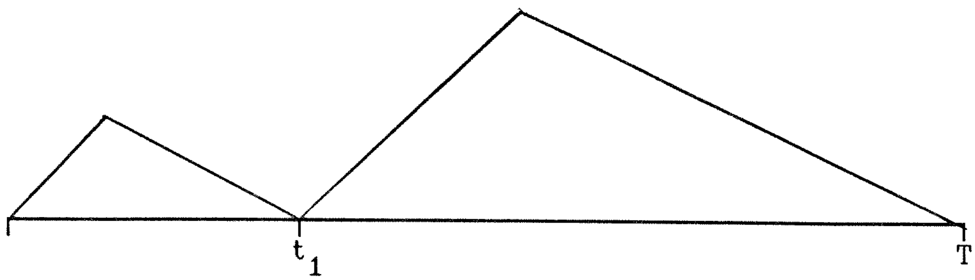



FIGURE 3-5

One Product Inventory Pattern

$$\text{Average inventory} = r(1-\rho)[t_1^2+(T-t_1)^2] \;/\; 2T$$

$$= r(1-\rho)[2t_1^2/T + T - 2t_1] \;/\; 2 \;.$$

The minimum average inventory with respect to $t_1$ is achieved when $t_1$ = T/2, that is, the production lots are balanced. Then the cost function can be written as

$$C \;=\; hr(1-\rho)[2t_1^2/T + T - 2t_1] \;/\; 2$$

or

$$C(t_1) \;=\; H[2t_1^2/T + T - 2t_1] \;.$$

The rate of change of C with respect to $t_1$ is

$$dC/dt_1 \;=\; 4H[t_1 - T/2] \;/\; T \;.$$

Observe that if $t_1 < T/2$ , increasing $t_1$ decreases the average inventory and decreasing $t_1$ increases the average inventory. Conversely, if $t_1 > T/2$ , decreasing $t_1$ decreases the average inventory and increasing $t_1$ increases the average inventory. Hence, balancing the production cycles decreases the average inventory.

3.2.2.2 Cost of Using a Non-zero Switch.

Consider two adjacent production runs of one of the products with lower holding costs. Suppose we again alter the cycle to accommodate a non-zero switch while retaining the same overall length T for both cycles.

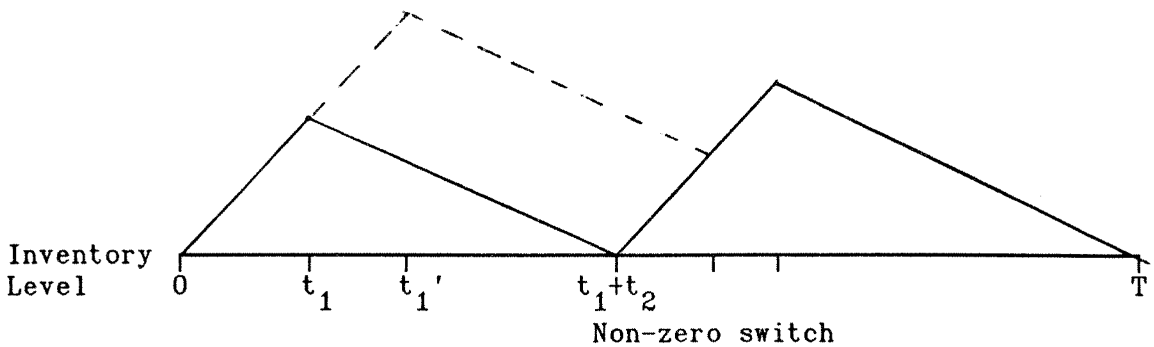



FIGURE 3-6

One Product Inventory Pattern

If we fix T, increasing $t_1$ decreases the length of the second production run. We can show the marginal increase in average inventory costs. Suppose inventory for a given product has the following pattern:

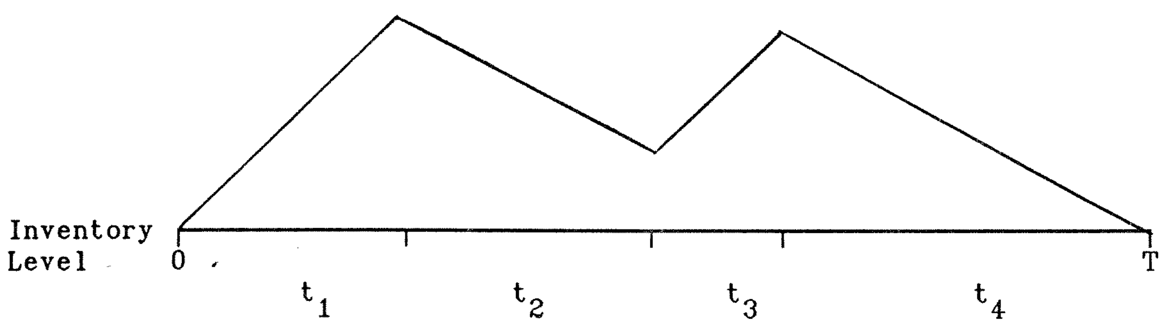



FIGURE 3-7

One Product Inventory Pattern

The time parameters $t_1, t_2, t_3, t_4$, and T are related as follows:

$$t_1 + t_3 = \rho T \qquad \text{or} \qquad t_3 = \rho T - t_1, \text{ and}$$

$$t_2 + t_4 = (1-\rho)T \qquad \text{or} \qquad t_4 = (1-\rho)T - t_2 \; .$$

$$\text{Average inventory} = \Big\{ (p-r)t_1^2/2 + [(p-r)t_1 + (p-r)t_1 - rt_2]t_2/2$$

$$+ [(p-r)t_1 - rt_2 + rt_4]t_3/2 \; + \; rt_4^2 \Big\} / T$$

$$= pt_1t_2/T \; - \; rt_2 \; + \; r(1-\rho)T/2 \; .$$

Observe that $t_1 \leq \rho T$. Hence the coefficient of $t_2$ is always less than or equal to 0. Because the inventory at the non-zero switch point must be non-negative, we get the following constraint:

$$(p-r)t_1 \; - \; rt_2 \; \geq \; 0 \; .$$

This constrains $t_2$ to

$$0 \leq t_2 \; \leq t_1(p-r)/r \; .$$

Hence, average inventory of this product would be minimized with respect to $t_2$ by $t_2 = t_1(p-r)/r$. That is, average inventory of that particular product is minimized by following the zero switch rule. Note also that average inventory of this product is then minimized with respect to $t_2$ (under the ZSR) when

$$t_1 \; = \; \rho T/2 \; .$$

That is, average inventory is minimized when the production lots are exactly balanced. Then the cost function can be written as

$$C = h\{pt_1t_2/T - rt_2 + rT(1-\rho)/2\} .$$

The rate of change of C with respect to $t_1$ is

$$dC/dt_1 = hpt_2/T .$$

3.2.2.3 Improving the ZSR Solution.

We can improve the ZSR solution in the following way. Take two products that have two adjacent production runs in the ZSR solution. Adjust their production times retaining ZSR for the high cost product (product i) and incorporating a non-zero switch for the lower cost product (product j), as shown in Figures 3-4 and 3-6, such that we don't affect the rest of the production cycle. We then make the production pattern of the higher cost product more balanced at the expense of using a non-zero switch policy for the lower cost product.

Then we should use the non-zero switch option if the marginal cost savings from making production of product i more balanced exceeds the marginal cost of using a non-zero switch on product j . That is

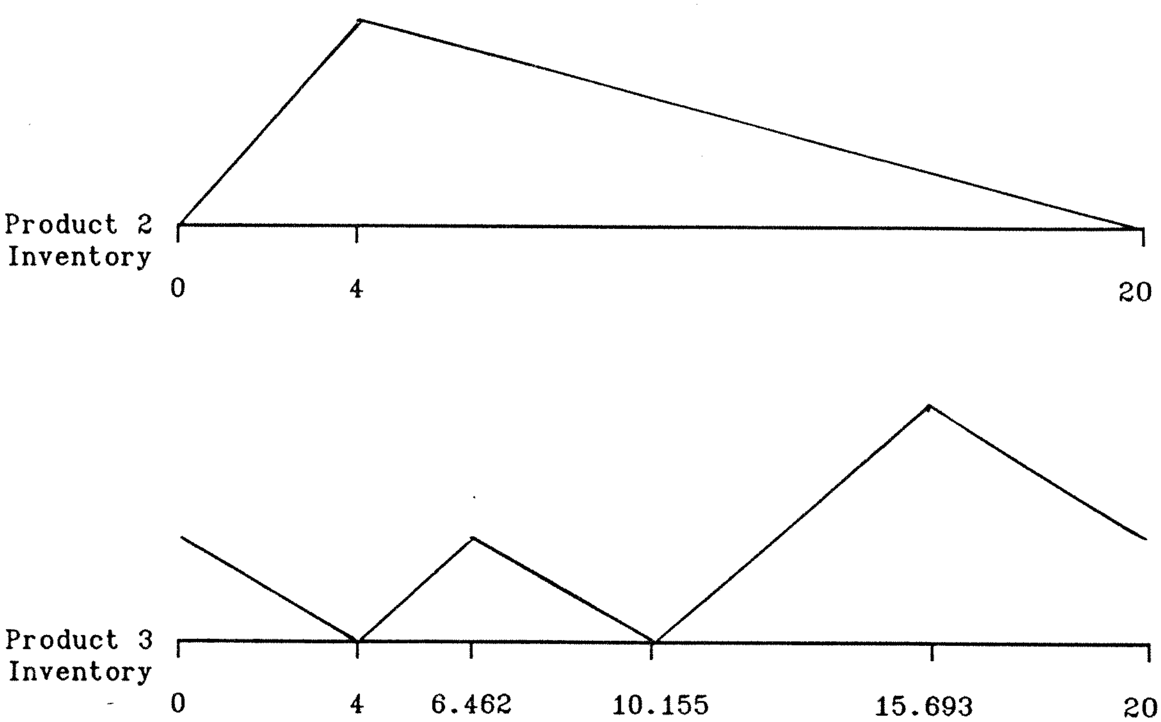
$$2H_i(T_i-2t_{i1})/T_i > h_jp_jt_{j2}/T_j$$

or (9) $$\frac{h_i}{h_j} \quad \frac{2r_i(1-\rho_i)}{p_j} > \frac{T_i}{(T_i/2-t_{i1})} \quad \frac{t_{j2}}{T_j} .$$

Thus we see that we can improve a ZSR solution when that solution gives a lumpy production pattern to a product with dominant holding costs. By using a non-zero switch on some cheaper product, we can balance production of the dominant product.

3.2.2.4 Delporte's Counterexample.

Delporte presented a counterexample to the optimality of the ZSR (6, App IV, A Counterexample to the Optimality of the ZSR). The data for this problem is as follows: $p_2 = 1$, $r_2 = .2$, $\rho_2 = .2$, $p_3 = 1$, $r_3 = .4$, $\rho_3 = .4$, $h_3 = .1$, $r_1 = .4$, $\rho_1 = .4$, and $h_1 = 100$. The solution to this problem with respect to the ZSR is given by Figure 3-8.



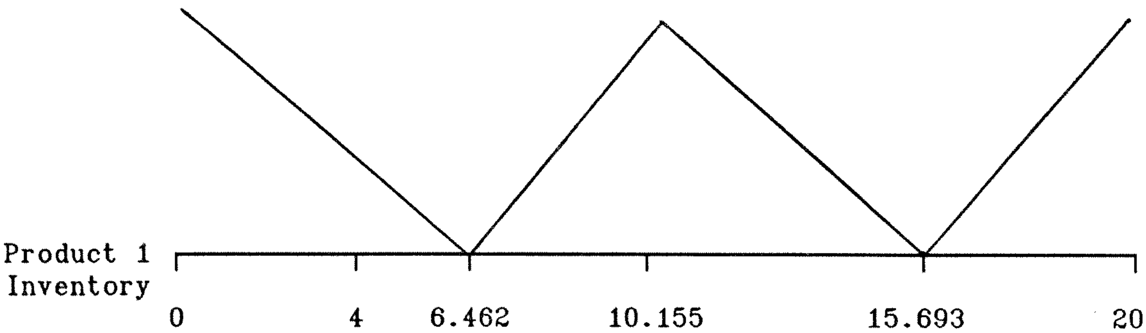



FIGURE 3-8

Three Product Inventory Pattern

Substituting this data into equation (9), we get 480 > 1.3. In this example, we see that the ZSR solution is not optimal even if we take less extreme values of relative holding costs, that is, as long as $h_1/h_3 > 370$ , the ZSR solution is not optimal.

3.2.3 Conditions When the ZSR is not Optimal

Under what conditions is it likely that the ZSR is not optimal? We can best answer that by asking when the inequality (9) is likely to hold. Consider two products whose independent solutions for natural cycle length are approximately equal, that is, using the Lagrangian relaxation method, we obtain

$$T_i^* = \left\{ \sum [H_i s_i]^{1/2} \right\} [s_i/H_i]^{1/2} \Big/ \left\{ 1-\sum \rho_i \right\} .$$

Hence, these two products should be produced at roughly the same relative frequency. Then $T_i^* = T_j^*$ or $s_i/H_i = s_j/H_j$. If we let $T_i = T_j$ and substitute into equation (9), we get

$$(s_i/s_j)\ \rho_i(1-\rho_i) \ > \ t_{j2}/(T/2 - t_{i1}) .$$

Then under this situation, the ZSR solution is not optimal if

(a) Two products have roughly the same natural cycle length while one of the products has both dominant holding costs and setup time.

and

(b) The ZSR solution gives uneven production patterns to the higher cost product.

In summary, conditions (a) and (b) above describe when the ZSR is likely to not be optimal and we may be able to improve the solution by incorporating a non-zero switch. When these conditions do not hold, we can safely conclude that the ZSR is a good policy. For example, if we use a power-of-two policy (Roundy, 29), all production lots for a given product are equal so we could not improve the solution by using a non-zero switch.

# CHAPTER 4

# STOCHASTIC INPUT TO A BOTTLENECK MACHINE

## 4.1 Introduction

Consider a bottleneck machine which processes several products (see Figure 4-1). The traditional ELSP approach is to assign costs for setup of each product and costs for holding inventory after processing until the parts are demanded or consumed. It ignores the issue of delivery of raw parts to the bottleneck machine and resulting holding costs for those raw parts. If delivery of the raw parts is deterministic, the delivery of each product could be scheduled to coincide exactly with its start time. In this deterministic case, it is reasonable to ignore the delivery issue when scheduling the bottleneck machine. However, if the deliveries are not deterministic, that is if there is variability in shipping time or processing time on the predecessor machines, then ignoring this issue when scheduling the bottleneck machine can seriously affect total operating costs. For example, if delivery of the raw parts is requested too early, then excessive work-in-process (WIP) inventory will accumulate before the bottleneck machine. On the other hand, if delivery of the raw parts is requested too late, then productive capacity of the entire system is diminished because the bottleneck machine is delayed.

This chapter addresses the combined issues of scheduling delivery of raw parts to a bottleneck machine and scheduling lot sizes on that machine. Section 4.2 focuses on the delivery issue assuming the production schedule has already been defined for the bottleneck machine. Section 4.3 analyzes how state information can be used to schedule the machines using real time data. Section 4.4 develops an aggregate model for combining the issues of lot sizing and delivery scheduling. Section 4.5 relaxes the constraint that all demand be satisfied in each period and looks at a dynamic programming approach. Section 4.6 looks at the special case of an assembly model with a different inventory pattern.

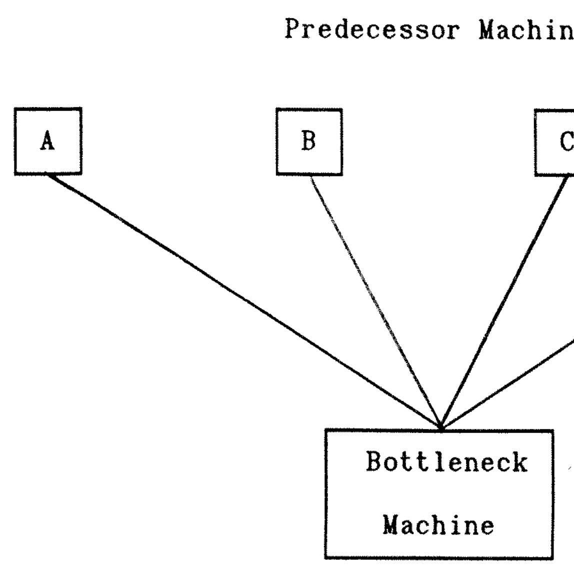



FIGURE 4 - 1

Machine Network

## 4.2 Scheduling Deliveries to a Bottleneck Machine

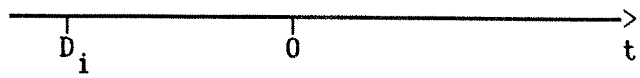
### 4.2.1 Simple Newsboy Model

Consider one particular production lot on the bottleneck machine and focus only on the delivery issue. Suppose the machine will be available and production of this lot is scheduled to start at a specific time (reference this as time zero). Define the following additional parameters:

| | |
|---|---|
| $d_i$ | requested delivery time of product i raw parts , |
| $D_i$ | random variable representing actual delivery time (note: $D_i > 0$ means the lot arrived late, $D_i < 0$ means the lot arrived early) , |
| $f_i(\cdot)$ | probability density of delivery time of product i centered about mean zero (assumed independent of lot size -- this assumption may not be valid in certain cases, Karmarkar, 22) , |
| $F_i(\cdot)$ | cumulative distribution of $f_i(\cdot)$ , |
| $\bar{h}_i$ | holding cost at the bottleneck machine of raw part i per unit time , |
| $r_i T_i$ | lot size for product i , and |
| $\lambda$ | value of lost machine capacity per unit time . |

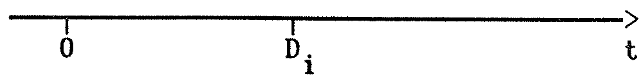
Given that $d_i$ is the requested delivery time, there are two possible outcomes and costs incurred.

Case 1. Early delivery ($D_i < 0$)




The cost incurred is $\overline{h}_i r_i T_i(-D_i)$ which is the inventory holding cost of raw parts.

Case 2. Late delivery ($D_i > 0$)




The cost incurred is $\lambda\, D_i$ which is the value of lost machine capacity. The expected cost, given that $d_i$ is the requested delivery time, can be formulated as

$$C(d_i) \;=\; \overline{h}_i r_i T_i \int_{-\infty}^{0} -x\, f_i(x - d_i)\, dx \;+\; \lambda \int_{0}^{\infty} x\, f_i(x - d_i)\, dx\ .$$

Using a change of variable, $y = x - d_i$, we simplify this expression to

$$C(d_i) \;=\; -\,\overline{h}_i r_i T_i \int_{-\infty}^{-d_i} (y + d_i)\, f_i(y)\, dy \;+\; \lambda \int_{-d_i}^{\infty} (y + d_i)\, f_i(y)\, dy$$

$$= \lambda\, d_i - (\lambda + \overline{h}_i r_i T_i) \int_{-\infty}^{-d_i} y\, f_i(y)\, dy - (\lambda + \overline{h}_i r_i T_i)\, d_i \int_{-\infty}^{-d_i} f_i(y)\, dy\ .$$

To find the optimal requested delivery time, determine

$$\frac{d\,C}{d\,d_i} = \lambda \;-\; (\lambda + \overline{h}_i r_i T_i) \int_{-\infty}^{-d_i} f(y)\, dy\ .$$

Since $\dfrac{d^2 C}{d\, d_i^{\,2}} = (\lambda + \bar{h}_i r_i T_i)\, f_i(-d_i) \geq 0$ , C is convex in $d_i$.

Then the optimal requested delivery time is to choose $d_i^*$ such that

$$F_i(-d_i^*) = \frac{\lambda}{\lambda + \bar{h}_i r_i T_i} .$$

This is the familiar solution to the newsboy problem. In this case, we are balancing the expected value of lost machine capacity with the expected value of inventory holding costs.

### 4.2.2 Stochastic Delivery with Two Random Variables

Consider one production lot (product i) scheduled to start at a specific time, reference this as time zero, and another production lot (product j) scheduled to start upon completion of product i (see figure 4-2).

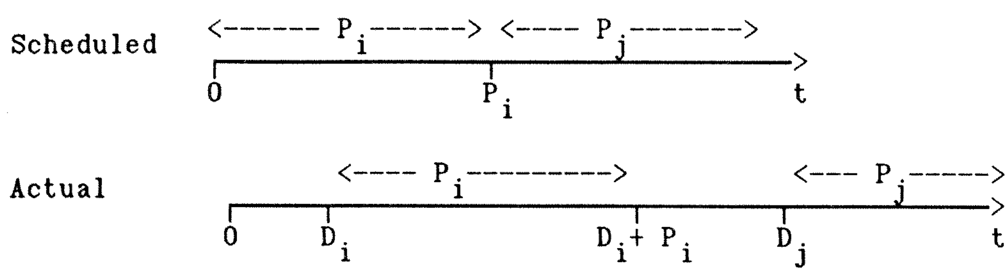



Figure 4-2

Gantt Chart

If $D_i \leq 0$, the production of lot i will start on time, that is, time zero, and if $D_i > 0$, it will start late. Conditioning on $D_i$, there are again two possible outcomes and costs incurred due to product j.

Case 1. Early Delivery, that is, $D_j < P_i + \max\{0, D_i\}$ .

The cost incurred is $-h_j r_j T_j (D_j - P_i - \max\{0, D_i\})$ .

Case 2. Late Delivery, that is, $D_j > P_i + \max\{0, D_i\}$ .

The cost incurred is $\lambda\, (D_j - P_i - \max\{0, D_i\})$ .

The expected cost, given that $d_j$ is the requested delivery time and $f_i$ is the distribution of the start time of production lot i, can be calculated as

$$C_j(d_j) = \int_0^{\infty} f_i(y)\, dy \,[\, -\bar{h}_j r_j T_j \int_{-\infty}^{P_i + y} (x - P_i - y)\, f_j(x - d_j)\, dx$$

$$+ \lambda \int_{P_i + y}^{\infty} (x - P_i - y)\, f_j(x - d_j)\, dx \,] .$$

Using a change of variable, $z = x - d_j$, this simplifies to

$$C_j(d_j) = \int_0^{\infty} f_i(y)\, dy \,[\, -\bar{h}_j r_j T_j \int_{-\infty}^{P_i + y - d_j} (z + d_j - P_i - y)\, f_j(z)\, dz$$

$$+ \lambda \int_{P_i + y - d_j}^{\infty} (z + d_j - P - y)\, f_j(z)\, dz \,] .$$

The optimal requested delivery time $d_j$ is found using the following equation:

$$\frac{d\, C_j}{d\, d_j} = \lambda - (\lambda + \bar{h}_j r_j T_j) \int_0^{\infty} f_i(y\,)\, F_j(P_i + y - d_j)\, dy = 0 .$$

Since

$$\frac{d^2C_j}{d\,d_j^2} = (\lambda + \bar{h}_j r_j T_j) \int_0^\infty f_i(y)\, f_j(P_i + y - d_j)\, dy \geq 0 ,$$

$C_j$ is convex in $d_j$. Then the optimal requested delivery time is to choose $d_j^*$ such that

$$\text{(4-1)} \qquad \int_0^\infty f_i(y)\, F_j(P_i + y - d_j)\, dy = \frac{\lambda}{\lambda + \bar{h}_j r_j T_j} .$$

If $f_i$ represents the arbitrary start time of production lot i, incorporating all random events up to that point, then the same form of solution applies.

### 4.2.3 Three Echelon Machine Network

Suppose the inputs to the predecessors of the bottleneck are stochastic (see figure 4 - 3). The issue is when to schedule deliveries to machine i. If the delivery arrives early, WIP is added to the system at machine i. If the delivery arrives late, the subsequent delivery from i to the bottleneck might be late thereby causing a reduction in capacity of the overall system. Since we know the rate of change of cost at level i with respect to $d_i$, this is the marginal rate at which the overall system costs would be affected by a delay in delivery to i.

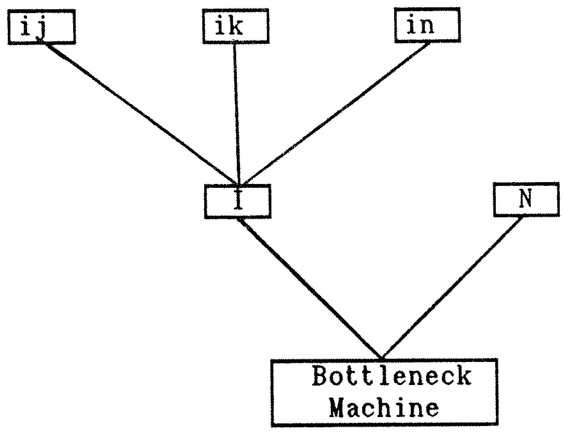



FIGURE 4 - 3

Machine Network

The overall expected system cost with respect to delivery from machine ij can be expressed as

$$
\begin{aligned}
C_{ij}(d_{ij}) &= -\bar{h}_{ij} r_{ij} T_{ij} \int_{-\infty}^{0} x\, f_{ij}(x - d_{ij})\, dx \\
&\quad + [\lambda - (\lambda + \bar{h}_i r_i T_i) F_i(-d_i)] \int_{0}^{\infty} x\, f_{ij}(x - d_{ij})\, dx \\
&= [\lambda - (\lambda + \bar{h}_i r_i T_i)\, F_i(-d_i)]\, d_{ij} \\
&\quad - [\lambda + \bar{h}_{ij} r_{ij} T_{ij} - (\lambda + \bar{h}_i r_i T_i)\, F_i(-d_i)] \int_{-\infty}^{-d_{ij}} y\, f_{ij}(y)\, dy \\
&\quad - [\lambda + \bar{h}_{ij} r_{ij} T_{ij} + (\lambda + \bar{h}_i r_i T_i)\, F_i(-d_i)] \int_{-\infty}^{-d_{ij}} f_{ij}(y)\, dy\ .
\end{aligned}
$$

This yields the optimal delivery time $d_{ij}^{*}$ that satisfies

$$
F_{ij}(-d_{ij}^{*}) = \frac{\lambda - (\lambda + \bar{h}_i r_i T_i)\, F_i(-d_i)}{\lambda - (\lambda + \bar{h}_i r_i T_i)\, F_i(-d_i) + \bar{h}_{ij} r_{ij} T_{ij}}\ .
$$

In summary, delivery times should be chosen to balance the expected costs of holding inventory with the expected cost of lost machine capacity. Having completed this analysis for computing deliveries for general points in time, we now develop a method that looks at the current state of the machine network for real time scheduling of deliveries.

## 4.3 Using State Information to Schedule Pre-Bottleneck Machines

### 4.3.1 General Formulation

Using the concepts developed in Section 4.2, we can then use current information on the state of the system to schedule production in the predecessors to the bottleneck machine. Suppose a given machine i has finished production of a lot and sent it to the bottleneck machine. When should we initiate a new production lot of i knowing current state information, i.e. the status of the bottleneck machine and other predecessors to the bottleneck? In general, the expected delivery time of the raw parts should match the expected availability of the bottleneck machine plus an allowance for some safety time. The amount of safety time depends on the relative value of machine capacity to the cost of holding parts.

Let $g_i(x \mid \text{state})$ be the conditional distribution of the time the bottleneck machine will be available again for product i, given the current state of the system. Then the contribution of the delivery of product i to the system cost can be formulated as

$$C_i(d_i) = \int_0^{\infty} g_i(x \mid \text{state}) \left[-\bar{h}_i r_i T_i \int_{-\infty}^{x} (y-x)\, f_i(y-d_i)\, dy \right.$$

$$\left. + \lambda \int_x^{\infty} (y-x)\, f_i(y-d_i)\, dy \right] dx \,.$$

The optimal solution is to choose $d_i^*$ such that

$$\int_0^{\infty} g_i(x \mid \text{state})\, F_i(x-d_i)\, dx = \frac{\lambda}{\lambda + \bar{h}_i r_i T_i} \,.$$

To shorten future notation, let $R_i = \dfrac{\lambda}{\lambda + \bar{h}_i r_i T_i}$ .

4.3.2 Two Product Case

A two product case illustrates how current state information can be used to schedule pre-bottleneck machines. Consider two machines A and B which directly precede the bottleneck (see figure 4 - 4).

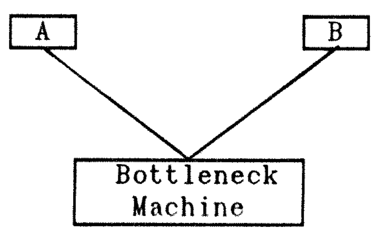



FIGURE 4 - 4

Machine Network

Suppose machine A has finished and sent a production lot to the bottleneck (see figure 4 - 5). When do we initiate production of A again?

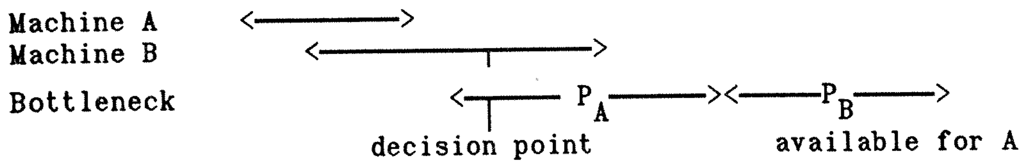



FIGURE 4 - 5

Gantt Chart

We want to know how much work remains on the bottleneck, including the distribution of possible delays as a function of state variables, before the bottleneck will be available for product A again.

The possible states for A and B can be defined as follows:

Production lot A.

State 1. Finished on the bottleneck.
State 2. Started but not finished on the bottleneck.
State 3. Not started on the bottleneck.

Production lot B.

State 1. Started on the bottleneck.
State 2. Awaiting processing on the bottleneck.
State 3. Started but not finished on machine B.
State 4. Not started on machine B.

The state of the system can be expressed as (a,b) where a represents the status of production lot A, and b represents the status of production lot B. Observe that we don't need to consider state (1,2) because B has nothing to wait for in this case and states (2,1) and (3,1) because of incompatibility. For each of the possible states, we can then formulate an expression corresponding to the optimal requested delivery time developed in Section 4.4.1. For each of these states, we will illustrate the results with exponential and uniform distributions of processing times on machines A and B with constant deterministic processing times on the bottleneck machine.

<u>State (1,1). A has finished on the bottleneck, and B has started on the bottleneck.</u>

Let $t_b$ be the time B has already been processed on the bottleneck.

Bottleneck $\longleftarrow\!\!\underset{t_b}{\vert}\!\!\longrightarrow P_B \longrightarrow$ available for A

The optimal solution is to choose $d_a$ such that $F_a(P_B - t_b - d_a) = R_a$ .

<u>Case 1. Exponential processing times for machines A and B.</u>

Assume A and B have exponential processing times on the pre-bottleneck machines, with deterministic processing times on the bottleneck machine, that is, for machines A and B,

$$f_i(x) = \lambda_i \, e^{-\lambda_i x} \quad , \text{ for } x \geq 0 \, ,$$

$$\text{and } F_i(x) = 1 - e^{-\lambda_i x} \, , \text{ for } x \geq 0 \, , \quad i = \text{A and B}.$$

The optimal solution in this case is to choose $d_a$ such that

$$1 - e^{-\lambda_a (P_B - t_b - d_a)} = R_a$$

$$\text{or} \qquad d_a^* = P_B - t_b + \frac{1}{\lambda_a} \log (1 - R_a) \, .$$

Since $0 \le R_a \le 1$, $\log(1 - R_a) \le 0$. Observe that this corresponds to matching the delivery time with the availability of the bottleneck machine (in this case, $P_B - t_b$) plus allowing for allowing for safety time (in this case, $1/\lambda_a \log(1 - R_a)$).

<u>Case 2. Uniform processing times for machines A and B.</u>

For machines A and B,

$$f_i(x) = 1 / 2u_i \text{ , for } 0 \le x \le 2u_i \text{ ,}$$

$$\text{and } F_i(x) = x / 2u_i \text{ , for } 0 \le x \le 2u_i \text{ .}$$

The optimal solution in this case is to choose $d_a$ such that

$$(P_B - t_b - d_a) / 2u_a = R_a$$

$$\text{or} \quad d_a^* = P_B - t_b - 2u_a R_a \text{ .}$$

Observe that this corresponds to matching the delivery time with the availability of the bottleneck machine (in this case, $P_B - t_b$) plus allowing for safety time (in this case, $2u_a R_a$).

<u>State (1,3). A has finished on the bottleneck, and B has started on machine B.</u>

Let $t_B$ be the time B has already been processed on machine B and $f_B(x \mid t_B)$ be the distribution of time remaining on machine B.

Machine B <——┬————>
$t_B$

Bottleneck <———— $P_B$————>
available for A

The optimal solution is to choose $d_a$ such that

$$\int_0^\infty f_B(x \mid t_B)\, F_a(x + P_B - d_a)\, dx \;=\; R_a \,.$$

Case 1. Exponential processing times for machines A and B.

The optimal solution is to choose $d_a$ such that

$$\int_0^\infty \lambda_B e^{-\lambda_B x}\, [1 - e^{-\lambda_a(x + P_B - t_b - d_a)}]\, dx \;=\; R_a$$

$$\text{or} \quad d_a^{*} \;=\; P_B + \frac{1}{\lambda_a} \log[\frac{a\ \ b}{\lambda_b}] \;+\; \frac{1}{\lambda_a} \log\,(1 - R_a) \,.$$

Case 2. Uniform processing times for machines A and B.

Assume that the distribution $f_B(x \mid t_B)$ is uniform $(0,\, 2u_b - 2t_B)$ .

The optimal solution is to choose $d_a$ such that

$$\int_0^{2u_b - 2t_B} \frac{1}{2u_b - 2t_B} [\frac{x + P_B - d_a}{2u_a}] \;=\; R_a$$

$$\text{or} \quad d_a^{*} \;=\; P_B + u_b - t_B - 2u_a R_a \,.$$

State (1,4). A has finished on the bottleneck, and B has not started on machine B.

Let $f_{B1}(x)$ be the distribution of start time on machine B.

```
 Machine B    <— x —><— y —>
Bottleneck                        <———— P_B ————>
                                              available for A
```

The optimal solution is to choose $d_a$ such that

$$\int_0^\infty f_{B1}(x) \int_0^\infty f_B(y)\, F_a(x + y + P_B - d_a)\, dy\, dx \;=\; R_a \,.$$

Case 1. Exponential processing times for machines A and B.

The optimal solution is to choose $d_a$ such that

$$\int_0^\infty \lambda_{B1} e^{-\lambda_{B1} x} \int_0^\infty \lambda_B e^{-\lambda_B y} \left[1 - e^{-\lambda_a (x + y + P_B - t_b - d_a)}\right] dy\, dx = R_a$$

$$\text{or} \quad d_a^* = P_B + \frac{1}{\lambda_a} \log\left[\frac{\lambda_a + \lambda_b}{\lambda_b}\right] + \frac{1}{\lambda_a} \log\left[\frac{\lambda_a + \lambda_{b1}}{\lambda_{b1}}\right] + \frac{1}{\lambda_a} \log(1 - R_a).$$

Case 2. Uniform processing times for machines A and B.

Assume that the distribution $f_B(x \mid t_B)$ is uniform $(0,\ 2u_b - 2t_B)$.

The optimal solution is to choose $d_a$ such that

$$\int_0^{2u_{b1}} \frac{1}{2u_{b1}} \int_0^{2u_b} \frac{1}{2u_b} \left[\frac{x + y + P_B - d_a}{2u_a}\right] dy\, dx = R_a$$

$$\text{or} \quad d_a^* = P_B + u_b + u_{b1} - 2u_a R_a .$$

State (2,2). A has started on the bottleneck, and B is awaiting processing on the bottleneck.

$$\text{Bottleneck} \quad \longleftarrow \underset{t_a}{\vert} \longrightarrow P_A \longrightarrow\longleftarrow P_B \longrightarrow \text{available for A}$$

The optimal solution is to choose $d_a$ such that

$$F_a(P_A - t_a + P_B - d_a) = R_a .$$

Case 1. Exponential processing times on machines A and B.

The optimal solution is to choose $d_a$ such that

$$1 - e^{-\lambda_a (P_A - t_a + P_B - d_a)} = R_a$$

$$\text{or} \qquad d_a^* = P_A - t_a + P_B + \frac{1}{\lambda_a} \log (1 - R_a) .$$

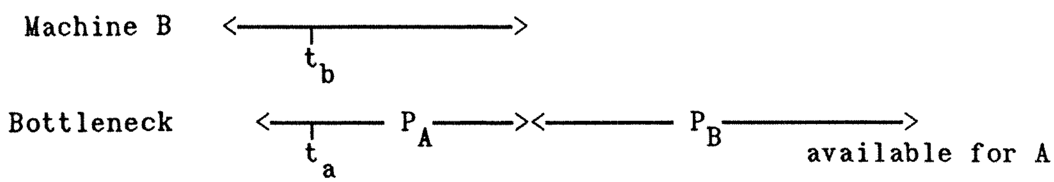
<u>Case 2. Uniform processing times on machines A and B.</u>

The optimal solution is to choose $d_a$ such that

$$(P_A - t_a + P_B - d_a) / 2u_a = R_a$$

$$\text{or} \qquad d_a^* = P_A - t_a + P_B - 2u_a R_a .$$

<u>State (2,3). A has started on the bottleneck, and B has started on machine B.</u>




Then the distribution of the start time of B on the bottleneck is defined by the maximum of the completion time of B on machine B and the completion time of A on the bottleneck.

Let $f_{sb}(x \mid t_a, t_b)$ be the distribution of the start time of B on the bottleneck and

$$f_{sb}(x \mid t_a, t_b) = \begin{cases} F_b(x \mid t_b) & \text{for } x = P_A - t_a \\ f_b(x \mid t_b) & \text{for } x > P_A - t_a . \end{cases}$$

The optimal solution is to choose $d_a$ such that

$$F_b(P_A - t_a \mid t_b)\, F_a(P_A - t_a + P_B - d_a)$$

$$+ \int_{P_A - t_a}^{\infty} f_b(x \mid t_b)\, F_a(x + P_B - d_a)\, dx = R_a .$$

Case 1. Exponential processing times on machines A and B.

In the exponential case, we get

$$f_{sb}(x \mid t_a, t_b) = \begin{cases} 1 - e^{-\lambda_b x} & , \text{ for } x = P_A - t_a . \\ \lambda_b e^{-\lambda_b x} & , \text{ for } x > P_A - t_a . \end{cases}$$

The optimal solution is to choose $d_a$ such that

$$[1 - e^{-\lambda_b (P_A - t_a)}]\,[1 - e^{-\lambda_a (P_A - t_a + P_B - d_a)}]$$

$$+ \int_{P_A - t_a}^{\infty} \lambda_B e^{-\lambda_B x}\,[1 - e^{-\lambda_a (x + P_B - d_a)}]\, dx = R_a$$

or

$$d_a^* = P_A - t_a + P_B - \frac{1}{\lambda_a} \log[1 - \frac{\lambda_a}{\lambda_a + \lambda_b} e^{-\lambda_b (P_A - t_a)}] + \frac{1}{\lambda_a} \log (1 - R_a).$$

Case 2. Uniform processing times on machines A and B.

In the uniform case, we get

$$f_{sb}(x \mid t_a, t_b) = \begin{cases} \dfrac{P_A - t_a}{2u_b - 2t_b} & , \text{ for } x = P_A - t_a . \\ \dfrac{1}{2u_b - 2t_b} & , \text{ for } P_A - t_a < x \le 2u_b - 2t_b . \end{cases}$$

The optimal solution is to choose $d_a$ such that

$$\frac{P_A - t_a}{2u_b - 2t_B} [\frac{P_A - t_a + P_B - d_a}{2u_a}] + \int_{P_A - t_a}^{2u_b - 2t_b} \frac{1}{2u_b - 2t_b} [\frac{x + P_B - d_a}{2u_a}]\, dx = R_a$$

$$\text{or} \qquad d_a^* = P_B + u_b - t_b + \frac{(P_A - t_a)^2}{4(u_b - t_b)} - 2u_a R_a .$$

<u>State (2,4). A has started on the bottleneck, and B has not started on machine B.</u>

Machine B $\longleftarrow x \longrightarrow\longleftarrow y \longrightarrow$

Bottleneck $\longleftarrow \underset{t_a}{\vdash} \; P_A \longrightarrow\longleftarrow P_B \longrightarrow$ available for A

$f_{sb}$ = the distribution of $\max\{x + y,\ P_A - t_a\}$

<u>Case 1. Exponential processing times on machines A and B.</u>

In the exponential case, we get

$$f_{sb}(k \mid t_a) = \begin{cases} 1 - e^{-(\lambda_b+\lambda_{b1})k} & \text{, for } k = P_A - t_a \text{ .} \\ (\lambda_b+\lambda_{b1})\, e^{-(\lambda_b+\lambda_{b1})k} & \text{, for } k > P_A - t_a \text{ .} \end{cases}$$

The optimal solution is to choose $d_a$ such that

$$[1 - e^{-(\lambda_b+\lambda_{b1})(P_A - t_a)}]\,[1 - e^{-\lambda_a(P_A - t_a + P_B - d_a)}]$$

$$+ \int_{P_A - t_a}^{\infty} (\lambda_b+\lambda_{b1})\, e^{-(\lambda_b+\lambda_{b1})x}\,[1 - e^{-\lambda_a(x + P_B - d_a)}]\, dx = R_a$$

$$\text{or} \qquad d_a^* = P_A - P_B - t_a + \frac{1}{\lambda_a}\log(1 - R_a)$$

$$- \frac{1}{\lambda_a}\log\Big[1 - \frac{\lambda_a}{\lambda_a+\lambda_b+\lambda_{b1}}\, e^{-(\lambda_b+\lambda_{b1})(P_A - t_a)}\Big] .$$

Case 2. Uniform processing times on machines A and B.

In the uniform case, we get

$$f_{sb}(k \mid t_a) = \begin{cases} \dfrac{k}{2u_b+2u_{b1}} , & \text{for } k = P_A - t_a . \\ \dfrac{1}{2u_b+2u_{b1}} , & \text{for } P_A - t_a < k \le 2u_b+2u_{b1} . \end{cases}$$

The optimal solution is to choose $d_a$ such that

$$\frac{P_A - t_a}{2u_b+2u_{b1}} \left[\frac{P_B + P_A - t_a - d_a}{2u_a}\right] + \int_{P_A - t_a}^{2u_b + 2u_{b1}} \frac{1}{2u_b + 2u_{b1}} \left[\frac{x + P_B - d_a}{2u_a}\right] dx = R_a$$

$$\text{or} \qquad d_a^* = P_B + \frac{(P_A - t_a)^2}{4u_b+4u_{b1}} + u_b + u_{b1} - 2u_a R_a .$$

State (3,2). A has not started on the bottleneck, and B is awaiting processing on the bottleneck.

Bottleneck <— x —><———— $P_A$ ————><——— $P_B$ ————> available for A

Let $f_{a1}(x)$ be the distribution of the start time of the old lot A on bottleneck. The optimal solution is to choose $d_a$ such that

$$\int_0^\infty f_{a1}(x)\, F_a(x + P_A + P_B - d_a)\, dx = R_a .$$

Case 1. Exponential processing times on machines A and B.

The optimal solution is to choose $d_a$ such that

$$\int_0^\infty \lambda_{a1} e^{-\lambda_{a1} x} \left[1 - e^{-\lambda_a (x + P_A + P_B - d_a)}\right] dx = R_a$$

$$\text{or} \quad d_a^* = P_A + P_B + \frac{1}{\lambda_a} \log (1 - R_a) + \frac{1}{\lambda_a} \log \left(\frac{\lambda_a + \lambda_{a1}}{\lambda_{a1}}\right) .$$

<u>Case 2. Uniform processing times on machines A and B.</u>

The optimal solution is to choose $d_a$ such that

$$\int_0^{2u_{a1}} \frac{1}{2u_{a1}} \frac{(x + P_A + P_B - d_a)}{2u_a} = R_a$$

$$\text{or} \quad d_a^* = P_A + P_B + u_{a1} - 2u_a R_a .$$

<u>State (3,3). A has not started on the bottleneck, and B has started on machine B.</u>

Machine B <—— y ——>

Bottleneck <— x —><—— $P_A$ ——><—— $P_B$ ——>
available for A

$f_{sb}(k)$ = the distribution of $\max\{x + P_A, y\}$.

<u>Case 1. Exponential processing times on machines A and B.</u>

In the exponential case, we get for $k \geq P_A$

$$f_{sb}(k) = \lambda_b e^{-\lambda_b k} \int_{P_A}^{k} \lambda_{a1} e^{-\lambda_{a1}(x - P_A)} dx$$

$$+ \lambda_{a1} e^{-\lambda_{a1}(k - P_A)} \int_0^k \lambda_b e^{-\lambda_b x} dx$$

$$= \lambda_b e^{-\lambda_b k} + \lambda_{a1} e^{-\lambda_{a1}(k - P_A)} - (\lambda_{a1} + \lambda_b) e^{-\lambda_b k - \lambda_{a1}(k - P_A)} .$$

The optimal solution is to choose $d_a$ such that

$$\int_{P_A}^{\infty} f_{sb}(k)\,[1 - e^{-\lambda_a(k + P_B - d_a)}]\,dk = R_a$$

or

$$d_a^* = P_A + P_B - \frac{1}{\lambda_a}\log\left[\frac{\lambda_{a1}}{\lambda_{a1} + \lambda_a} - \frac{\lambda_{a1}\lambda_a\, e^{-\lambda_b P_A}}{(\lambda_a + \lambda_b)(\lambda_a + \lambda_b + \lambda_{a1})}\right] + \frac{1}{\lambda_a}\log(1 - R_a).$$

Case 2. Uniform processing times on machines A and B.

The optimal solution is to choose $d_a$ such that

$$\int_0^{2u_b - t_b} \left[\frac{k}{2u_{a1}}\,\frac{1}{2u_b - 2t_b - P_A} + \frac{k}{2u_b - t_b}\,\frac{1}{2u_{a1}}\right]\left[\frac{k + P_B - d_a}{2u_a}\right] dk$$

$$+ \int_{2u_b - 2t_b}^{P_A + 2u_{a1}} \frac{1}{2u_{a1}}\left[\frac{k + P_B - d_a}{2u_a}\right] = R_a$$

$$\text{or} \quad d_a^* = P_B - \frac{4u_{a1}u_a R_a + 3(2u_b - t_b)P_A}{2u_{a1} - P_A^2/(2u_b - 2t_b - P_A)}.$$

State (3,4). A has not started on the bottleneck, and B has not started on machine B.

Machine B $\longleftarrow z \longrightarrow\longleftarrow y \longrightarrow$

Bottleneck $\longleftarrow x \longrightarrow\longleftarrow P_A \longrightarrow\longleftarrow P_B \longrightarrow$ available for A

$f_{sb}(k)$ = the distribution of $\max\{x + P_A,\ y + z\}$

The optimal solution is to choose $d_a$ such that

$$\int_0^{\infty} f_{sb}(k)\,F_a(k + P_B - d_a)\,dk = R_a .$$

Case 1. Exponential processing times on machines A and B.

Similar to state (3,3) except $\lambda_b$ is replaced by $\lambda_b + \lambda_{b1}$.

Case 2. Uniform processing times on machines A and B.

Similar to state (3,3) except with added time delay at machine B. In general, the use of state information to determine delivery time can be characterized as matching the expected delivery time to the availability of the bottleneck machine in addition to allowing for safety time dependent on the relative value of machine capacity to holding cost.

## 4.4 Aggregate Planning Model with Lot Sizes and Deliveries as Variables.

In the previous sections, delivery times were determined when the lot sizes were known. In this section, both delivery times and lot sizes will be considered as variables. The traditional ELSP (relaxing scheduling constraints) is formulated as follows:

$$\text{minimize} \quad \sum (H_i T_i + A_i / T_i)$$

$$\text{subject to} \quad \sum (s_i / T_i + \rho_i) \leq 1 \ .$$

In this model, the optimal lot size is given by $r_i T_i$. The average inventory cost when delivery times are random variables is

$$\sum \frac{1}{T_i} \bar{h}_i r_i T_i \int_{-\infty}^{0} -x \ f_i(x - d_i) \ dx \ .$$

To shorten future notation, let $\varepsilon_i(d_i)$ be the expected earliness given

$d_i$, that is, $\varepsilon_i(d_i) = \int_{-\infty}^{0} -x\, f_i(x - d_i)\, dx$ ,

and $\tau_i(d_i)$ be the expected delay given $d_i$, that is,

$$\tau_i(d_i) = \int_{0}^{\infty} x\, f_i(x - d_i)\, dx \ .$$

Then the combined problem, relaxing scheduling constraints can be formulated as follows:

$$\text{minimize} \quad \sum \{H_i T_i + A_i / T_i + \bar{h}_i r_i \varepsilon_i(d_i)\}$$

$$\text{subject to} \quad \sum \{s_i / T_i + \rho_i + \tau_i(d_i) / T_i\} \leq 1 \ .$$

Assuming the constraint is binding, the Lagrangian relaxation of this problem is

$$\mathcal{L} = \sum \{H_i T_i + A_i / T_i + \bar{h}_i r_i \varepsilon_i(d_i) + \theta[s_i / T_i + \rho_i + \tau_i(d_i) / T_i]\}$$

where $\theta \geq 0$, the Lagrangian multiplier of the constraint, measures the imputed value of machine capacity. The optimal lot sizes are determined by solving

$$\frac{d\,\mathcal{L}}{d\,T_i} = H_i - \frac{1}{T_i^2}[A_i + \theta s_i + \theta \tau_i(d_i)] = 0 \ .$$

Since $\dfrac{d^2 \mathcal{L}}{d^2 T_i} = \dfrac{2}{T_i^3}[A_i + \theta s_i + \theta \tau_i(d_i)] \geq 0$ , L is convex in $T_i$.

Furthermore, $T_i^* = \sqrt{\dfrac{A_i + \theta s_i + \theta \tau_i(d_i)}{H_i}}$ .

Observe that this is similar to the traditional ELSP solution with the expected delay being added to the setup time. The optimal delivery times are determined by

$$\frac{d\,\mathcal{L}}{d\,d_i} = -\bar{h}_i r_i F_i(-d_i) + \frac{\theta}{T_i}\,[1 - F_i(-d_i)] = 0\,.$$

Since $\dfrac{d^2\,\mathcal{L}}{d\,d_i^{\,2}} = \bar{h}_i r_i f_i(-d_i) + \dfrac{\theta}{T_i}\, f_i(-d_i) \geq 0$ , $\mathcal{L}$ is convex in $d_i$.

Hence, we choose $d_i^*$ such that $F_i(-d_i) = \dfrac{\theta}{\theta + \bar{h}_i r_i T_i}$ .

Observe that this is similar to the solution developed in previous sections with $\theta$ being the value of machine capacity. To check convexity in both variables,

$$\frac{d^2\,\mathcal{L}}{d\,d_i\;d\,T_i} = -\frac{\theta}{T_i^{\,2}}\,[1 - F_i(-d_i)]\,.$$

The determinant of the Hessian is

$$\frac{2}{T_i^{\,3}}\,[A_i + \theta s_i + \theta\tau_i(d_i)][\bar{h}_i r_i f_i(-d_i) + \frac{\theta}{T_i}\, f_i(-d_i)]$$

$$- \left\{ \frac{\theta}{T_i^{\,2}}\,[1 - F_i(-d_i)] \right\}^2 ,$$

which is positive if

$$[A_i/\theta + s_i + \tau_i(d_i)]\;[1 + \bar{h}_i r_i T_i/\theta] \;>\; 1/2\;[1 - F_i(-d_i)]^2/\; f_i(-d_i)\,.$$

Observe that the right hand side of this inequality gets very small in the right tails of most distributions. For example, in the normal distribution, at the 75th percentile this expression equals 0.10, at the 90th percentile it is 0.03, at the 95th percentile it is 0.012, and at the 99th percentile it is 0.004 . Since we are focusing on the bottleneck machine, we can assume that deliveries will be scheduled to arrive close to on-time, hence we will be looking at the right tails of the distributions.

## 4.5 A Dynamic Programming Model.

### 4.5.1 General Formulation.

The traditional ELSP is infeasible if the capacity of the machine is exceeded, that is, when $\Sigma \rho_i > 1$. In this case, it is impossible to satisfy the total demand. Suppose, however, that demand is dynamic rather than constant and that, on the average, demand can be met. Consider the following model where superscript notation is used to denote specific periods.

Let $b^j$ be the backlog, or excess capacity required, at the end of period j -- normalized by machine capacity, that is, (hours of backlog)/(hours in period). We assume that backlog from a given period uses capacity from the next period. Then we get the transition equation

$$b^{j+1} - b^j = \sum [\rho_i^{j+1} + s_i / T_i^{j+1} + \tau_i(d_i^{j+1}) / T_i] - 1$$

or $$b^{j+1} = b^j + \sum [\rho_i^{j+1} + s_i / T_i^{j+1} + \tau_i(d_i^{j+1}) / T_i] - 1 .$$

Let k be the unit cost of exceeding machine capacity. The penalty cost due to backlog (charged at the end of each period) is

$$P(b^j) = \begin{cases} k\, b^j & , \text{ for } b^j > 0 , \\ 0 & , \text{ for } b^j \leq 0 . \end{cases}$$

The average cost for period j is

$$C^j = \sum \; [1/2 \; h_i r_i^j (1 - \rho_i^j) T_i^j + \bar{h}_i r_i^j \varepsilon_i (d_i^j) + K_i / \; T_i^j] \; .$$

$$\text{Let} \quad G^j(b^{j-1} - b^j) \; = \; \min \quad C^j$$

$$\text{subject to} \;\; \sum \;\; [\rho_i^j + s_i / \; T_i^j + \tau_i (d_i^j) / \; T_i^j] \leq 1 - b^{j-1} + b^j \; ,$$

where $b^{j-1} - b^j$ is the relative capacity in period j used to diminish the backlog. Then we can formulate the problem as a dynamic programming problem.

Let $W^j\,(b^{j-1})$ be the optimal cost of going to the end of the planning horizon, given that we start period j with backlog $b^{j-1}$. For the last period, assume $b^n = 0$, that is, we have no backlog at the end of the planning horizon. Then for the last period, we have

$$\begin{aligned} W^n(b^{n-1}) \;\; &= \;\; G^n(b^{n-1}) \\ &= \;\; \min \;\; C^n \end{aligned}$$

$$\text{subject to} \;\; \sum \;\; [\rho_i^n + s_i / \; T_i^n + \tau_i (d_i^n) / \; T_i^n \;] \;\; \leq \;\; 1 - b^{n-1} \; .$$

The solution to this is given by

$$T_i^{n\;*} = \sqrt{\frac{K_i + \theta \sum \; [s_i + \tau_i (d_i^n)]}{1/2 \; h_i r_i^n \; (1 - \rho_i^n)}}$$

and
$$F_i(-d_i^{n^*}) = \frac{\theta}{\theta + r_i^n \bar{h}_i T_i^n} ,$$

where θ is chosen such that the constraint is satisfied. Observe that larger values of $b^{n-1}$ make the constraint tighter, hence the imputed value of machine time θ becomes higher. This causes longer cycle times and earlier deliveries and thus increases $C^n$. For the next to last period, we get

$$W^{n-1}(b^{n-2}) = \underset{b^{n-1}}{\text{Min}} \{P(b^{n-1}) + G^{n-1}(b^{n-2} - b^{n-1}) + W^n(b^{n-1})\} .$$

We only need to consider $b^{n-1} \leq 1 - \Sigma\, \rho_i^n$ since we assume $b^n = 0$. In general, the following recursion exists:

$$W^j(b^{j-1}) = \underset{b^j}{\min} \{P(b^j) + G^j(b^{j-1} - b^j) + W^{j+1}(b^j)\} .$$

We assume that the boundary conditions are $b^n = 0$ and $b^0 = 0$.

4.5.2 A Special Case.

Consider the following special case of the general problem developed in the previous section.

Let $\bar{h}_i = 0$ for all i, that is, ignore the delivery problem, and let $K_i = 0$ for all i, that is, setup costs equal zero.

Then for the solution to $G^j(b^{j-1} - b^j)$ we get

$$T_i^{j^*} = \sqrt{\theta / H_i^j \sum s_i} ,$$

where $\theta$ is chosen to make the constraint binding. By substitution we get

$$\theta = \left\{ \frac{\sum s_i \sqrt{H_i^j}}{\sqrt{\sum s_i}\,[1 - \sum \rho_i^j - b^{j-1} + b^j]} \right\}^2$$

and $$T_i^{j^*} = \left\{ \frac{\sum s_i \sqrt{H_i^j}}{\sqrt{H_i}\,[1 - \sum \rho_i^j - b^{j-1} + b^j]} \right\} .$$

Let $B^j = b^{j-1} - b^j$ . Then we can write $G^j$ as

$$G^j(B^j) = \frac{[\sum \sqrt{H_i^j}\,]\,[\sum s_i \sqrt{H_i^j}\,]}{[1 - \sum \rho_i^j - B^j]} .$$

$G^j(B^j)$ is a strictly increasing function of $B^j$ up to $B^j = 1 - \sum \rho_i^j$ and

$$d\,G^j / d\,B^j = G^j(B^j) / [1 - \sum \rho_i^j - B^j] .$$

Since $B^j$ is restricted to be less than $1 - \sum \rho_i^j$ , to minimize $G^j$ choose $B^j$ as small as possible, or for any given $b^{j-1}$, choose $b^j$ as large as possible. Looking at the overall problem for the last period, we have

$$W^n(b^{n-1}) = G^j(b^{n-1}) = \frac{[\sum \sqrt{H_i^n}\,]\,[\sum s_i \sqrt{H_i^n}\,]}{[1 - \sum \rho_i^n - b^{n-1}]} .$$

For the next to last period, we get

$$W^{n-1}(b^{n-2}) = \min_{b^{n-1}} \{P(b^{n-1}) + G^{n-1}(b^{n-2} - b^{n-1}) + W^n(b^{n-1})\} .$$

Assuming $b^{n-1^*} \geq 0$, we want to minimize

$$M^{n-1}(b^{n-1}) = k\, b^{n-1} + \frac{[\sum \sqrt{H_i^{n-1}}\,]\,[\sum s_i \sqrt{H_i^{n-1}}\,]}{[1 - \sum \rho_i^{n-1} - b^{n-2} + b^{n-1}]}$$

$$+ \frac{[\sum \sqrt{H_i^{n}}\,]\,[\sum s_i \sqrt{H_i^{n}}\,]}{[1 - \sum \rho_i^{n} - b^{n-1}]} .$$

Observe that if $b^{n-1^*} < 0$, the term $k\, b^{n-1}$ drops out.

$$\frac{d\, M^{n-1}}{d\, b^{n-1}} = k - \frac{[\sum \sqrt{H_i^{n-1}}\,]\,[\sum s_i \sqrt{H_i^{n-1}}\,]}{[1 - \sum \rho_i^{n-1} - b^{n-2} + b^{n-1}]^2}$$

$$+ \frac{[\sum \sqrt{H_i^{n}}\,]\,[\sum s_i \sqrt{H_i^{n}}\,]}{[1 - \sum \rho_i^{n} - b^{n-1}]^2}$$

$$\frac{d^2 M^{n-1}}{d\, {b^{n-1}}^2} = \frac{2[\sum \sqrt{H_i^{n-1}}\,]\,[\sum s_i \sqrt{H_i^{n-1}}\,]}{[1 - \sum \rho_i^{n-1} - b^{n-2} + b^{n-1}]^3}$$

$$+ \frac{2[\sum \sqrt{H_i^{n}}\,]\,[\sum s_i \sqrt{H_i^{n}}\,]}{[1 - \sum \rho_i^{n} - b^{n-1}]^3} \geq 0 .$$

Hence $M^{n-1}$ is convex in $b^{n-1}$ and a minimum with respect to $b^{n-1}$ can be found. Observe that if $b^{n-1^*} \leq 0$, the term k drops out of the first derivative and the second derivative remains the same, hence convexity is preserved.

Conditions for backlogging.

Using the above results, we can then define the condition for backlogging as

$$k < \frac{[\sum \sqrt{H_i^{n-1}}]\,[\sum s_i \sqrt{H_i^{n-1}}]}{[1 - \sum \rho_i^{n-1} - b^{n-2}]^2} - \frac{[\sum \sqrt{H_i^{n}}]\,[\sum s_i \sqrt{H_i^{n}}]}{[1 - \sum \rho_i^{n}]^2} .$$

For example, if the penalty for backlogging k were very small, then it might be advantageous to backlog capacity from one period to the next. The condition for negative backlogging, that is, getting ahead of demand, is

$$\frac{[\sum \sqrt{H_i^{n-1}}]\,[\sum s_i \sqrt{H_i^{n-1}}]}{[1 - \sum \rho_i^{n-1} - b^{n-2}]^2} < \frac{[\sum \sqrt{H_i^{n}}]\,[\sum s_i \sqrt{H_i^{n}}]}{[1 - \sum \rho_i^{n}]^2} .$$

This can be defined roughly as the ratio of scaled holding costs to slack machine capacity. If this ratio is smaller for one period than for the subsequent period, then it is advantageous to get ahead. The conditions where no backlogging is optimal can be defined as anything in between the two ranges described above. Suppose the optimal decision in period n-1 is $b^{n-1}$. Then for period n-2, we have

$$W^{n-2}(b^{n-3}) = \min_{b^{n-2}} \{P(b^{n-2}) + G^{n-2}(b^{n-3}-b^{n-2}) + W^{n-1}(b^{n-2})\}$$

$$= k\, b^{n-2} + \frac{[\,[\sum \sqrt{H_i^{n-2}}]\,[\sum s_i \sqrt{H_i^{n-2}}]}{[1 - \sum \rho_i^{n-2} - b^{n-3} + b^{n-2}]}$$

$$+ \frac{[\sum \sqrt{H_i^{n-1}}\,]\,[\sum s_i \sqrt{H_i^{n-1}}\,]}{[1 - E\ \rho_i^{n-1} - b^{n-2} + n^{n-1}]} + \frac{[\sum \sqrt{H_i^{n}}\,]\,[\sum s_i \sqrt{H_i^{n}}\,]}{[1 - \sum \rho_i^{n} - b^{n-1}]}$$

plus an additional term k $b^{n-1}$, if $b^{n-1} > 0$. Observe that we can define ranges for backlogging similar to that in period n-1.

## 4.6 Assembly Model

### 4.6.1 Introduction

The traditional ELSP assumes constant, continuous demand. An extension to this is suggested by the OPT problem (5) in which several parts are processed through a machine system and assembled. In this case, inventory holding costs for WIP are charged until the finished parts are assembled. There are three issues that must be addressed in this assembly model:

lot sizing (and scheduling),

sequencing,

and inventory pattern.

The inventory pattern issue will be discussed in the following two sections, with subsequent comments on lot sizing and sequencing.

### 4.6.2 Two Product Case

In the two product case (see Figure 4 - 6), we get an inventory pattern as illustrated in Figure 4 - 7.

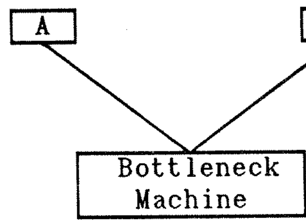



FIGURE 4 - 6

Machine Network

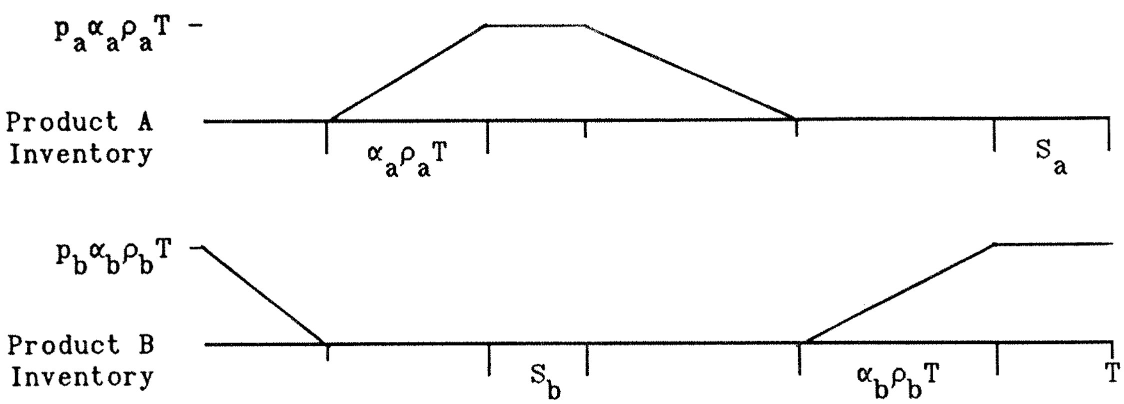



FIGURE 4 - 7

Two Product Inventory Pattern

In this inventory pattern, $\alpha_i$ is the fraction of the production time of i during which inventory is accumulated. During the other portion $(1 - \alpha_i)$ of the production time all parts produced are used for assembly.

Note that $p_b(1 - \alpha_b)\rho_bT = p_a\alpha_a\rho_aT$ and $p_a(1 - \alpha_a)\rho_aT = p_b\alpha_b\rho_bT$ .

The cycle length is defined by

$$T = \rho_aT + \rho_bT + S_a + S_b$$

$$= (S_a + S_b) / (1 - \rho_a - \rho_b) ,$$

where $S_i \geq s_i + \tau_i(d_i)$ , for $i =$ a and b.

Then we can formulate an expression for the average inventory, $\mathcal{S}$, of each part as follows:

$$\mathcal{S}_A = (p_a \alpha_a \rho_a T / 2)(\alpha_a \rho_a T / T) + (p_a \alpha_a \rho_a T)(S_b / T)$$

$$+ (p_a \alpha_a \rho_a T / 2)[(1 - \alpha_a)\rho_b T / T]$$

$$= (p_a / p_b)\ \alpha_a \rho_a / 2\ \ [\alpha_a \rho_a T\ (p_a + p_b) + 2\ p_b S_b]$$

$$\mathcal{S}_B = (p_b \alpha_b \rho_b T / 2)(\alpha_b \rho_b T / T) + (p_b \alpha_b \rho_b T)(S_a / T)$$

$$+ (p_b \alpha_b \rho_b T / 2)[(1 - \alpha_a)\rho_a T / T]$$

$$= (p_a / p_b)(1 - \alpha_a)\rho_a / 2\ \ \{(1 - \alpha_a)\rho_a T\ (p_a + p_b) + 2\ p_b S_a\} .$$

The total average cost to minimize is

$$C = H_a (p_a / p_b)\ \alpha_a \rho_a / 2\ \ [\alpha_a \rho_a T\ (p_a + p_b) + 2\ p_b S_b]$$

$$+ H_b (p_a / p_b)(1 - \alpha_a)\rho_a / 2\ \ [(1 - \alpha_a)\rho_a T\ (p_a + p_b) + 2\ p_b S_a]$$

$$+ K_a / T + K_b / T + r\bar{h}_a \varepsilon_a(d_a) + r\bar{h}_b \varepsilon_b(d_b)$$

subject to

$$0 \le \alpha_a \le 1$$

$$\tau_a(d_a) + s_a \le S_a$$

$$\tau_b(d_b) + s_b \le S_b .$$

First we find the minimum with respect to $S_a$ and $S_b$.

$$d\,C / d\,S_a = H_b p_a (1 - \alpha_a)\rho_a \ge 0$$

$$d\,C / d\,S_b = H_a p_a \alpha_a \rho_a \ge 0$$

Hence always choose $S_a$ and $S_b$ as small as possible, that is,

$$S_a = \tau_a(d_a) + s_a$$ , and similarly for B.

This gives $T = [\tau_a(d_a) + s_a + \tau_b(d_b) + s_b] / [1 - \rho_a - \rho_b]$.

Now that we know how to choose $S_a$ and $S_b$, we can find the minimum with respect to $\alpha_a$.

$$\begin{aligned} dC / d\alpha_a &= H_a(p_a / p_b)\, \rho_a / 2 \;\; [\alpha_a \rho_a T\, (p_a + p_b) + 2\, p_b S_b] \\ &\quad + H_a(p_a / p_b)\, \alpha_a \rho_a / 2 \;\; [\rho_a T\, (p_a + p_b)] \\ &\quad - H_b(p_a / p_b)\, \rho_a / 2 \;\; [(1 - \alpha_a)\rho_a T\, (p_a + p_b) + 2\, p_b S_a] \\ &\quad - H_b(p_a / p_b)(1 - \alpha_a)\rho_a / 2 \;\; \rho_a T\, (p_a + p_b) \\ &= \rho_a p_a / p_b \; \{H_a[\alpha_a \rho_a T\, (p_a + p_b) + p_b S_b] \\ &\quad - H_b[(1 - \alpha_a)\rho_a T\, (p_a + p_b) + p_b S_a]\} \end{aligned}$$

$$d^2C / d\alpha_a^{\,2} = \rho_a p_a / p_b \;\; (H_a + H_b)(p_a + p_b)\, \rho_a T \geq 0$$

Hence C is convex in $\alpha_a$. The optimal value of $\alpha_a$ is

$$\alpha_a^{\,*} = \frac{H_b}{H_a + H_b} + \frac{p_b}{p_a + p_b} \left\{ \frac{H_b}{H_a + H_b} \, \frac{S_a}{\rho_a T} - \frac{H_a}{H_a + H_b} \, \frac{S_b}{\rho_a T} \right\}$$

Suppose that $H_a = H_b$ and $S_a = S_a$. Then $\alpha^* = 1/2$. This gives us a general solution to the two product case. However, an important question is when does one product dominate the problem so that we never carry any inventory of that product?

<u>Conditions for dominant product.</u>

Under what conditions should we never accumulate finished inventory of A, that is, $\alpha_a^{\,*} \leq 0$? These conditions can be summarized as

$$\frac{H_a}{H_b} \geq \frac{\rho_a + \rho_b + S_a / T}{S_b / T} \geq \frac{\rho_a + \rho_b}{1 - \rho_a - \rho_b} .$$

This condition holds when the holding cost of A is much larger than the holding cost of B. The last inequality is strict unless $S_a = 0$. For example, if the machine utilization, $\rho_a + \rho_b$, equals 0.8, then $H_a/H_b$ must be at least 4. To find the minimum with respect to $d_a$,

$$d\,C / d\,d_a = H_b p_a (1 - \alpha_a)\rho_a \, [1 - F_a(-d_a)] - r\,\bar{h}_a F_a(-d_a) = 0.$$

$$d^2\,C / d\,d_a^{\,2} = H_b p_a (1 - \alpha_a)\rho_a f_a(-d_a) + r\,\bar{h}_a f_a(-d_a) \geq 0$$

Hence C is convex in $d_a$. Choose $d_a$ such that

$$F_a(-d_a) = \frac{H_a p_a (1 - \alpha_a)\rho_a}{H_a p_a (1 - \alpha_a)\rho_a + r\,\bar{h}_a} .$$

To check convexity in $d_a$ and $\alpha_a$ ,

$$d^2\,C / d\,d_a\,d\,\alpha_a = -H_b p_a \rho_a [1 - F_a(-d_a)] .$$

The determinant of the Hessian is positive if

$$\rho_a p_a / p_b \, (H_a + H_b)(p_a + p_b) \, \rho_a T \; [H_b p_a (1 - \alpha_a)\rho_a f_a(-d_a) + r\,\bar{h}_a f_a(-d_a)]$$
$$\geq H_b^{\,2} p_a^{\,2} \, \rho_a^{\,2} [1 - F_a(-d_a)]^2$$

or

$$\left\{\frac{H_a + H_b}{H_b} \; \frac{p_a + p_b}{p_b} \; \rho_a T\right\} \left\{ (1 - \alpha_a) + \frac{r\,\bar{h}_a}{H_b p_a \rho_a} \right\} \geq \frac{[1 - F_a(-d_a)]^2}{f_a(-d_a)} .$$

Observe that the right hand side of this inequality is the same as that developed in Section 4.4, and convexity again holds for reasonable problems.

### 4.6.3 Three Product Case.

Consider three products produced on a rotation cycle (see Figure 4-8).

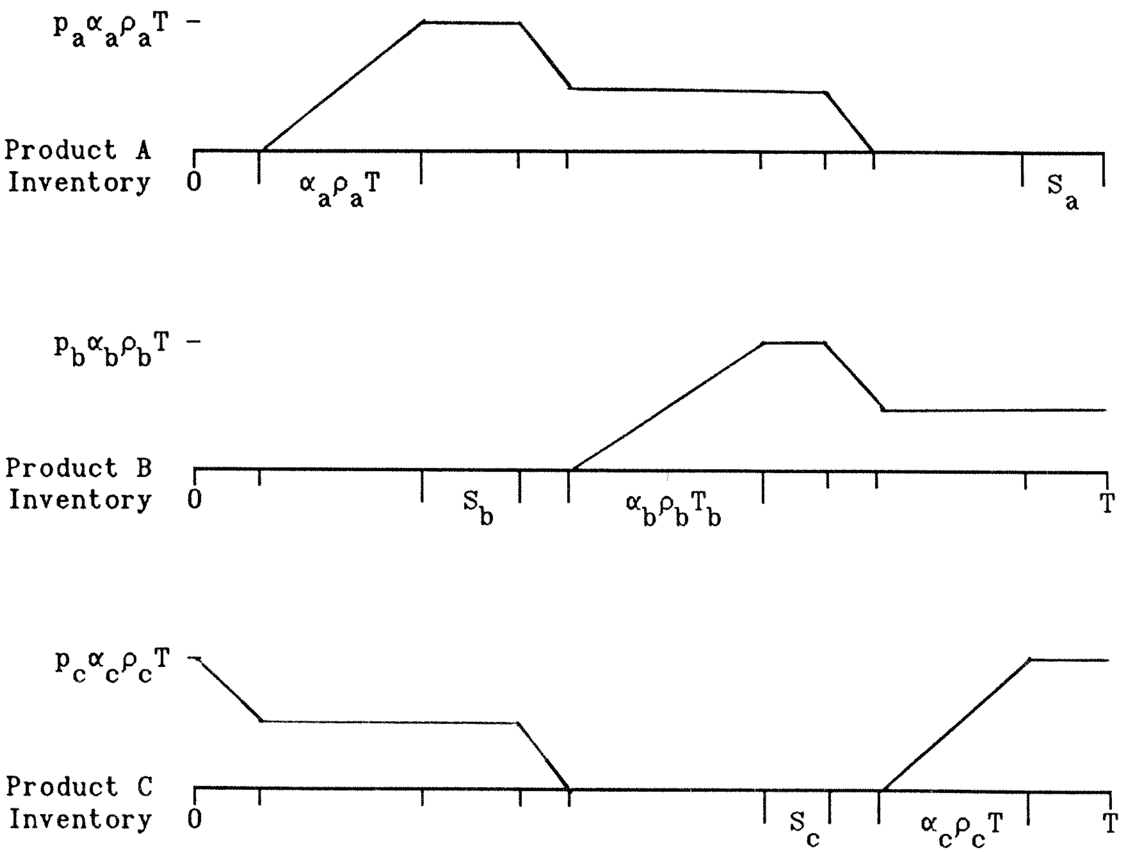



FIGURE 4 - 8

Three Product Inventory Pattern

The average inventory of each part is

$$\mathcal{I}_A = (p_a\alpha_a\rho_a T / 2)\,(\alpha_a\rho_a T / T) \;+\; (p_a\alpha_a\rho_a T)\,(S_b / T)$$

$$+ [p_a\alpha_a\rho_a T / 2 + p_c(1 - \alpha_c)\rho_c T / 2]\,[(1 - \alpha_b)\rho_b T / T]$$

$$+ [p_c(1 - \alpha_c)\rho_c T]\,[(\alpha_b\rho_b T + S_c) / T]$$

$$+ [p_c(1 - \alpha_c)\rho_c T / 2]\,[(1 - \alpha_c)\rho_c T / T]$$

$$\mathcal{I}_B = [p_b(1 - \alpha_a)\rho\ T\ /_a 2]_a[(1 - \alpha\ )\rho\ T\ /_a T]_a + (p_b\alpha_b\rho_b T)\ (S_c\ /\ T)$$

$$+ [(p\ \alpha\ \rho_b T_b/_b 2) + p\ (1 \underset{a}{-} \alpha\ )\rho\ \underset{a}{T}\ /_a 2]$$

$$\cdot\ [(1 - \alpha_c)\rho_c T\ /\ T]p_a(1 - \alpha_a)\rho_a T]\ [(\alpha_c\rho_c T + S_c)\ /\ T]$$

$$\mathcal{I}_C = [(p_c\alpha_c\rho_c T\ /\ 2) + p_b(1 - \alpha_b)\rho_b T\ /\ 2]\ [(1 - \alpha_a)\rho_a T\ /\ T]$$

$$+ [p_b(1 - \alpha_b)\rho_b T\ /\ 2]\ [(\alpha_b\rho_b T + S_b)\ /\ T]$$

$$+ [p_b(1 - \alpha_b)\rho_b T\ /\ 2]\ [(1 - \alpha_b)\rho_b T\ /\ T]$$

$$+ (p_c\alpha_c\rho_c T\ /\ 2)\ (\alpha_c\rho_c T\ /\ T)\ .$$

The following is the total inventory cost:

$$C = H_a\mathcal{I}_a + H_b\mathcal{I}_b + H_c\mathcal{I}_c\ .$$

C is convex in $\alpha_i$ and the determinant of the Hessian with respect to $\alpha_i$ and $\alpha_j$ is positive, hence we have convexity in two variables.

$$\alpha_a^* = \frac{p_a}{p_a + p_b}(1 - \alpha_c)\rho_c + \frac{p_b}{p_a + p_b}\ \frac{1}{\rho_a T}\left\{\frac{-H_aS_b + 2H_bS_c - H_cS_b}{H_a + H_b + H_c}\right\}$$

$$+ \frac{H_b}{H_a + H_b + H_c}\left\{1 + \frac{p_a\ p_b}{p_c(p_a + p_b)}\right\}$$

Suppose $p_a = p_b = p_c$ , $S_a = S_b = S_c$ , and $H_a = H_b = H_c$. Then $\alpha_a^* = 2/3$ .

## 4.7.4 Sequencing When One Product is Dominant.

Suppose one product, call it product A, is dominant so that no inventory of that product is ever accumulated. How should the remaining products be sequenced? Consider a three product case with inventory pattern shown in figure 4-9.

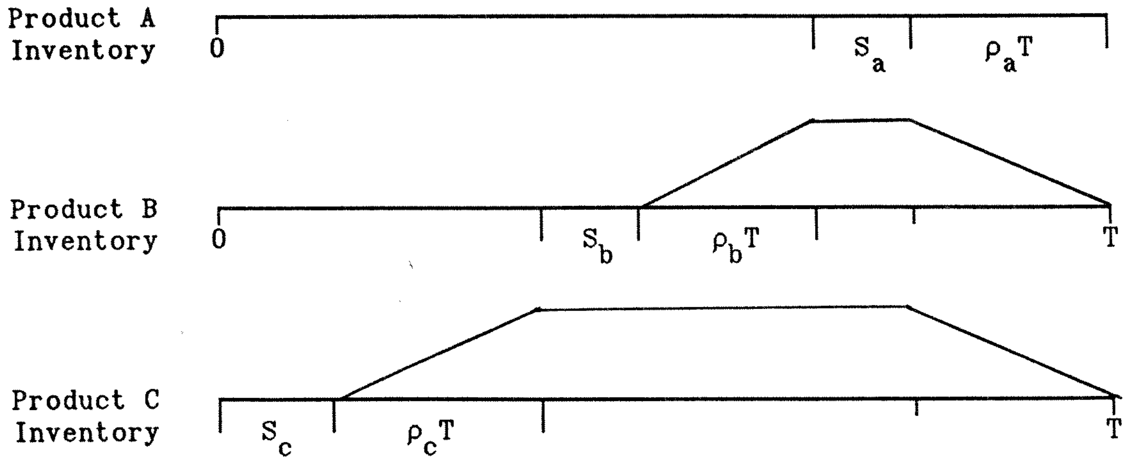



FIGURE 4 - 9

Three Product Inventory Pattern

The average inventory for B is

$$a_b = (rT / 2)\,(\rho_b T / T) + r\,(S_a / T) + (rT / 2)\,(\rho_a T / T)\ .$$

The average inventory for C is

$$a_c = (rT / 2)\,(\rho_c T / T) + r\,(S_a / T) + (rT / 2)\,(\rho_a T / T) + r\,[(S_b + \rho_b T) / T]\ .$$

Observe that the only term in these expressions that is dependent on the sequence of B and C is $r\,[(S_b + \rho_b T) / T]$. Then using the sequence dependent costs, B should follow C if

$$H_c r\,[(S_b + \rho_b T) / T] \leq H_b r\,[(S_c + \rho_c T) / T]$$

or

$$1/H_b\ (S_b + \rho_b T) \leq 1/H_c\ (S_c + \rho_c T)\ .$$

This is similar to classic scheduling theory's weighted shortest processing time rule except that here we use weighted longest processing time with the weights being the inverse of holding costs.

# CHAPTER 5

# AGGREGATE PLANNING ALGORITHM

## 5.1 Algorithm Development

### 5.1.1 Background

Section 4.4 presented a model which combined the issues of deliveries to a bottleneck machine and lot sizing. The optimal order intervals and requested delivery times were as follows:

$$T_i = \sqrt{\frac{A_i+\theta(S_i+\tau_i(d_i))}{H_i}}$$

and

$$F_i(-d_i) = \frac{\theta}{\theta+\bar{h}_i r_i T_i}\ , \qquad (1)$$

where $\theta$, the imputed value of machine capacity, is chosen such that the machine capacity constraint is satisfied at equality, that is,

$$\Sigma\{S_i/T_i + \rho_i + \tau_i(d_i)/T_i\} = 1$$

or

$$\Sigma\{S_i + \tau_i(d_i)\}/T_i = 1 - \Sigma\rho_i. \qquad (2)$$

If the setup cost is zero, that is, the value of setup consists only of the value of lost machine capacity,

$$T_i = \sqrt{\theta}\sqrt{\frac{S_i+\tau_i(d_i)}{H_i}} . \qquad (3)$$

Substituting this into the capacity constraint (2), we have

$$\Sigma \frac{1}{\sqrt{\theta}} \frac{S_i+\tau_i(d_i)}{\sqrt{\frac{s_i+\tau_i(d_i)}{H_i}}} = 1 - \Sigma\rho_i$$

or

$$\theta = \left\{\frac{\Sigma\sqrt{H_i(S_i+\tau_i(d_i))}}{1-\Sigma\rho_i}\right\}^2 . \qquad (4)$$

5.1.2 Algorithm

This leads to the following algorithm which iterates on the unknown imputed value of machine capacity:

Step 1. Let $\tau_i(d_i) = 0$ for all i.

Step 2. Compute $\theta$ using (4).

$$\theta = \left\{\frac{\Sigma\sqrt{H_i(S_i+\tau_i(d_i))}}{1-\Sigma\rho_i}\right\}^2 .$$

Step 3. Compute $T_i$ for all i using (3).

$$T_i = \sqrt{\theta}\sqrt{\frac{S_i+\tau_i(d_i)}{H_i}} .$$

Step 4. Compute $d_i$ for all i using (1).

$$F_i(-d_i) = \frac{\theta}{\theta + \bar{h}_i r_i T_i}$$

The inverse of F(.) depends on the form of the distribution distribution of delivery times.

Step 5. Compute $\tau_i(d_i)$ for all i. $\tau$ (.) depends on the form of the distribution of delivery times.

Step 6. Go to step 2 until $\theta$ converges. Convergence is assured by the convexity verified in chapter 4.

### 5.1.3 Normal Distribution Assumption

If we assume that the delivery times to the bottleneck machine are normally distributed, for step 5 of the algorithm we get

$$\tau_i(d_i) = \sigma_i \; f(-d_i/\sigma_i) + d_i \; F(d_i/\sigma_i)$$

where f(.) is the standard normal density function and F(.) is the standard normal cumulative distribution function. This algorithm is coded in PASCAL in Appendix 1.

## 5.2 Results

Two key questions to ask using this model are:

When can we ignore the delivery issue in scheduling a machine and what affect do the parameters of this model have on the overall system? These questions can be answered by looking at the results of various data input to the model.

### 5.2.1 Holding Costs for Raw Parts

We can assume that the ratio of holding costs for raw parts to the holding costs for processed parts is between 0 and 1, that is, the value of the raw parts lies somewhere between zero and the value of the processed parts. We can compare the resulting lot sizes determined from the aggregate model with the corresponding lot sizes determined ignoring the delivery issue. The ratio of these lot sizes, call it the $\beta$-ratio, is one when the value of raw parts is zero, hence we can ignore the delivery issue in scheduling a machine when the relative value of raw parts is small. In this case, we can carry sufficient inventory of raw parts to keep the bottleneck machine fully utilized because the raw parts inventory is very cheap. However, Figure 5-1 shows that as the relative value of raw parts increases, the corresponding lot sizes in the aggregate model also increase. Hence, as the relative value of raw parts increases, it becomes more important to consider the delivery issue when scheduling a machine.

TABLE 5 - 1

Data Used in Figure 5 - 1

| Part | $\rho_i$ | $s_i$ | $\sigma_i$ | $h_i$ |
|---|---|---|---|---|
| 1 | 0.25 | 3 | 60 | 4 |
| 2 | 0.33 | 2 | 40 | 2 |
| 3 | 0.33 | 1 | 20 | 3 |

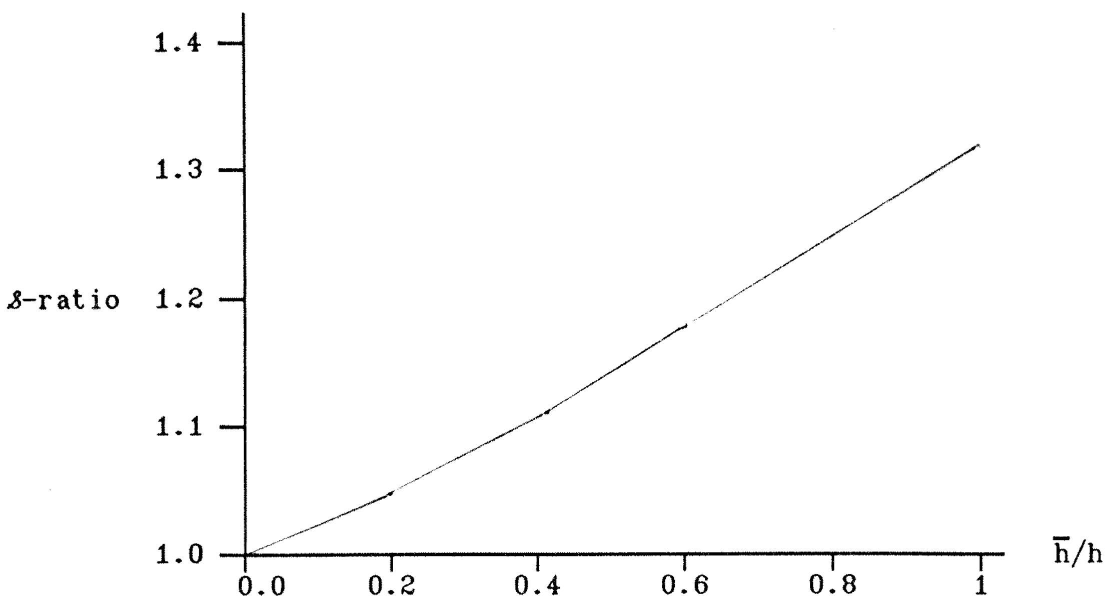



FIGURE 5 - 1

Graph

## 5.2.2 Variance of Delivery Times

If the variance of delivery times is small, we can ignore the delivery issue when scheduling a machine. To quantify how small the variance should be, we can compare the standard deviation of delivery time to the setup times. As shown in Figure 5-2, if the standard deviation of delivery time is small relative to the setup times, we can ignore the delivery issue. We also notice from Figure 5-3 that the total system costs go up at an almost linear rate with respect to the standard deviation of delivery time. Hence the variance of the input process plays a critical role in both lot sizes and overall costs.

TABLE 5 - 2

Data Used in Figures 5 - 2 and 5 - 3

| Part | $\rho_i$ | $s_i$ | $h_i$ | $\bar{h}_i$ |
|---|---|---|---|---|
| 1 | 0.200 | 3 | 5 | 4.0 |
| 2 | 0.250 | 2 | 1 | 0.8 |
| 3 | 0.125 | 1 | 2 | 1.6 |
| 4 | 0.250 | 3 | 3 | 2.4 |
| 5 | 0.111 | 2 | 4 | 3.2 |

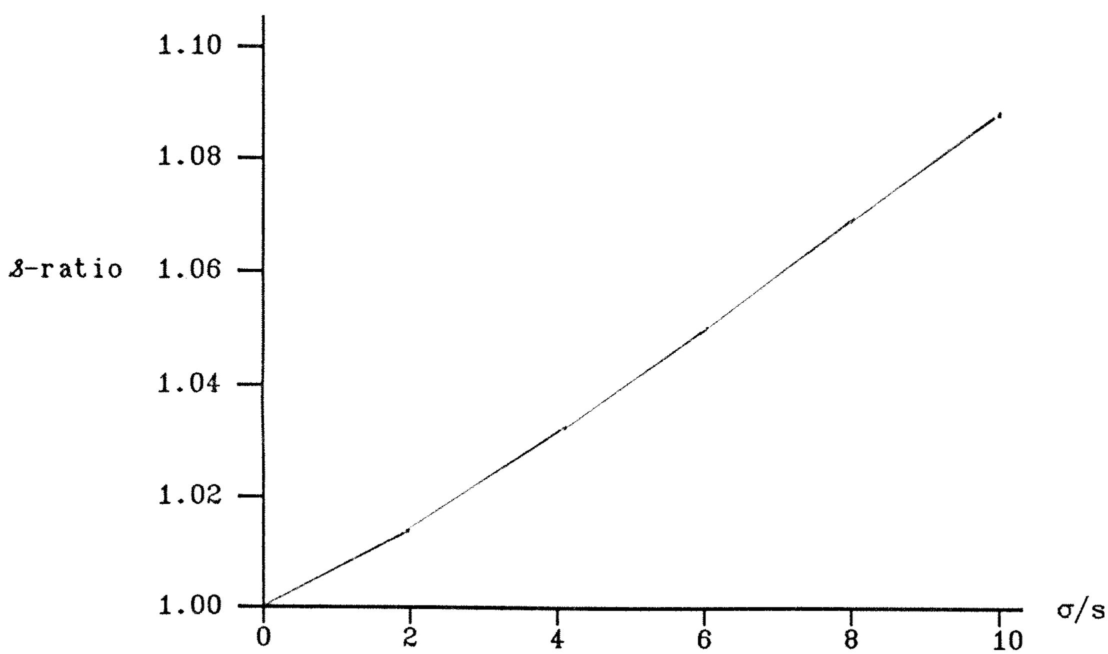



FIGURE 5 - 2

Graph

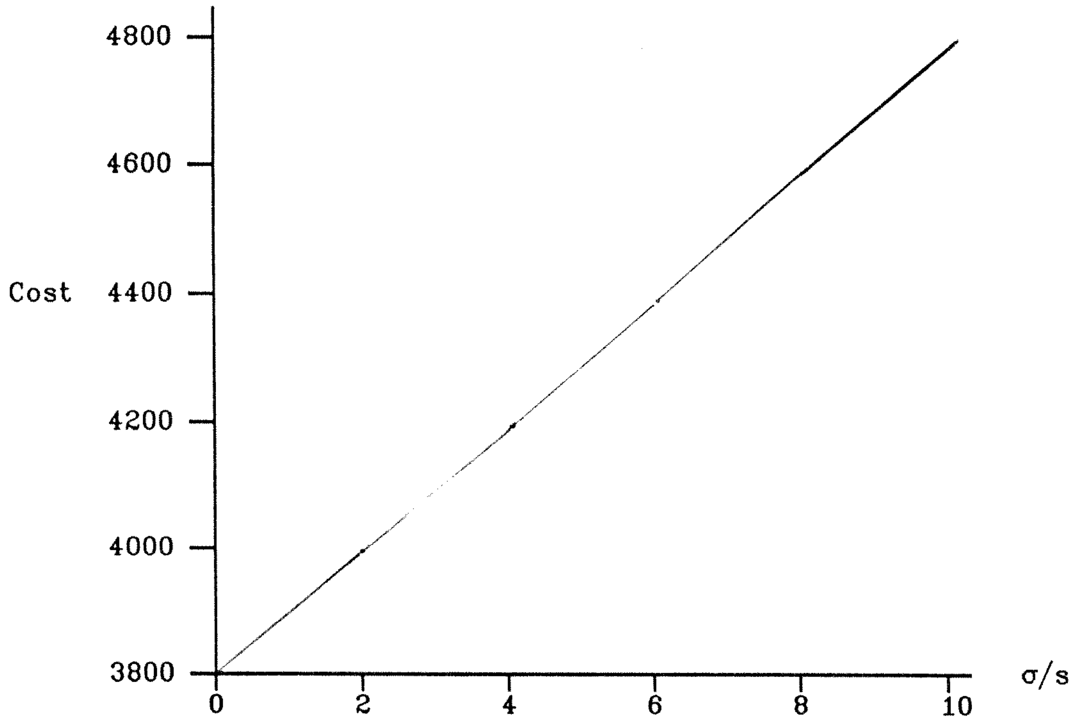



FIGURE 5 - 3

Graph

Reducing Setup Times

Much of the literature purporting the advantages of the Japanese philosophy of scheduling, just-in-time, support reducing setup times on machines as a means to decrease lot sizes. However, if we look at reducing setup times in the aggregate model, we find that the variance of the input process plays a critical role in determining how much the lot sizes can be reduced. As shown in Figure 5-4, reducing setup times produces a corresponding reduction in lot sizes until the setup time gets small compared to the variance of the input process. In Figure 5-5, we see that the ratio of lot sizes determined from the aggregate model compared to the corresponding lot sizes determined ignoring the delivery issue increases almost exponentially as the setup times approach zero. Hence, in trying to achieve just-in-time via small lot sizes, the variance of the input process must be reduced as well as reducing the setup time on the machine.

TABLE 5 - 3

Data Used in Figures 5-4 and 5-5

| Part | $\rho_i$ | $s_i$ | $\sigma_i$ | $h_i$ |
|---|---|---|---|---|
| 1 | 0.25 | 3 | 60 | 4 |
| 2 | 0.33 | 2 | 40 | 2 |
| 3 | 0.33 | 1 | 20 | 3 |

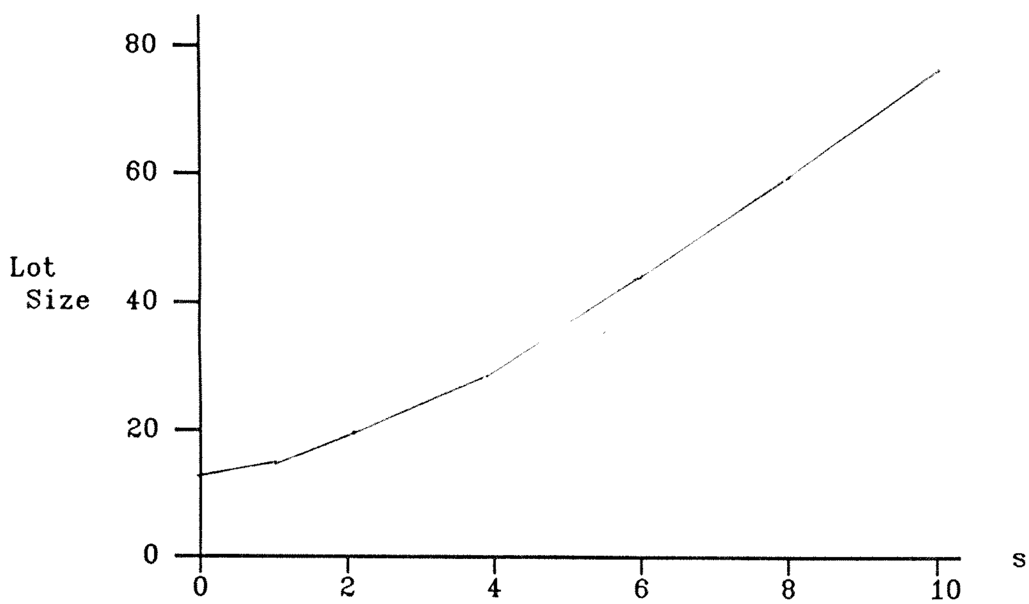



FIGURE 5 - 4

Graph

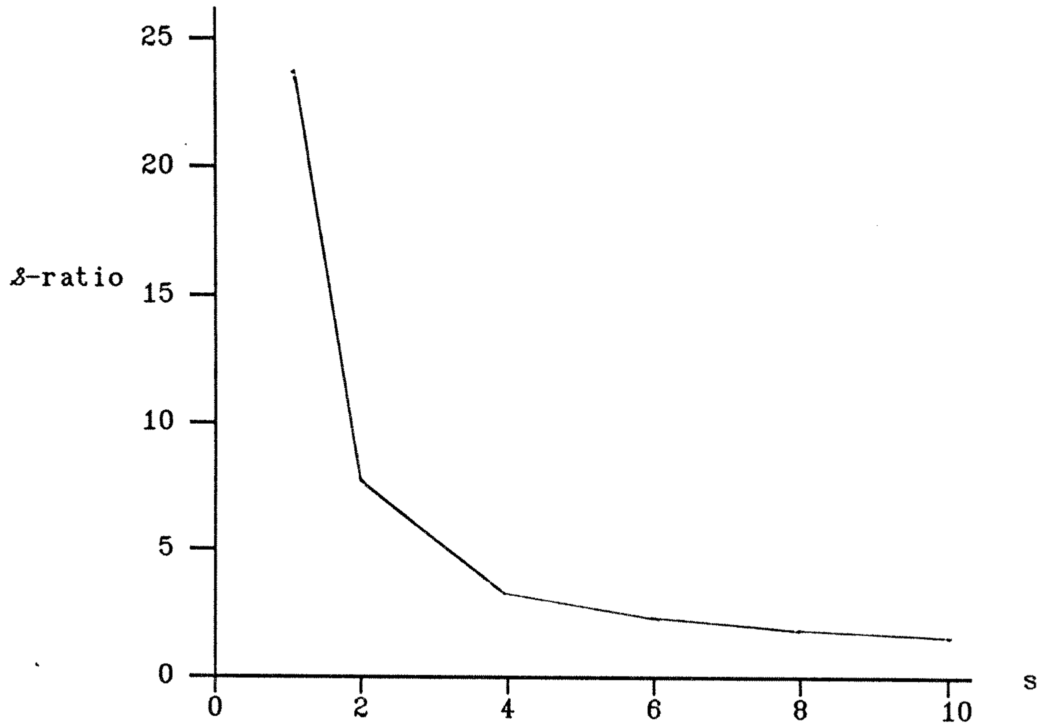



FIGURE 5 - 5

Graph

# CHAPTER 6

## CONCLUSIONS

If we put the lot scheduling problem in the context of the hierarchy of decision models and look both at the physical setting and the nature of the problem, we can get better solutions to a given problem in a more realistic setting. Due to the inherent difficulty in solving scheduling problems, the best we can usually hope for is heuristics which provide good solutions over a reasonable range of problems. However, if we encounter a problem with a dominant product, we can make use of that fact to simplify the search for a solution.

The concept of a dominant product can be used to determine when to insert idle time into a schedule. In this case, a dominant product is one with dominant holding costs. If in a given problem we have a dominant product and the remaining products have low machine utilization, that is, we have slack machine time available, then we can produce a better schedule by inserting idle time for the dominant product.

The role of the dominant product also tells us when not to use the zero switch rule (ZSR). We can conclude that the zero switch rule (ZSR) is a good scheduling policy for most problems. The exception to this happens when the ZSR solution yields lumpy production patterns for a dominant product. In this situation, we can sometimes improve upon the ZSR solution by making the dominant product's production pattern more even while using a non-zero switch on another product.

In looking at the context of a machine being scheduled, if there is variability in the input process to the machine, we can achieve lower overall system costs by considering the issues of determining delivery times and lot sizes concurrently instead of looking at each independently. We can use this approach both in aggregate planning and in real-time detailed scheduling. When determining the optimal delivery times, we balance the cost of holding raw parts with the value of lost machine capacity in a manner similar to the Newsboy Problem. This approach can be extended to consider problems with non-constant demand when we allow backlogging of demand and machine capacity. However, we find that much of the time, we will still try to satisfy current period demand with current period production.

The results of the aggregate model combining the issues of delivery and lot sizes show that the conditions when we should consider the delivery issue in conjunction with the lot size issue are when the holding cost of raw parts is high with respect to the holding cost of processed parts and when the variance of delivery times is high with respect to the corresponding setup times. We find that the overall system costs increase at an almost linear rate with respect to the standard deviation of delivery times.

In summary, we find that by placing a given problem in its proper context, we can more effectively derive solutions to the real problem at hand.

APPENDIX 1

AGGREGATE PLANNING ALGORITHM

```
PROGRAM sp (INPUT,OUTPUT,mdata); {***  TABLE OF CONTENTS

      1   Declarations
      2   Readin      -- reads input data
      3   Setup       -- sets up in proper format
      4   Norminv     -- finds inverse of normal CDF
      5   Normden     -- calculates density of std normal rv
      6   Cum         -- computes value of the normal CDF
      7   Thetacalc   -- calculates value of machine time
      8   Results     -- summarizes part schedules
      9   Display     -- writes results
      10  CONTROLLING PROGRAM                     ********************}

CONST p = 3;          {number of parts}
      n = 4;          {number of iterations}
      lead = 5;       {lead time for delivery}

TYPE  index    = integer;
      parray   = array [1..p] of real;

VAR   mdata: text;     {input file}
      r    : parray;   {demand rate}
      su   : parray;   {setup time}
      pr   : parray;   {production rate}
      ru   : parray;   {relative utilization  -  r/pr}
      h    : parray;   {holding cost after processing}
      hb   : parray;   {holding cost before processing}
      sd   : parray;   {standard deviation of delivery time}
      t    : parray;   {cycle length  - time between production runs}
      fcum : real;     {fraction of deliveries on time}
      comp : real;     {1 - fcum}
      ninv : real;     {normal inverse of comp}
      fden : real;     {density of delivery dist'n}
      d    : parray;   {scheduled delivery, neg = early, pos = late}
      tau  : parray;   {expected delivery delay}
      mtau : parray;   {expected early delivery}
      sum1 : real;     {sum of sqrt(h) * (su + tau) }
      sum2 : real;     {sum of su + tau }
      sl   : real;     {slack  = 1 - sum of ru }
      theta: real;     {value of machine time = sum1**2/(sum2*1**2)}
      cost : real;     {cost of solution at last iteration}
      i,j  : index;
```

```
PROCEDURE readin;          {read in data}
    var i: index;
    begin {readin}
       reset(mdata);
       for i := 1 to p do begin
          read(mdata,r[i]);
          read(mdata,su[i]);
          read(mdata,pr[i]);
          read(mdata,sd[i]);
          read(mdata,h[i]);
          read(mdata,hb[i]);
          end;
       close(mdata);
    end;  {readin}


PROCEDURE setup;           {setup data in proper format}
    var i: index;
    begin {setup}
       sl := 1.0;
       for i := 1 to p do begin
          ru[i] := r[i] / pr[i];
          sl := sl - ru[i];
          tau[i] := 0;     {intitializes tau to zero}
          end;
       writeln('   slack capacity =  ',sl);
    end;  {setup}

PROCEDURE norminv;{computes the appoximate inverse of the normal CDF}
                  {reference: HANDBOOK OF MATHEMATICAL FUNCTIONS by }
                  {Abramowitz and Stegun, error < .00045            }
                  {given comp, where  0 < comp < .5 , finds         }
                  {complementary cumulative inverse ninv,           }
                  {where 1 - F(x) = comp                            }
    var t1,t2,n1,d1,x: real;
    begin {norminv}
        t1 := 1.0;
        if comp > 0.5 then begin
           t1 := -1.0;
           comp := 1.0 - comp;
           end;
        t2 := sqrt(2*ln(1.0/comp));
        n1 := 2.515517 + 0.802853*t2 + 0.010328*t2*t2;
        d1 := 1.0 + 1.432788*t2 + 0.189269*t2*t2 + 0.001308*t2*t2*t2;
        x := t2 - n1/d1;
        ninv := t1*x;
    end;  {norminv}
```

```
PROCEDURE normden;        {calculates density of std normal rv  }
    var n2: real;
    begin  {normden}
       n2 := -(ninv*ninv)/2.0;
       fden := 0.3989423 * exp(n2);
    end;    {normden}

PROCEDURE cum; {computes the appoximate value of the normal CDF }
               {reference: HANDBOOK OF MATHEMATICAL FUNCTIONS by}
               {Abramowitz and Stegun, error < .00001           }
               {given ninv, finds value of the normal CDF, fcum }
    var t1,t2: real;
    begin {cum}
        if ninv > 0.0 then begin
           t1 := 1.0 / (1.0 + 0.33267*ninv);
           t2 := 0.4361836*t1 - 0.1201676*t1*t1 + 0.937298*t1*t1*t1;
           fcum := 1 - fden*t2;
           end
        else begin
           t1 := 1.0 / (1.0 - 0.33267*ninv);
           t2 := 0.4361836*t1 - 0.1201676*t1*t1 + 0.937298*t1*t1*t1;
           fcum := fden*t2;
           end;
    end;  {cum}


PROCEDURE thetacalc;      {calculates value of macine capacity}
    var i: index;
    begin {thetacalc}
       sum1 := 0.0;
       sum2 := 0.0;
       for i := 1 to p do begin
          sum1 := sum1 + sqrt(h[i]) * (su[i] + tau[i]);
          sum2 := sum2 + su[i] + tau[i];
          end;
       theta := sum1*sum1 / (sl*sl);
    end; {thetacalc}


PROCEDURE results;        {summarize results}
  var  i: index;
       cut: real;         {truncated left tail}
  begin
     cost := 0.0;
     for i := 1 to p do begin
        if lead + d[i] - 3*sd[i] < 0.0 then begin
           ninv := -lead / sd[i];  {truncates delivery to lead time}
           normden;                   {computes density of normal rv}
```

```
          cum;                   {computes cumulative dist'n of ninv}
          cut := sd[i] * fden + lead * fcum;        {expected early}
          mtau[i] := mtau[i] - cut;
          end;
        cost := cost + h[i]*t[i] + hb[i]*r[i]*mtau[i];
        end;
  end;


PROCEDURE display;      {write results}
   var i: index;
   begin   {display}
      writeln('INPUT DATA');
      writeln('    r/p        s         sd           h           hb');
      for i := 1 to p do begin
         writeln(ru[i],su[i],sd[i],h[i],hb[i]);
         end;
      writeln('RESULTS');
      writeln('Part No  Order Int   Delivery Time  Expected Delay');
      for i := 1 to p do begin
         writeln(i,t[i],d[i],tau[i]);
         end;
      writeln('  COST OF SOLUTION =  ',cost);
   end;     {display}


begin      {sp}
   writeln('Start program');
   readin;            {read in data}
   setup;             {setup data in proper format}
   writeln;
   for i := 1 to n do begin
      thetacalc;     {calculate value of machine time}
      writeln('Iteration',i,'   theta =  ',theta);
      for j := 1 to p do begin
       t[j] :=sqrt(theta *(su[j]+tau[j])/h[j]); {production interval}
         fcum :=theta / (theta+r[j]*hb[j]*t[j]); {deliveries on time}
         comp := 1.0 - fcum;       {complementary cumulative of fcum}
         norminv;               {computes functional inverse of fcum}
         d[j] := - ninv * sd[j];                  {scheduled delivery}
         normden;                     {computes density of normal rv}
         tau[j] := sd[j] * fden + d[j] * (1 - fcum); {expected delay}
         mtau[j] := sd[j] * fden - d[j] * fcum;      {expected early}
         end;
      end;
   results;           {summarize results}
   display;           {write reslults}
end.      {sp}
```

## APPENDIX 2
## THE BAKER AND BOMBERGER PROBLEMS

The Bomberger problem (3) is defined by the following parameters:

| $i$ | $s_i$ | $A_i$ | $h_i$ | $p_i$ | $\rho_i$ | $T_i^*$ |
|---|---|---|---|---|---|---|
| 1 | .125 | 15 | .00065 | 30000 | 400 | 167.5 |
| 2 | .125 | 20 | .01775** | 8000 | 400 | 37.7 |
| 3 | .25 | 30 | .01275 | 9500 | 800 | 39.3 |
| 4 | .125 | 10 | .01 | 7500 | 1600 | 19.5 |
| 5 | .5 | 110 | .2785 | 2000 | 80 | 49.7 |
| 6 | .25 | 50 | .02675 | 6000 | 80 | 106.6 |
| 7 | 1.0 | 310 | .15 | 2400 | 24 | 204.3 |
| 8 | .5 | 130 | .59 | 1300 | 340 | 20.5 |
| 9 | .75 | 200 | .09 | 2000 | 340 | 61.4 |
| 10 | .125 | 5 | .004 | 15000 | 400 | 39.3 |

$T_i^* = [A_i/H_i]^{1/2}$

**Originally .01175, however all subsequent authors have used .01775.

For this problem, we get the following results:

$$\sum \rho_i = 0.88$$

$$\sum s_i/T_i^* = 0.07$$

$$\sum s_i/T_i^* + \sum \rho_i = 0.95$$ (reference constraint (2-2) in Chapter 2).

Hence, only 95% of the machine capacity is utilized in the theoretical optimum solution.

The Baker problem (1) is defined by the following parameters:

| $i$ | $s_i$ | $A_i$ | $h_i$ | $p_i$ | $\rho_i$ | $H_i$ | $T_i^*$ |
|---|---|---|---|---|---|---|---|
| 1 | .08 | 75 | .01 | 2500 | .08 | .92 | 9.03 |
| 2 | .04 | 30 | .10 | 1000 | .25 | 9.38 | 1.79 |
| 3 | .02 | 25 | .04 | 500 | .20 | 1.60 | 3.95 |
| 4 | .12 | 35 | .08 | 200 | .35 | 1.82 | 4.39 |

$T_i^* = [A_i/H_i]^{1/2}$

For this problem, we get the following results:

$$\sum \rho_i = 0.88$$

$$\sum s_i/T_i^* = 0.06$$

$$\sum s_i/T_i^* + \sum \rho_i = 0.94 < 1$$ (reference constraint (2-2) in Chapter 2).

Hence, only 94% of the machine capacity is utilized in the theoretical optimum solution.

BIBLIOGRAPHY

1. BAKER, K.R., "On Madigan's Approach to the Deterministic Multi-Product Production and Inventory Problem," Management Science, Vol 16, No. 9 (1970), pp. 636-638.

2. BITRAN, G.B., HAAS, E.A., HAX, A.C., "Hierarhical Production Planning: A Single Stage System," Operaions Research, Vol. 29, No. 4 (1981), pp. 717-743.

3. BOMBERGER, E., "A Dynamic Programming Approach to a Lot Size Scheduling Problem," Management Science, Vol. 12, No. 11 (1966), pp. 778-784.

4. CONWAY, R.W., MAXWELL, W.L., and MILLER, L.W., Theory of Scheduling, Addison-Wesley, Reading, Mass. (1967).

5. CREATIVE OUTPUT, "The OPT Quiz: A Challenge in Scheduling," advertisement in Industrial Engineering, Vol. 17, No. 3 (1985), pp. 38-39.

6. DELPORTE, C.M., "Lot Size and Sequence Decisions for N Products on One Facility with Deterministic Demand," unpublished Ph.D. Dissertation, Cornell University (1976).

7. DELPORTE, C.M. and THOMAS, L.J., "Lot Sizing and Sequencing for N Products on One Facility," Management Science, Vol. 23 (1977), pp. 1070-1079.

8. DEMPSTER, M.A.H., FISHER, M.L., JANSEN, L., LAGEWEG, J., LENSTRA, K., RINNOOY KAN, A.H.G., "Analytial Evaluation of Hierarhical Planning Systems," Operaions Research, Vol. 29, No. 4 (1981), pp. 707-716.

9. DOBSON, G., "The Economic Lot Sizing Problem: A Resolution of Feasibility," Working Paper Series No. QM8504, The Graduate School of Management, University of Rochestor, February 1985.

10. DOLL, C.L. and WHYBARK, D.C., "An Iterative Produced for the Single-Machine Multi-Product Lot Scheduling Problem," Management Science, Vol. 20, No. 1 (1973), pp. 50-55.

11. ELMAGHRABY, S.E., "The Economic Lot Scheduling Problem (ELSP) Review and Extensions," Management Science, Vol. 24 (1978), pp. 587-598.

12. FUJITA, S., "The Application of Marginal Analysis to the Economic Lot Scheduling Problem," AIIE Transactions, Vol. 10, No. 4, pp. 354-361.

13. GOYAL, S.K., "Scheduling a Multi-Product Single-Machine System," Operaional Research Quarterly, Vol. 24, No. 2 (1973), pp.261-266.

14. GOYAL, S.K., "Scheduling a Multi-Product Single Machine System-- A New Approach," Int. J. Prod. Res., Vol. 13, No. 5 (1975), pp. 487-493.

15. GRAVES, S.C., "Using Lagrangian Techniques to Solve Hierarchical Production Planning Problems," Management Science, Vol. 28 (1982), pp. 260-275.

16. GUNTHER, H.O., "The Design of an Heirarchical Model for Production Planning and Scheduling," in Lecture Notes in Economics and Mathematical Systems, Vol. 266, Multi-Stage Production Planning and Inventory Control, S. Axsater, Ch. Schneeweiss, and E. Silver, editors, (Springer-Verlag, 1986), pp. 227-260.

17. HAX, A.C., and GOLOVIN, J.J., "Hierarchical Production Planning Systems," in Studies in Operations Management, A.C. Hax, editor, (Elsevier, Amsterdam, 1978), pp. 400-428.

18. HAX, A.C., and MEAL, H.C., "Hierarchical Integration of Production Planning and Scheduling," in TIMS Studies in Management Science, Vol 1, Logistics, M.A. Geisler, editor, (Elsevier, Amsterdam, 1975), pp. 53-69.

19. HAESSLER, R.W., "An Improved Extended Basic Period Procedure for Solving The Economic Lot Scheduling Problem, AIIE Ttansactions, Vol. 11 (1979), pp. 336-340.

20. HSU, W., "On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine," Management Science, Vol. 29 (1983), pp. 93-105.

21. KARMARKAR, U.S., "Lot Sizes, Manufacturing Lead Times and Utilization," Working Paper, The Graduate School of Management, University of Rochestor, May 1983.

22. KARMARKAR, U.S., "Lot Sizing and Sequencing Delays," Working Paper, The Graduate School of Management, University of Rochestor, June 1983.

23. MADIGAN, J.G., "Scheduling a Multi-Product Single Machine for an Infinite Planning Period," Management Science, Vol. 14 (1968), pp. 713-719.

24. MAXWELL, W.L., "An Investigation of Multi-Product Single Machine Scheduling and Inventory Problems," Unpublished Doctoral Thesis, Cornell University, 1961.

25. MAXWELL, W.L., "The Scheduling Of Economic Lot Sizes," Naval Research Logistics Quarterly, Vol. 11, No. 2-3 (1964), pp. 89-124.

26. MAXWELL, W.L., and MUCKSTADT, J.A., "Establishing Consistent and Realistic Reorder Intervals in Production-Distribution Systems," Operaions Research, Vol. 33, No. 6 (1985), pp. 1316-1341.

27. MAXWELL, W.L., MUCKSTADT, J.A., THOMAS, L.J., and VANDEREECKEN, J., "A Modeling Framework for Planning and Control of Production in Discrete Parts Manufacturing and Assembly Systems," Interfaces, Vol. 13, No. 6 (1983), pp. 92-104.

28. MAXWELL, W.L., and SINGH, H., "The Effect of Restricting Cycle Times in the Economic Lot Scheduling Problem," AIIE Transactions, Vol. 15, No. 3 (1983), pp.

29. ROUNDY, R., "Rounding Off to Powers of Two in the Economic Lot Scheduling Problem," Technical Report No. 663, School of Oper. Res. and Ind. Eng., Cornell University, July 1985.