

### **3. Dynamic overtime decision model**

#### **Introduction**

In this chapter we develop models to evaluate a production plan for an unreliable machine, and determine when it is cost optimal to run overtime. To motivate this discussion and place these models in context, let us consider the following (intentionally oversimplified) example. Suppose we must deliver 500 units of our product to our customer tomorrow morning, but we only have 400 units in inventory, so we must manufacture 100 units. It is now 4:00 PM; there is one hour left in the work day; our machine is currently set up to produce this product; and the machine can produce 200 units per hour. Unfortunately, the machine fails on average every 30 minutes and when it fails, requires 15 minutes on average to fix. If the machine did not fail, we could produce the 100 units in half an hour. However, due to machine failures, there is some probability that we will not be able to produce the 100 units by the deadline.

The production manager is now faced with several questions. What is the probability that we will be able to meet our demand? What is the expected shortfall? At 4:01 PM, the machine fails. Now what is the probability that we will be able to meet our demand? The production manager now considers using overtime. Suppose union rules dictate that plant management must decide by 4:30 PM if overtime will be run for one hour at a cost of \$200. What should the decision be? Suppose instead the amount of overtime can be chosen, up to 4 hours. How much should be chosen? Suppose instead that after running one hour of overtime, the production manager can stop overtime at any point. When should we stop? Suppose that we can delay shipping the product until noon if we pay \$5 per unit extra for express freight shipping. How does this change the decision?

Next, consider a case where the plant manufactures two products. Suppose we are manufacturing product #1 and have just produced enough units to satisfy our demand for tomorrow. Our production schedule, however, dictates that we continue manufacturing product #1 for another 200 units, at which point we are scheduled to perform a changeover to begin manufacturing product #2. Due to random machine failures, we do not know what time we will finish producing product #1, but we *expect* to begin producing product #2 around 4:00 PM. As before, this would leave us one hour to manufacture 100 units that we must ship by tomorrow morning.

The production manager is now faced with an even more difficult set of questions. Now what is the probability that we will be able to meet our demand, and what is the expected shortfall? Should we stop production of product #1 before we build the full economic lot size? How would that impact our ability to meet our next shipping deadline?

This chapter will develop a series of models that will assist a decision maker in answering questions such as the ones posed above. These models could be used as part of a manufacturing control system in a real manufacturing operation. One can (and should) envision these models embedded in a software tool that would receive data in real-time from the shop floor and assist plant management in decision making.

### **Literature review**

In this subsection we briefly review the literature that is related to the problem of deciding when to run overtime on an unreliable machine. Although no paper

addresses this specifically, many papers have addressed some aspect of this problem. We will divide our literature review into two parts: those that incorporate overtime opportunities, and those that model an unreliable machine. Our intent is not to cite every paper that has been written on these subjects, but rather to give the reader a sense for the types of models that have been studied by others. The interested reader is also referred to the literature review in Chapter 1.

### Unreliable machine

The presence of machine unreliability in a manufacturing system has been studied in a variety of different contexts, including problems of sequencing, scheduling, and lot sizing. We briefly review each of these areas.

There has been some limited work on sequencing of jobs on an unreliable machine. The earliest work is that of Glazebrook (1984), who models the problem as a rather general cost-discounted Markov decision process. He shows the conditions under which the optimal policy is of an index type (i.e., the job to be processed is the one with the smallest *Gittins index*; see Gittins, 1979). Pinedo and Rammouz (1988) find the optimal non-preemptive policies for several objective functions in the case of a Poisson failure process. For a general failure process and a discrete time model, Birge and Glazebrook (1988) find bounds on the error of following the strategy that is optimal when the failure process is memoryless. Birge et al. (1990) study in greater detail the problem of minimizing weighted flow-time and obtain results that are consistent with and complementary to Pinedo and Rammouz. For a detailed and current overview of this research area, see Pinedo (1995).

There has also been some work on lot sizing on an unreliable machine. Groenevelt et al. (1992a, 1992b) extend the basic economic manufacturing quantity (EMQ) model

to incorporate the effects of machine breakdowns. The first paper assumes that repairs are instantaneous but bear a fixed cost. The second paper assumes (as we do) that repairs are not instantaneous but instead consume machine time. This model permits any repair time distribution, but assumes that the time between failures is exponentially distributed. Under the assumption of lost sales, the authors seek an optimal lot size and safety stock level to minimize cost subject to a constraint on the service level. They require, however, some awkward assumptions regarding safety stock to achieve separability in the optimization of the lot sizes and safety stock level. The authors do not explore the impacts of multiple parts sharing the same machine.

Other authors, such as Sethi and Zhang (1994) have approached the problem from a control theoretic perspective. These authors consider the problem of finding an optimal setup schedule (a sequence of parts and the times at which the changeovers will occur) for an unreliable machine. They show that in the limit (as the length of the horizon tends to infinity), the stochastic problem can be reduced to a deterministic problem, and show how to obtain the optimal control policy. The authors also cite many other similar works.

Reiman and Wein (1994) study a two customer class, single server system with setups. The authors use heavy traffic diffusion approximations to analyze a system with a renewal arrival process, general service times, and either setup costs or setup times. They solve a control problem to minimize a linear function of the queue length plus setup costs, if any. Within these heavy traffic diffusion approximations one could model the unreliability of the machine within the service time distribution.

### Overtime opportunities

There has been little work on modeling manufacturing systems where overtime opportunities exist. The research that we have found is quite different from the problem context presented here. Some of these models have treated overtime decisions as a tactical planning problem. For example, Gelders and Kleindorfer (1974, 1975) present a coordinated planning and scheduling model for a one-machine job shop with overtime opportunities. The planning problem is to determine overtime usage levels in each period over a finite horizon, where costs can be time varying. The scheduling model determines job release dates to minimize tardiness plus flow time costs. The authors present a branch and bound scheme and discuss many properties of the optimal solution.

In the area of scheduling, Matsuo (1988) has studied the problem of job sequencing on a single machine to minimize weighted total tardiness plus overtime costs. The author presents an approximate algorithm based on solving a capacitated transportation problem.

Adshead and Price (1989) investigate, via simulation, the impact of different overtime adjustment rates and rules for determining the amount of overtime and where to use it in a make-to-stock shop. They treat the shop as deterministic and stationary, with the exception of the demand pattern, which they obtained from real, non-stationary data. These authors find little value in frequently adjusting overtime levels, which is not surprising in light of the deterministic assumptions they have made.

Many authors have studied queueing systems in which the server is not always available, perhaps due to machines failures or overtime (or lack thereof). These models could be used to analyze a make-to-order system in which jobs arrive to the system from the outside. Federgruen and Green (1986, 1989) and Sengupta (1990) present a general model for a single machine and develop bounds and approximations for typical performance measures. Sengupta also gives exact results for the case of exponentially distributed off times. Bitran and Tirupati (1991) study an open network of queues with fixed overtime opportunities. Based on their earlier works, the authors develop an approximation for the work-in-process levels (queue lengths) at each work center.

### **Overview of this chapter**

In the next section we describe many of our assumptions and introduce much of the notation that we will use throughout the chapter. In Section 3.2 we show how to evaluate the expected cost of a given production plan. Section 3.3 describes how to formulate a dynamic program that extends the model of Section 3.2 to include a simple overtime decision problem. This model forms the basic building block that we extend and explore in subsequent sections. We describe the computational complexity of the algorithm and then exercise the model under a variety of scenarios and show its behavior under a variety of scenarios.

We then characterize the structure of the costs and optimal solution of the model and discuss a computational issue associated with this model in Sections 3.4 and 3.5. These sections can be omitted by the reader without loss of continuity.

In Section 3.6 we consider static optimal solutions, that is, the optimal solution where all decisions must be made at time zero and cannot be changed over the

horizon. We begin by showing how to find the static optimal solution by numerical integration or by numerical Laplace transform inversion. We then describe an approach that can more quickly identify the optimal solution in certain circumstances. Lastly, the cost of the dynamic solution is compared to the cost of the static optimal solution. We show that even under moderate uncertainty and a short horizon, there can be significant benefits to dynamic optimization.

In Section 3.7 we consider a variety of extensions to the basic model of Section 3.3. The first extension we consider is early overtime authorization. In some situations it is necessary to decide whether or not to run overtime earlier than the point at which overtime actually begins. We show how to accommodate this situation. In some cases, a simple revision of the inputs to the model is all that is required. At worst, a minor modification to the algorithm is required.

Previous sections assumed that the overtime opportunities are of fixed length. We consider two extensions that relax this restriction. The first extension allows overtime to be consumed in a series of discrete blocks. After a block of overtime is purchased, the overtime is performed and the resulting state of the system is observed before a decision must be made whether or not to purchase additional overtime. We show how, by adding additional stages, the dynamic programming algorithm can be used to incorporate this extension, provided that the overtime costs are convex and increasing as more overtime is consumed. The second extension allows choosing among a set of possible overtime opportunities of varying lengths. This corresponds to the situation in which the amount of overtime must be chosen before any overtime is begun. This second extension does not have a convexity restriction on the overtime costs. We first describe how the solution of the dynamic program provides us with information to easily evaluate

an overtime opportunity at time zero of variable size. We then show how to modify the dynamic programming algorithm to accommodate the case where there is a set of possible overtime opportunities of varying lengths in the middle of the horizon.

The last extension that we consider in Section 3.7 is a constraint on the number of overtime opportunities used over the horizon. We show how to modify the dynamic programming algorithm without a large increase in computation time. With limited additional computational effort, the resulting dynamic programming solution can also provide information about reductions in the number of opportunities available. We also show that without additional computational burden we can accommodate more elaborate constraint structures, such as a constraint on the number of overtime opportunities used over the first half of the horizon, and a separate constraint on the number of overtime opportunities used in the second half. Lastly we describe how, by similar methods, to incorporate a constraint on the quantity of overtime used (e.g., no more than eight hours per week). These extensions are not computationally burdensome.

In Section 3.8 we examine the impact of the finite horizon assumption that we have made in the preceding sections. First, we show empirically how the optimal decisions are affected by increasing the length of the horizon, and the factors that influence the rate at which the steady state is attained.

In Section 3.9 we explore certain types of rescheduling and sensitivity analysis. We begin by describing how to compute the marginal benefit of shifting production between two scheduled production batches. This information can help decide when it makes sense to “cut short”, i.e., shift some of today’s workload to a future time,

and when to “get ahead”. This is essentially sensitivity analysis on the production quantities. We show how this information can be used to estimate the shadow prices of the lengths of the overtime opportunities. We describe how to compute these marginal benefits with minimal computational effort if the machine reliability is the same across all parts. Lastly, we show that with minimal computational effort we can compute the sensitivity of the total cost to the demand quantities and to the overtime and shortage costs, irrespective of whether or not machine reliability is part dependent.

In Section 3.10 we attempt to make some progress toward modeling overtime decisions when demand is stochastic. We consider two special cases. The first incorporates stochastic demands in the special case where only one part is produced on the machine. The second special case we consider assumes that the demand for all parts occurs at the same point in time, there is only one such point over the horizon, and the uncertainty in the demand quantity is not revealed until the last moment. We show that this is essentially a multi-item newsvendor problem where the amount that can be ordered is constrained (due to available machine time), and the amount that is received is uncertain (due to machine unreliability). Given a production sequence, we show how to numerically find cost minimizing production quantities. We then show how to dynamically update this strategy based on the realized output of the machine. In particular, for any point in time we show how to find a critical inventory level, above which the production of the current part should be stopped so that production of the next part in sequence can begin. Lastly, we show how to determine a cost minimizing overtime decision.

### **3.1 Problem statement and notation**

In this chapter we will focus on a single machine that repetitively produces a set of parts. We will only consider cases in which batching is necessary on the machine, presumably because setups consume precious machine time, are expensive, or both. We will assume that this machine is unreliable, and further, that breakdown is the only source of uncertainty over a short horizon. We will consider a finite horizon and assume that the time and quantity of demand is known over this horizon.

The models described in this chapter will assume that a production schedule (described below) is given as input. In the next section we will describe how to evaluate the expected shortfall cost of a given production schedule. We will then expand this discussion in Section 3.3 to include options to run overtime, and describe how to determine when it is optimal to run overtime to minimize the expected overtime and shortfall costs.

Each of these assumptions was discussed with various individuals responsible for production planning and scheduling at a General Motors metal stamping plant. The overall conclusion was that these assumptions were reasonably consistent with their manufacturing system. First, each metal part is usually assigned to a single machine on which it will be produced, as machines are different and the dies and automation used to produce the parts are tailored to a specific machine. A single machine might be assigned as few as two or as many as twenty different metal parts. Between production runs, the machine must be stopped and a specialized changeover crew must set up the machine to produce the next part. Thus, changeovers are both costly and consume machine time. Some of the machines do fail quite often (many times per day) and incur highly variable repair times (a few

seconds to a few hours). Lastly, the requirements on the machine are often known with a reasonably high degree of certainty over a period of two weeks, during which each part would certainly be produced at least once. The schedulers conveyed that within a two week period, machine unreliability was the greatest source of disruption to the schedule, and that schedule disruptions were a common occurrence. For a more detailed description of a stamping line, see Kletter (1994). For a good overview of a real automobile stamping plant, see Brooke (1993).

### **Notation and assumptions**

In this chapter we will focus on a single machine that produces a set of parts indexed by  $k = 1, \dots, K$ . When the machine is working it produces parts at a deterministic rate, but is subject to random failures and random repair times. We assume that the failure times and repair times are each i.i.d. exponential random variables. We assume operation dependent failures, i.e., the machine can not fail while it is under repair, nor can it fail when it is not working or in changeover.

Our model will consider decisions over a finite horizon. The model takes as input a plan for production over this horizon. There are two parts to the production plan. The first is a production sequence that defines the number of production runs (and therefore, the number of changeovers) and which part will be produced during each production run. Note that because the machine is unreliable, the time at which each production run begins is not known in advance. The second half of the production plan is the quantity that is planned for each production run. Without loss of generality we assume that the production runs are indexed by  $i = 1, \dots, N$  in the order that they are planned. Changeover times between production runs can be sequence dependent but are assumed to be deterministic. Within these changeover times we can include the time for any planned maintenance.

Based on the above assumptions, let us define the following *inputs* to the model

$P_i$  = production rate during the  $i^{\text{th}}$  production run,

$\lambda_i$  = failure rate during the  $i^{\text{th}}$  production run,

$\mu_i$  = repair rate during the  $i^{\text{th}}$  production run,

$S_i$  = changeover time required to begin the  $i^{\text{th}}$  production run,

$Q_i$  = planned production quantity for the  $i^{\text{th}}$  production run, in parts,

$IK_i$  = part to be produced during the  $i^{\text{th}}$  production run.

Logically, we would expect that if  $IK_a = IK_b$ , then  $P_a = P_b$ ,  $\lambda_a = \lambda_b$  and  $\mu_a = \mu_b$ , although there is nothing in the model that requires this to be so. It will be assumed throughout that all times and rates are expressed in a common time unit.

We assume that demand for each part is known with certainty over the horizon, and that all of the demands occur at known points in time. Without loss of generality, index the demand points by  $j = 1, \dots, M$  in the order in which they occur.

Let

$JK_j$  = part demanded at the  $j^{\text{th}}$  demand point,

$D_j$  = cumulative number of parts of type  $JK_j$  demand at the  $j^{\text{th}}$  demand point,

$TD_j$  = time of the  $j^{\text{th}}$  demand point.

To ensure that our definition of  $D_j$  is clear, let us consider an example. Suppose the first four demand points are for parts 1, 2, 1, and 2, respectively, for quantities of 15, 7, 2, and 3. Then  $D_1 = 15$ ,  $D_2 = 7$ ,  $D_3 = 17$ , and  $D_4 = 10$ .

If there are not enough parts in inventory of type  $JK_j$  by the deadline  $TD_j$ , there is a stockout charge  $cs_j$  per unit not filled. The stockout charge is a one time penalty and therefore is not a function of the length of time that the unfilled demand is outstanding. The planned production quantities are not affected when stockouts occur; we assume that backordered demand must still be satisfied. These assumptions would be appropriate in a remote metal stamping plant, for example, where all demand must be filled, so extra freight costs must be paid for express shipment whenever a shipping deadline is missed, so that the shipment will arrive on time. At this point we do not assume any relationship between the production plan and the demand requirements.

We require that at any point in time, the current state of the system is known: current inventory levels and the machine state are assumed to be given as inputs. Accordingly, define

$I_k(t)$  = inventory of part  $k$  at time  $t$ ,

$(t) = 1$  if the machine is working at time  $t$ ,  $0$  if it is failed.

We emphasize that if the machine is in changeover,  $(t) = 1$  by assumption.

## 3.2 Evaluation of a production plan

In this section we will describe two ways in which we can evaluate a given production plan. The first is an algorithm that will be central to the development in the remainder of this chapter. We will also describe a simple calculus-based approach that relies on numerical methods.

### Algorithmic approach

We now describe one method to evaluate a production plan, as defined in Section 3.1. This first model is intended only to evaluate the expected cost of a particular production plan. Since we will not consider revenues in our model, the appropriate metric is minimization of total expected cost. Since this model considers decisions over a short horizon, we do not concern ourselves with discounting future costs, although this assumption could be relaxed without loss of generality. The only costs we include in the model at this point are shortfall costs incurred at the shipping deadlines  $TD_j$ .

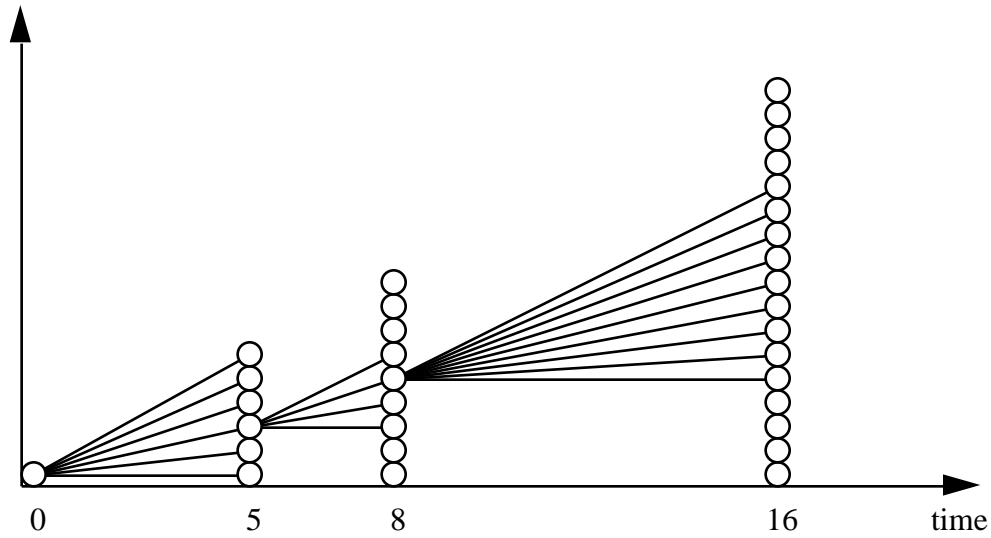
The key step for the evaluation of the production plan is how we model the state space. We are able to represent the state of the system at time  $t$  by two variables. The first is a quantity  $x(t)$  that denotes the amount of the production plan that has been completed by time  $t$  expressed in terms of machine time. The second is the (binary) state of the machine  $y(t)$ . We will denote the state of the system at time  $t$  by  $(t, x, y)$ . Our solution algorithm will require that a discretization of  $x$  be chosen. For the simplicity of our examples, we will discretize  $x$  in unit increments, although any discretization could be chosen.

Before we mathematically describe this system, we consider a few different visual interpretations. In Figure 3.1 we plot time on the horizontal axis and  $x$  on the vertical axis. By definition we start at  $x = 0$  at  $t = 0$ . Suppose the first demand point is at  $t = 5$ . The value of  $x$  that we reach at  $t = 5$  depends on the amount of time the machine has spent in the failed state. If the time axis and the  $x$ -axis are measured in the same units, then the largest value of  $x$  that we can achieve by  $t = 5$  is  $x = 5$  (if the machine does not fail). The stack of six circles at  $t = 5$  represent the feasible values of  $x$ , i.e.,  $x = 0, 1, \dots, 5$ .

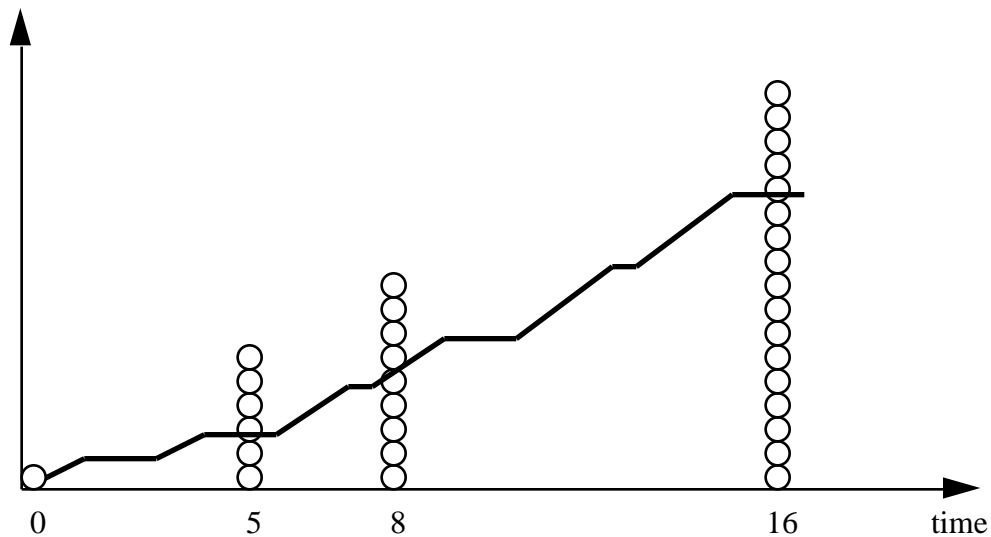
Suppose the next demand point is at  $t = 8$ . Irrespective of the value of  $x$  at  $t = 5$ , we can achieve at most 3 units of work in the interval between  $t = 5$  and  $t = 8$ . Therefore the maximum value of  $x$  that we can achieve by  $t = 8$  is  $x = 8$ . Further, it is also possible (although perhaps improbable) that  $x = 0$  at  $t = 8$ . Note that the points at which demand occurs need not occur at regular intervals.

In Figure 3.1 we represent the possible transitions in the state space from one point in time to the next by a straight line. We have not drawn all of the transitions that are possible from each state. We have only drawn the possible transitions from  $x = 0$  at  $t = 0$ , from  $x = 2$  at  $t = 5$ , and from  $x = 4$  at  $t = 8$ .

In Figure 3.2 we show one realization of this stochastic process. We plot the value of  $x$  for each point in time as a heavy black line. When the machine is working it produces parts at a (part-dependent) constant rate, so  $x$  increases linearly, and when the machine is failed  $x$  remains constant. This results in an upward sloping step-like function.



**Figure 3.1** State space representation



**Figure 3.2** State space with a realization of machine output

To evaluate a production plan we will not require all of the detail that is shown in Figure 3.2. We will only need to know the value of  $x$  at the demand points, since it is only at the demand points that penalty costs may be incurred. At a demand point, if a sufficient number of parts have been produced to satisfy all demand at the demand point, there will be no penalty costs. This means that there is a threshold value of  $x$  above which the immediate penalty cost at the state is zero. Below this

value of  $\beta$ , penalty costs increase as  $\beta$  is decreased, until the value of  $\beta$  is reached such that no demand is satisfied. The states in which penalty costs occur are darkened in Figure 3.3, where white indicates that no penalty was incurred, black indicates that no demand was satisfied, and shades of gray indicate the quantity of demand that was satisfied.

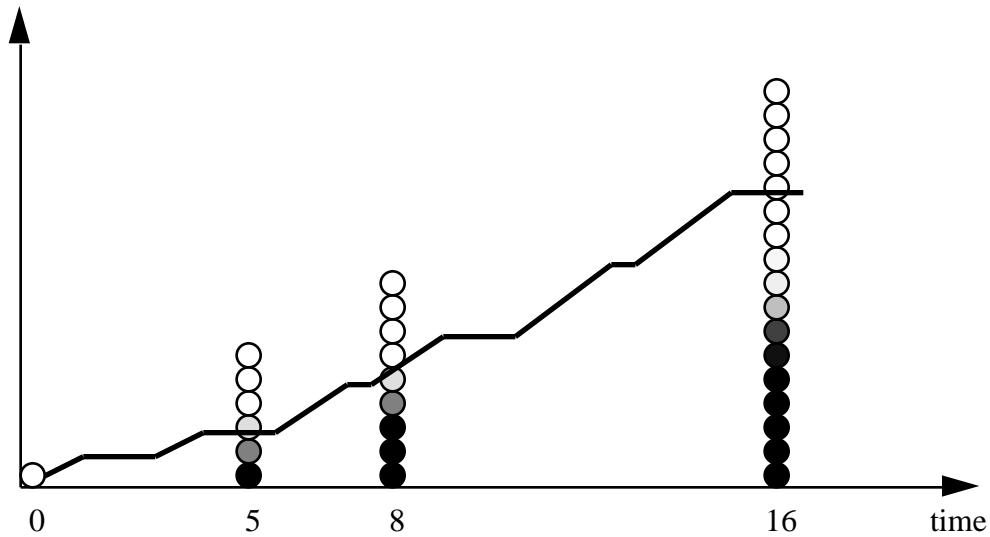


Figure 3.3 State space with penalty costs

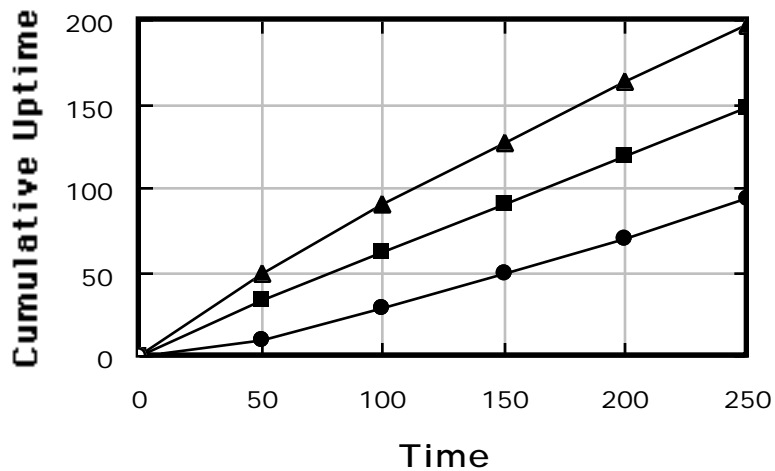


Figure 3.4 Confidence interval of machine uptime

From the results of Chapter 2 we can determine the probability that machine uptime over an interval is above (or below) any value. This allows us to trace the boundary of a confidence interval for machine uptime as a function of time. Figure 3.4 shows an example of this. The triangles, squares, circles are the 95%, 50% and 5%-iles of the cumulative output of the machine as of time zero, for parameters  $\lambda = 3/60, \mu = 4/60$ . The superposition of the penalty costs from Figure 3.3 and the confidence interval from Figure 3.4 gives the decision maker a visual indication of when and where stockouts are likely to occur.

We now formalize these notions by describing an algorithm for computing the expected future shortfall cost. Let  $c_j(x, t)$  be the expected future shortfall cost of the state  $(TD_j, x, t)$  and let  $c_0(x, t)$  be the expected future shortfall cost as of time zero. The algorithm proceeds backwards, starting at the last demand point. Let  $f_j(x)$  denote the shortfall at demand point  $j$  as a function of the value of  $x$ .  $c_M(x, t)$  is thus simply equal to the known penalty costs in each state,  $cs_{M-M}(x, t)$ . At the  $M$ -j<sup>th</sup> step of the algorithm we compute the expected future shortfall cost for the  $j$ <sup>th</sup> demand point for each state from

$$c_j(x_0, t_0) = \sum_{x_1=0}^1 \text{trans}(x_0, x_1; TD_{j+1} - TD_j | x_0, t_0) c_{j+1}(x_1, t_1) + cs_{j-j}(x_0, t_0)$$

where  $\text{trans}(x_0, x_1; T | x_0, t_0)$  is the probability of transitioning from  $x_0$  to  $x_1$  in an interval of length  $T$  if the initial machine state is  $x_0$  and the terminal machine state is  $x_1$ .

To determine the complexity of the algorithm, let  $s$  denote the number of discretized values that  $x$  can take. Then at each of the  $M$  demand points we must

compute the expected future shortfall cost of each possible state, of which there are  $O(s)$ . This requires computing  $O(s^2)$  transition probabilities. Given the transition probabilities, the expected future shortfall cost for any state is then found with a single vector multiplication requiring  $O(s)$  multiplications and additions. In total, this algorithm requires  $O(s^2 M)$  multiplications and additions, and the computation of  $O(s^2 M)$  transition probabilities. Thus, the time to compute the transition probabilities will dictate the running time of the algorithm.

The remainder of this subsection will provide the details for this algorithm, namely, how to compute  $U_i$ , the immediate penalty costs incurred as a function of  $U_i$ , the transition probabilities, and the computational effort required to find the transition probabilities. We address each of these in turn.

### Determination of

Denote the minimum time required to complete production runs 1, ..., i as  $U_i$ .

Then

$$U_i = \sum_{a=1}^i S_a + \frac{Q_a}{P_a}.$$

If we are currently producing the  $q+1^{\text{st}}$  part of the  $i^{\text{th}}$  production run, then  $U_i = U_{i-1} + S_i + q/P_i$ . Similarly, if we are  $s$  minutes into setting up for production run  $i$ , then  $U_i = U_{i-1} + s$ . In effect,  $U_i$  is a measure of cumulative output, measured in time units of machine uptime.

### Penalty costs

If the current state is  $x$ , the number of parts of production run  $i$  that have been produced is

$$N_i(x) = \min\left\{P_i \left[ x - (U_{i-1} + S_i) \right]^+, Q_i\right\},$$

where  $[x]^+$  denotes the greater of zero and  $x$ . Thus, each value of  $x$  uniquely defines how much of the production plan is completed. At demand point  $j$ , the shortfall is the cumulative demand minus the cumulative production of part  $k$  ( $k = JK_j$ ) minus any starting inventory. Therefore, the shortfall at demand point  $j$  is

$$s_j(x) = D_j - \sum_{i \in A_j} N_i(x) - I_k(0)$$

where  $A_j = \{i : IK_i = k\}$ , the index set of production runs for part  $k$ . The penalty costs incurred at demand point  $j$  are then  $cs_j - s_j(x)$ .

### Transitions between states

We will denote the current state as  $(t_0, i_0, s_0)$  and consider transitions to some future state  $(t_1, i_1, s_1)$  where  $t_1 > t_0$  and  $i_1 \geq i_0$ . Assume that  $s_0$  is such that at time  $t_0$  we are producing or setting up for the  $i^{\text{th}}$  production run, and  $s_1$  is such that at time  $t_1$  we are still in the  $i^{\text{th}}$  production run or setting up for the  $i+1^{\text{st}}$ . With these assumptions, the time available for production during  $[t_0, t_1)$  is

$$t_1 - t_0 - \left[ s_1 - U_i \right]^+ - \left[ U_{i-1} + S_i - s_0 \right]^+$$

where the expressions in brackets are zero if we are not setting up at the beginning or end of the interval. Further, in order to reach  $x_1$  by time  $t_1$ , we require the uptime over the interval  $[t_0, t_1)$  to be

$$x_1 - x_0 - [x_1 - U_i]^+ - [U_{i-1} + S_i - x_0]^+$$

where again the expressions in brackets are zero if we are not setting up at the beginning or end of the interval. More generally, if  $x_0$  is such that at time  $t_0$  we are producing or setting up the  $i^{\text{th}}$  production run, and  $x_1$  is such that at time  $t_1$  we are producing the  $j^{\text{th}}$  production run or setting up for the  $j+1^{\text{st}}$ ,  $j > i$ , then the time available for production during  $[t_0, t_1)$  is

$$t_1 - t_0 - [x_1 - U_j]^+ - [U_{i-1} + S_i - x_0]^+ - \sum_{k=i+1}^j S_k$$

and the required uptime over the interval  $[t_0, t_1)$  is

$$x_1 - x_0 - [x_1 - U_j]^+ - [U_{i-1} + S_i - x_0]^+ - \sum_{k=i+1}^j S_k.$$

Given these results, we can easily state that the condition for feasibility of transition from  $(t_0, x_0)$  to  $(t_1, x_1)$ : the time available for production must be no less than the required uptime.

Now that we have found the time available for production and the required uptime, we can compute the transition probabilities. Let us consider a simple numerical example. Suppose a transition from  $x_0$  to  $x_1$  means that over  $[t_0, t_1)$  we complete production of the last 20 units of part 1, incur a 30 minute setup, produce a

batch of 300 units of part 2, incur another 30 minute setup, and produce the first 10 units of part 3. If all time units are expressed in minutes and (for simplicity) all production rates are one per minute, then in the notation of Chapter 2, the probability of transition from  $(t_0, o, 0)$  to state  $(t_1, 1, 1)$  is

$$G_3(10; t_1-t_0-60, 20, 300 \mid 0, 1) - G_3(9; t_1-t_0-60, 20, 300 \mid 0, 1),$$

where  $G_i(x; T, T_1, \dots, T_{i-1} \mid (0)=0 \& (T)=1) = \Pr\{x \text{ or fewer parts have been produced in the } i+1^{\text{st}} \text{ production run} \mid \text{total time available for production} = T, \text{ first run requires } T_1, \dots, i-1^{\text{st}} \text{ run requires } T_{i-1}, \text{ machine is initially in state } 0 \text{ and ends in state } 1\}$ . Note that the transition from  $(t_0, o, 0)$  to state  $(t_1, 1+1, 1)$  is thus

$$G_3(11; t_1-t_0-60, 20, 300 \mid 0, 1) - G_3(10; t_1-t_0-60, 20, 300 \mid 0, 1),$$

and since we have already computed  $G_3(10; t_1-t_0-60, 20, 300 \mid 0, 1)$ , we must compute one additional value of  $G_i(\cdot)$  for each discretized interval of  $\Delta$ .

We have assumed in the above discussion that the discretization of the state space for  $\Delta$  occurs in single part increments, although a more fine or more coarse discretization can be chosen.

When machine failures and repairs are i.i.d. exponential (but with possibly different machine reliability parameters  $\lambda_k$  and  $\mu_k$  for each part), then the distribution  $G_i(x; T, T_1, \dots, T_{i-1})$  can be written as a convolution of  $i$  distributions of type R, as described in Chapter 2. However, if the machine reliability parameters  $\lambda_k$  and  $\mu_k$  are the same for all parts  $k = 1, \dots, i$ , then

$$G_i(x; T, T_1, \dots, T_{i-1}) = F(x; T - T_1 - \dots - T_{i-1});$$

that is,  $G_i(x; T, T_1, \dots, T_{i-1})$  is a distribution of type  $F(t; T)$  with machine parameters  $\mu_k$  and  $\lambda_k$ , where  $k = 1, \dots, i$ .

As described in the Appendix to Chapter 2, a distribution such as  $G$  or  $F$  can be evaluated at a point by Laplace transform inversion on a desktop computer in a fraction of a second. For example, on a Power Macintosh 7100/80 in emulation mode with SANE-based math instructions, the time required to evaluate a distribution of type  $F(t; T)$  to a high degree of accuracy (absolute error less than  $10^{-15}$ ) is on the order of 0.1 seconds. The computational effort required to evaluate  $G_i(x; T, T_1, \dots, T_{i-1})$  at a point by numerical Laplace transform inversion will be comparable to that for  $F(t; T)$ , except that the effort grows linearly in  $i$ . The rate of growth will depend on the computational effort required to evaluate the Laplace transform at a point.

As a final remark to this subsection, we note that the expected completion time of the production plan is

$$= \sum_{i=1}^N S_i + \frac{Q_i}{SAA_i P_i}$$

where  $SAA_i$  is the *stand-alone availability*  $\mu_i / (\lambda_i + \mu_i)^*$ . If we scale all time units such that the end of the horizon is at time 1,  $\lambda_i$  can also be interpreted as the

---

\* This is actually only an approximation if the initial state of the machine is known. However, if the length of the horizon is large relative to the MTBF and MTTR, the quality of the approximation will be excellent.

utilization of the machine required to complete all production by the end of the horizon. In this way  $U_{aj}$  gives us some indication for the criticality of the load on the machine. Although this is an important metric, it is not a substitute for the evaluation procedure that we have just described since it can not tell us the likelihood that we will make our shipments on time nor, perhaps more importantly, the expected shortfall.

### Formulation using calculus

The evaluative model can also be written as a summation of linear loss integrals that compute the expected shortfall cost at each demand point. In particular, the total expected cost can be written as

$$\sum_{j=1}^M \sum_{a=1}^{|A_j|} c_{s_j} L_{aj} + \left( D_j - I_{JK_j}(0) - Q_{a-1,j} \right)^+ \times G_{A_j(a)}(0; T_{A_j(a),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}) - G_{A_j(a-1)} \left( \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}; T_{A_j(a-1),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-2)}}{P_{A_j(a-2)}} \right),$$

where

$$L_{aj} = \int_0^{Q_{A_j(a)}} \left( D_j - I_{JK_j}(0) - Q_{a-1,j} - x \right)^+ g_{A_j(a)} \left( \frac{x}{P_{A_j(a)}}; T_{A_j(a),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}} \right) dx,$$

$Q_{aj} = Q_{A_j(1)} + \dots + Q_{A_j(a)}$ , and  $T_{ij} = TD_j - S_1 - \dots - S_i$ . We define  $Q_{0j} = 0$  and the cumulative distribution  $G_{A_j(0)}(0; T_{A_j(0),j})$  to equal 0. Recall that  $A_j$  is the index set of production runs for the part demanded at the  $j^{\text{th}}$  demand point. We have assumed that the members of the set  $A_j$  are  $A_j(1), A_j(2), \dots, A_j(|A_j|)$ , indexed such that  $IK_{A_j(a)} >$

$IK_{A_j(b)}$  if  $a > b$ . The notation  $( )^+$  denotes the greater of zero and the expression in parentheses.

Although notationally cumbersome, the above expression has a very simple interpretation.  $D_j - I_{JK_j}(0) - Q_{v_{a-1,j}}$  is the shortfall at demand point  $j$  if production runs 1, ...,  $a-1$  of part  $JK_j$  are completed but no parts have been produced in production run  $a$ . Therefore  $L_{aj}$  is the expected shortfall at demand point  $j$  given that production run  $a$  is still in progress at the demand point. The second term in the square brackets is the expected shortfall at demand point  $j$  given production runs 1, ...,  $a-1$  of part  $JK_j$  are completed but production run  $a$  has not yet started. When summed over all production runs in the set  $A_j$  and summed over all demand points  $j$ , this gives the total expected shortfall cost.

Provided that we can compute  $G_i(\cdot)$  and  $g_i(\cdot)$  without difficulty, the numerical challenge in computing the expected total cost from the above expression lies in computing the  $L_{aj}$ . This can be accomplished by numerical integration or by numerical Laplace transform inversion, as described in Chapter 2. Although our expression for total expected cost says that the number of  $L_{aj}$  integrals that we must compute is

$$\sum_{j=1}^M |A_j|,$$

there will typically be at most one production run intended to satisfy the demand at any one demand point. Therefore, the number of non-trivial  $L_{aj}$ 's that must be computed in practice is closer to  $M$ .

In summary, we have developed both an analytic and an algorithmic method for evaluating the cost of any particular production plan. These will be important “building blocks” as we explore this model further.

### 3.3 Deciding whether or not to run overtime

In this section we extend the model of the previous section to allow for one or more opportunities to run overtime between now and the end of the horizon. Although we will consider more complex extensions later, for now we extend the model of the previous section in the following way. At certain known points in time the decision maker has the option of purchasing a fixed size block of overtime at a fixed cost. Suppose there are  $N_{OT}$  such opportunities, where

$TO_p$  = time of the  $p^{\text{th}}$  opportunity to run overtime,

$OT_p$  = length of  $p^{\text{th}}$  overtime opportunity,

$co_p$  = cost of the  $p^{\text{th}}$  overtime opportunity,  $p = 1, \dots, N_{OT}$ .

We assume that the overtime opportunities are indexed such that  $a > b$  iff  $TO_a > TO_b$ .

The problem is to decide whether or not to run overtime to minimize expected stockout and overtime costs. Figure 3.5 is a modification of Figure 3.1 to account for overtime opportunities. We have assumed that there is an overtime opportunity of length 3 somewhere between  $t = 10$  and  $t = 16$ . As a result, the maximum output achievable over the interval if overtime is purchased is now  $6 + 3 = 9$ . Suppose for simplicity that the time axis and the  $y$ -axis are measured in the same units, and the discretization of the  $y$ -axis is chosen to be in unit increments. Then three additional transitions are possible if overtime is purchased; these are represented by dotted lines in Figure 3.5. Even if overtime is purchased, there is still some positive probability that there is no output over the interval.

In Figure 3.6 we show a modification of Figure 3.2 in which we have an overtime opportunity at  $t = 12$ . Since the time available for overtime is not represented on the time axis, the output achieved during overtime is seen as a vertical “jump” at  $t = 12$ , which we have represented with a dotted line. The size of this jump is a random variable of type  $F(t; OT_p)$  discussed in Chapter 2.

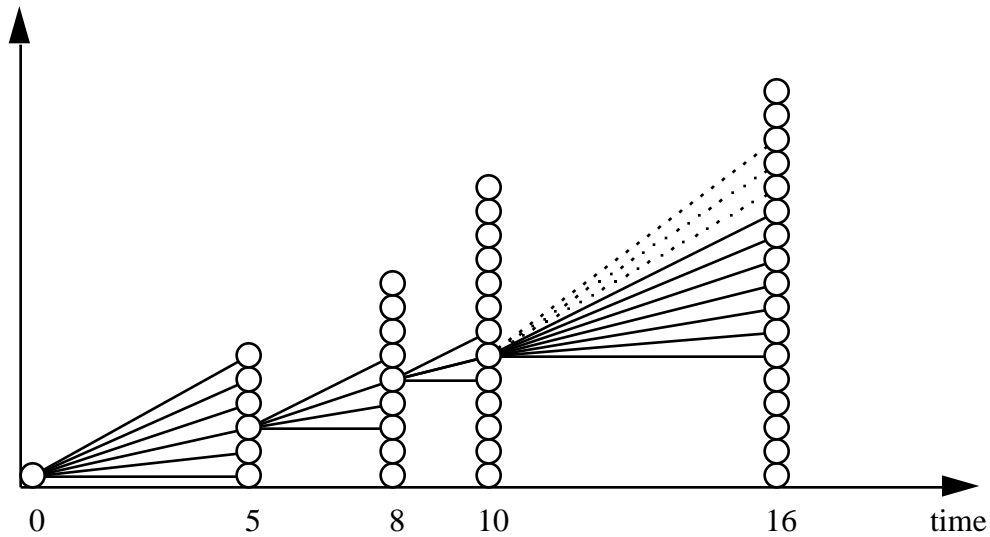


Figure 3.5 State space representation with overtime opportunity

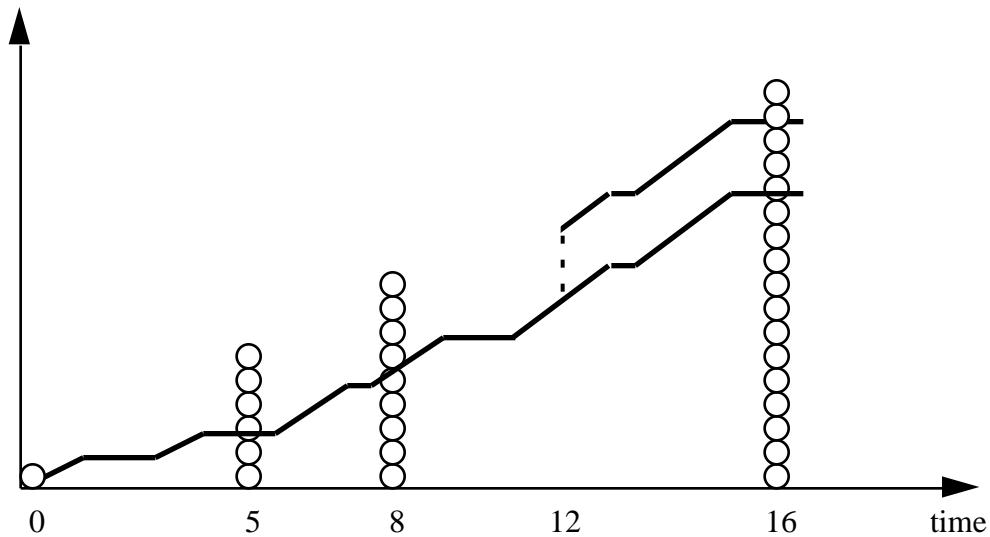


Figure 3.6 Realization of machine output under with and without overtime

Although we have shown only one overtime opportunity in these examples, we permit an arbitrary number of opportunities located anywhere within the horizon and of any cost.

### Dynamic programming formulation

We will show how to determine whether or not to run overtime at each opportunity by formulating the problem as a dynamic program. Dynamic programming is a well-established methodology for solving problems using digital computers (Bellman, 1957), which we now briefly introduce. The essential idea is to summarize the “state of the world” in one or more *state variables*, where each state has an associated cost (or benefit). At each *stage* of the dynamic program we are told the current state and may need to decide what *action* to take. Before we can find an optimal strategy we must write a recursive relationship to compute the cost of any action given the current state under the assumption that in the future we will behave in a cost minimizing manner. We now introduce some notation for the purposes of this discussion only. Let the stages be indexed by  $n$  in reverse chronological order,  $\mathbf{s}_n \in \mathbf{S}_n$  be the state at stage  $n$ ,  $\mathbf{a}_n \in \mathbf{A}_n$  be a vector describing the action taken at stage  $n$ , and  $c_n(\mathbf{s}_n)$  be the optimal expected cost to go with  $n$  stages remaining\* if the current state is  $\mathbf{s}_n$ . Then

$$c_n(\mathbf{s}_n) = \min_{\mathbf{a}_n \in \mathbf{A}_n} \left\{ \gamma c_n(\mathbf{s}_n, \mathbf{a}_n) + \sum_{\mathbf{s}_{n-1} \in \mathbf{S}_{n-1}} \Pr(\mathbf{s}_{n-1} | \mathbf{s}_n, \mathbf{a}_n, n) c_{n-1}(\mathbf{s}_{n-1}) \right\}$$

where  $\gamma$  is a discount rate ( $0 \leq \gamma < 1$ ),  $c_n(\mathbf{s}_n, \mathbf{a}_n)$  is the cost of being in state  $\mathbf{s}_n$  and taking action  $\mathbf{a}_n$  at stage  $n$ , and  $\Pr(\mathbf{s}_{n-1} | \mathbf{s}_n, \mathbf{a}_n, n)$  is the *transition probability* (the

---

\* We adopt the term “cost to go” from Bertsekas (1987). Other authors have called this the “remaining cost”.

probability that the next state is  $s_{n-1}$  given that the current stage is  $n$ , the current state is  $s_n$  and the action taken is  $a_n$ ). If the transition probabilities are known and the state and action spaces are finite, then it is a simple matter (at least in principle) to program a computer to find the optimal action for each stage and each possible state by solving the backwards recursion and computing the expected cost for each possible action at each stage. This brief overview of dynamic programming does not even scratch the surface of the well developed theory of the field. The interested reader is referred to one of many excellent introductory texts such as Bertsekas (1987), Denardo (1982) or Bellman and Dreyfus (1962).

We now describe our dynamic programming formulation. Although we will abandon the notation used above to describe dynamic programming, we now describe, in turn, the set of possible states  $S_n$ ; the stages  $1, \dots, n$ ; the set of permissible actions  $A_n$ ; the immediate cost function  $c_n(s_n, a_n)$ ; and the transition probabilities  $\Pr(s_{n-1} | s_n, a_n, n)$ .

### State space

The dynamic program that we construct will bear a close resemblance to the formulation in the previous section. The state space for the dynamic program will have two dimensions and , as described in the previous section. Recall that is a measure of cumulative output and is the (binary) state of the machine (working or failed).

### Stages

The  $M + N_{OT} + 2$  stages of the dynamic program will represent the beginning of the first production run at time zero, the  $M$  demand points, the  $N_{OT}$  points in time

where overtime can be purchased, and the end of the horizon. We index the stages in reverse chronological order.

### Actions and immediate costs

At the stages representing overtime opportunities, there are two possible actions: whether or not to purchase a fixed-size block of overtime. In this model, the decision maker will be motivated to purchase overtime if and only if it results in a reduction in total expected costs. At the stages representing overtime opportunities, the immediate costs are a function only of the action taken. If the  $p^{\text{th}}$  block of overtime is purchased, then we assess an immediate cost  $co_p$  at the stage. Otherwise there is no cost assessed.

At the stages representing the demand points, no decisions are made. The immediate costs at these stages are a function only of the state variable  $s$ , as described in the previous section.

A number of different assumptions could be made regarding the terminal costs, i.e., the immediate costs at the stage that represents the end of the horizon. One natural assumption would be to charge a penalty cost equal to the expected cost of the amount of overtime required to complete any unfinished portion of the production plan. The dynamic programming algorithm will be capable of handling any set of terminal costs, although logically speaking, one would expect that the terminal costs would be non-increasing as a function of  $s$ .

### Transition probabilities

The state transition probabilities to the next stage are also computed as in the previous section. If the  $p^{\text{th}}$  block of overtime is purchased, then we add  $OT_p$  to the

time available for production when evaluating the transition probabilities from  $(TO_p, \dots)$  to a state at the next stage.

### Optimization

The dynamic program is solved by backward recursion beginning with the second to last stage (closest to the end of horizon) and computing the expected cost to go for each possible value of the state variables and . At stages that represent overtime opportunities, the decision whether or not to run overtime in a particular state is determined by which choice results in the least expected cost to go. The recursion proceeds backwards until the first stage is reached, telling us the expected cost to go at time zero.

Typically, the optimal decisions at each overtime opportunity can be described by a pair of values\*. The larger of the two, which we will call the *critical overtime level*, is the value of above which it is not optimal to run overtime. Therefore, the critical overtime level at the  $p^{\text{th}}$  overtime opportunity is the largest value of at which the overtime cost  $co_p$  is exactly equal to the expected reduction in total cost to go if the overtime opportunity is purchased.

The fact that a critical overtime level exists is somewhat intuitive: as the value of decreases, we generally expect the benefit of overtime (in terms of reduced shortfall costs) to increase. Because we consider problems over a finite horizon, this need not be true. It could be that if we are so hopelessly behind schedule (i.e., at a very low value of ) that we will never catch up by the end of the horizon, so that stockouts

---

\* In Section 3.5 we describe one example we have found where this is not true. This example has extremely large MTBF and MTTR relative to the times between the stages.

are unavoidable even if overtime is purchased. In these cases, purchasing overtime results in a strictly negative expected benefit, since overtime costs are incurred yet there is little or no reduction in expected shortfall. If there exists a  $\alpha > 0$  such that it is *not* optimal to run overtime below this value of  $\alpha$ , then this is the second of the pair of values. We will call such a value the “lower envelope”.

Not running overtime when “hopelessly behind schedule” is clearly not a sensible action. In these situations one can conclude that the production plan is not realistic and should be reconsidered. In such situations, actions are often taken which can not (and we would argue should not) be modeled, such as re-negotiating deadlines or arranging for alternative sources of supply. Although the model can suggest a course of action which is not sensible, this should not be interpreted as an indication that the model is flawed, but rather that this model should not be applied in such a situation. Note that once the lower envelope is found, it is easy to superimpose these levels on the confidence interval for machine output (as shown in Figure 3.4) to ascertain the likelihood of falling “hopelessly behind schedule”.

### **Computational complexity**

The complexity of the dynamic programming algorithm that we have proposed can be determined in much the same way as the evaluative algorithm of the previous section. Let  $s$  denote the number of discretized values that  $x$  can take. Then at each of the  $M + N_{OT} + 1$  stages we must compute the expected cost to go for each possible state, of which there are  $O(s)$ . This requires computing  $O(s^2)$  transition probabilities if the machine reliability parameters are part dependent, and  $O(s)$  transition probabilities if not. Given the transition probabilities, the expected cost to go for any state is then found with a single vector multiplication requiring  $O(s)$  multiplications and additions. At the stages that represent overtime opportunities, we must do

twice the work, although this does not affect the computational complexity. In total, the algorithm requires  $O(s^2 (M + N_{OT}))$  multiplications and additions, the computation of  $O(s^2 (M + N_{OT}))$  transition probabilities if the machine reliability parameters are part dependent, and  $O(s (M + N_{OT}))$  transition probabilities if not. As before, the time to compute the transition probabilities will dictate the running time of the algorithm.

### **Empirical results**

We now present the results of some experiments performed using a computer program (written in Pascal and FORTRAN) that allows a user to perform numerical experiments with a variety of inputs\*.

The base case that we will consider is as follows. First we will assume that all parts to be produced have the same parameters (demand, machine reliability, etc.) This is not a necessary assumption of the model, it is made only for simplicity of this discussion. There are five parts to be produced once each over the horizon of length 1000 time units. The production quantity for each part is 120 units and the production rate for each part is one. There are five demand points, one for each of the five parts, for 60 units at intervals of 200 time units. The parts are produced in the order in which they are demanded. If we think of the time horizon as one week, this set of inputs corresponds to a production schedule in which we plan to produce each part every other week.

---

\* The percentiles of the cumulative output of the machine were obtained using Weeks' Method, and the state transition probabilities were obtained using Talbot's Method, as described in the Appendix to Chapter 2.

We set the machine reliability parameters  $1/\lambda$  (the mean time between failures, or MTBF) equal to 25, and  $1/\mu$  (the mean time to repair, or MTTR) equal to 15. Accordingly, the stand-alone availability (SAA) is  $MTBF / (MTBF + MTTR) = 0.625$ , and the (expected) utilization of the machine is  $(5 * 120 / 0.625) / 1000 = 96\%$ . There are five overtime opportunities of length 20, located 15 time units before each demand point. The costs are normalized so that the per unit backorder cost is always 1.0. The overtime cost is 0.33 per time unit. This data is summarized in Table 3.1.

In all experiments we will assume that setup times are zero. This is done only to make interpretation of the plots easier, and should not be interpreted as a change to the fundamental assumption that we are modeling a production line with non-trivial setup times and/or setup costs such that batching is a practical necessity.

In each experiment, the terminal costs are set to the expected cost of the amount of overtime required to complete any unfinished portion of the production plan. For the base case, the terminal costs are set to  $0.33 \times (5 \times 120 - ) / 0.625 + 0.33 \times (1 - ) / \mu$ . The first term is the expected cost of producing on overtime until  $= 5 \times 120$ , and the second term accounts for the expected cost of the additional overtime that is required if the machine must be repaired before production can resume.

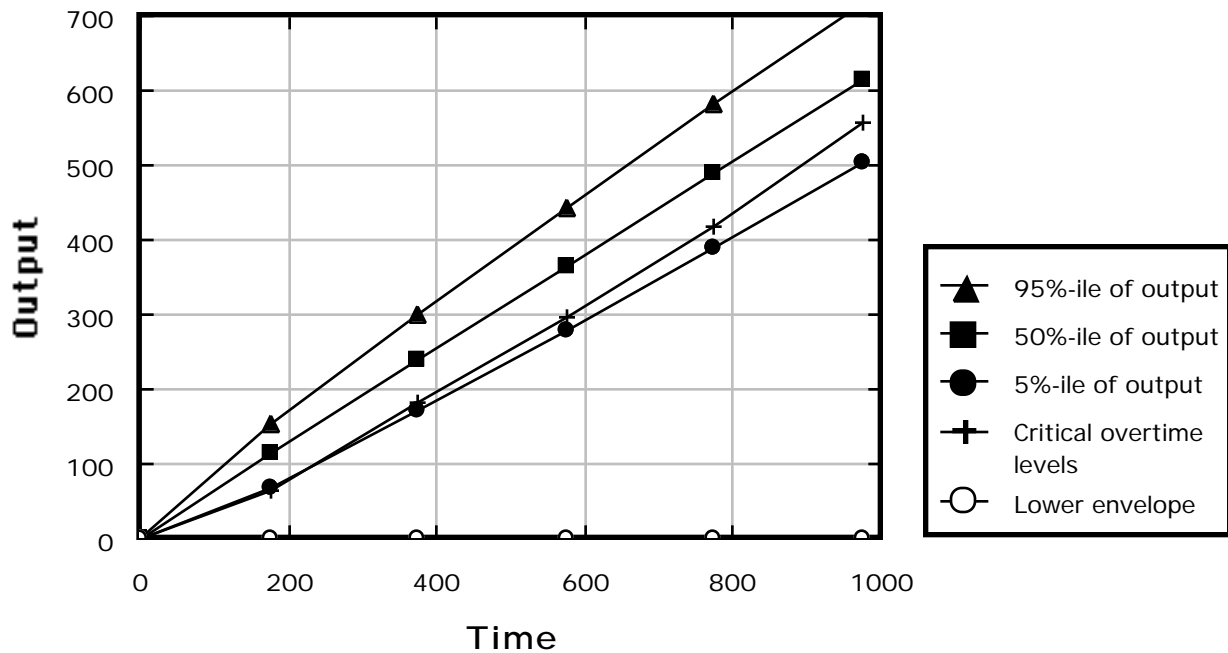
Figure 3.7 shows the confidence interval of machine output if the machine is working at time zero, and the critical overtime levels if the machine is working at the decision point. As in Figure 3.4, the triangles, squares, circles are the 95%, 50% and 5%-iles of the cumulative output of the machine as of time zero. The critical overtime levels are shown as plusses, and can be interpreted as the level above which it is never optimal to run overtime. There is also a “lower envelope” of

Demand points			Overtime Opportunities		
Part	Time	Quantity	#	Time	Length
1	200	60	1	175	20
2	400	60	2	375	20
3	600	60	3	575	20
4	800	60	4	775	20
5	1000	60	5	975	20

Horizon length = 1000

Production batch size = 120	MTBF = 25
Production rate = 1	MTTR = 15
Utilization = 96%	SAA = 62.5%
Backorder cost = 1	OT Cost = 0.33

**Table 3.1** Data for base case



**Figure 3.7** Critical overtime levels when machine is working. Base case. Cost to go = 9.6

overtime levels, represented by hollow circles, below which it is never optimal to run overtime. We see that the critical overtime levels are increasing and convex, indicating that, in this case, we become more willing to run overtime as we approach the end of the horizon. The lower envelope of overtime levels are all zero.

Figure 3.8 shows the confidence interval of machine output if the machine is failed at time zero, and the critical overtime levels if the machine is failed at the decision point. We first note that the cost to go as of time zero increases 29% from 9.6 to 12.4 if the machine is initially failed. We also observe that the critical overtime levels are generally higher at the decision points if the machine is down at that decision point. The critical overtime levels are 14% larger earlier in the horizon, but are 3% smaller at the last decision point. Further, the lower envelope takes on a positive value (440) at the last decision point. This is simply an end of horizon effect.

All experiments were performed on a Power Macintosh 7100/80 running in emulation mode with SANE-based math instructions. To create Figure 3.7, the 5%, 50% and 95%-iles of machine output were each evaluated at five points. These 15 points required a total of 43 seconds to compute. The dynamic program required 3 minutes and 7 seconds to solve with a discretization of the state variable of size 1, resulting in 700 possible discretized values of . The computational refinement to be described in Section 3.5 was implemented, although we did not exploit the fact that only two sets of transition probabilities need to be calculated since the time between any two stages is either 20 or 180. See Section 3.5 for a further discussion.

We now consider, in turn, a number of different changes to the base case. In the experiments that follow we show the critical overtime levels only for the case where

the machine is working. When appropriate we will comment if the effect of the change is substantively different if the machine is failed. In all of the experiments that we will describe, a two critical number policy was optimal.

The first change we consider is doubling the number of demand points while keeping the utilization of the machine constant. We accomplish this by creating a 10 part problem with 10 demand points at 100 time unit intervals. The lot size is halved to 60 and the demand at each demand point to 30. We also double the number of overtime opportunities to 10, still located 15 time units before each demand point, and halve the length of the opportunities to 10 time units. The net effect is that, because demand points occur more frequently, there is now significantly less opportunity to fall behind and still catch up before demand must be satisfied.

The resulting critical overtime levels are shown in Figure 3.9. The critical fractiles of machine output are unchanged from the base case. The cost to go increases substantially, as expected, since we have effectively placed additional “constraints” on the system. The critical numbers are also seen to significantly increase.

The above example has shown that it is desirable to have as much time as possible to “catch up” in the event that production falls behind. To demonstrate this principle further, we move the five demand points to the end of the horizon, at times (900, 925, 950, 975, 1000). We place the five overtime opportunities earlier in the horizon, at times (75, 275, 475, 675, 875). No other changes are made other than these timing changes.

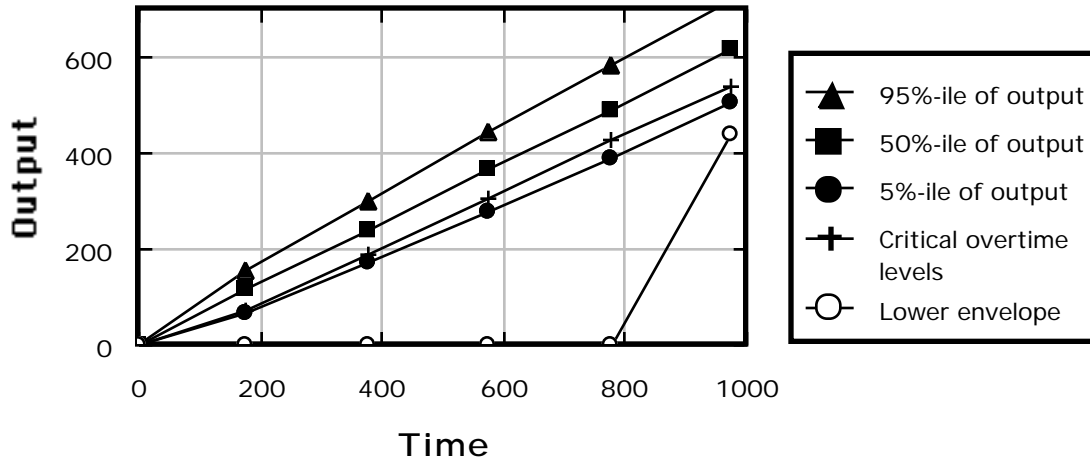


Figure 3.8 Critical overtime levels when machine is failed. Base case. Cost to go = 12.4

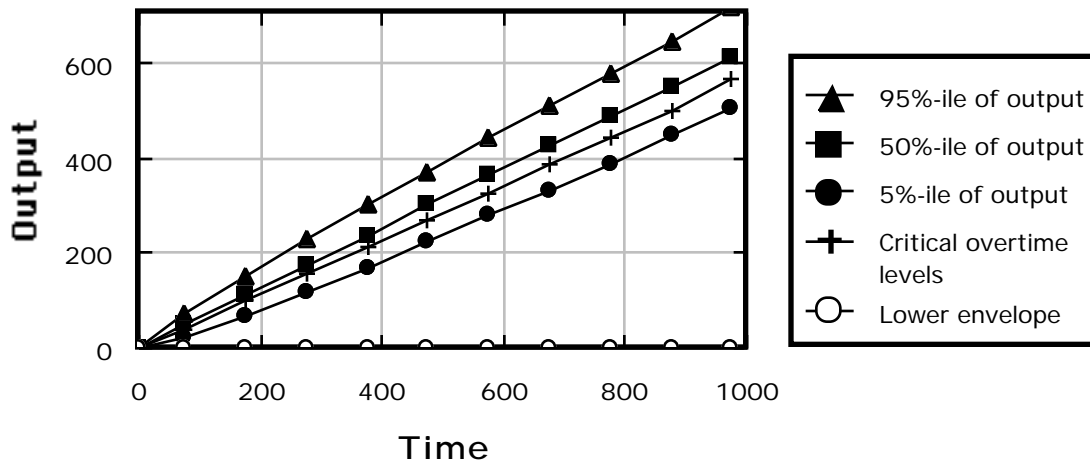


Figure 3.9 Critical overtime levels with ten demand points. Cost to go = 18.5

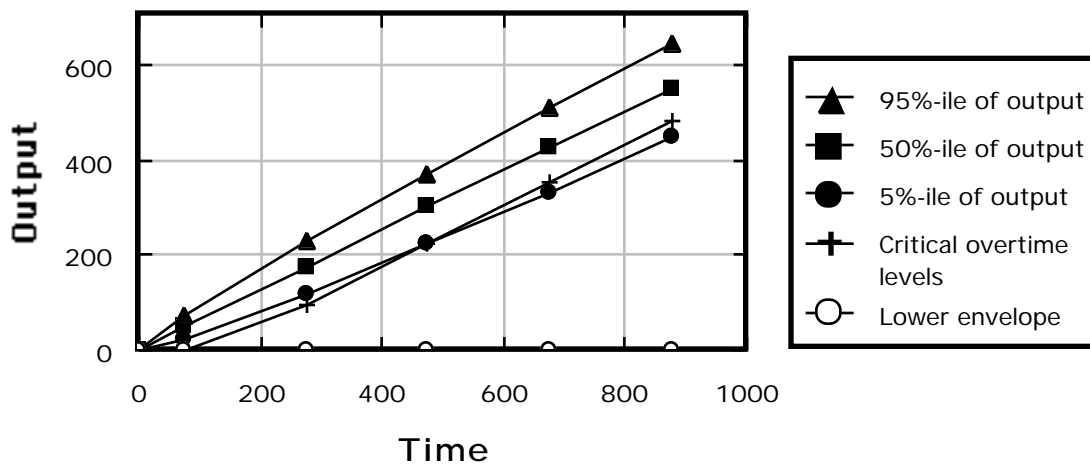


Figure 3.10 Critical overtime levels with demand at end of horizon. Cost to go = 8.2

The result of these changes is shown in Figure 3.10. First, we note that the cost to go decreases, so the timing changes are indeed beneficial. It is also interesting to note that we are not willing to run overtime at time 75, and that the critical overtime levels are convex increasing. This suggests a form of a wait-and-see strategy.

The final timing change that we consider is shifting the overtime opportunities farther away from demand points, to times (100, 300, 500, 700, 900). As a result, decisions must be made at an earlier time, that is, with less information. The resulting critical overtime levels are shown in Figure 3.11. The magnitude of the change is not large, but the direction of the effect is as we would expect. We note a slight increase in cost, and although it is difficult to see, the critical overtime levels fall substantially early in the horizon (by as much as 63%).

We now consider a variant of the last case, splitting each 20 time unit opportunity into two opportunities of length 10 time units. We place the opportunities at 100 time unit intervals starting at time 75. The net result of these changes is twofold. The first effect is as in the last case, where the decision to run overtime must be made earlier than in the base case. The second effect is the splitting of the opportunity, allowing a smaller sized block of overtime to be purchased. We know that the first effect causes an increase in cost to go and a decrease in the critical overtime levels. The result of these two effects taken together is shown in Figure 3.12. We see a net decrease in cost to go and a slight increase in the critical numbers. This suggests that the decision maker benefits from the added flexibility of smaller, more frequent overtime opportunities. Later in this chapter we will consider extensions to the model where the decision maker can choose the amount of overtime to purchase.

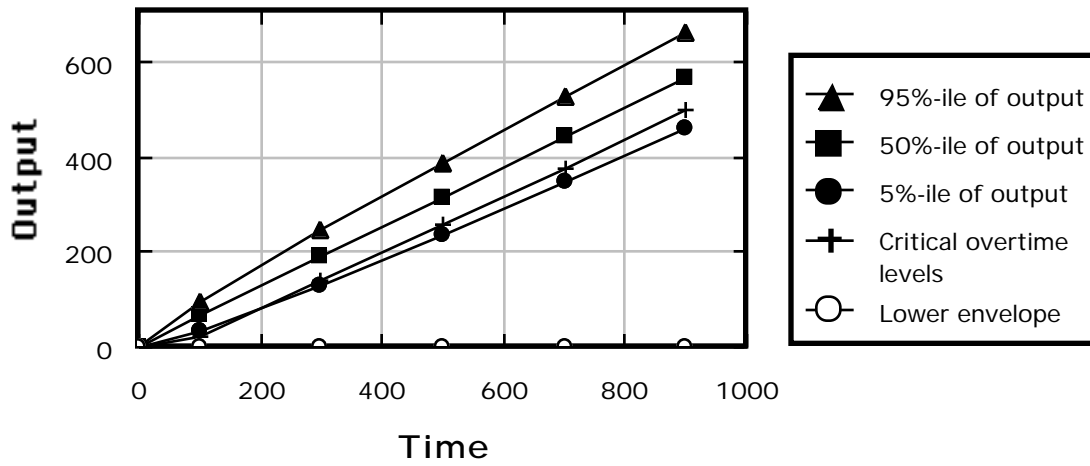


Figure 3.11 Critical overtime levels with opportunities moved up. Cost to go = 10.0

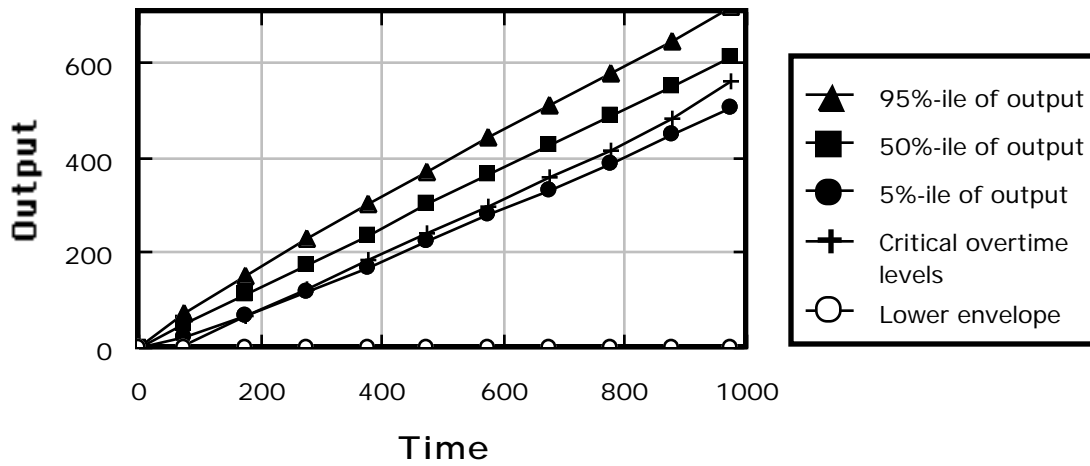
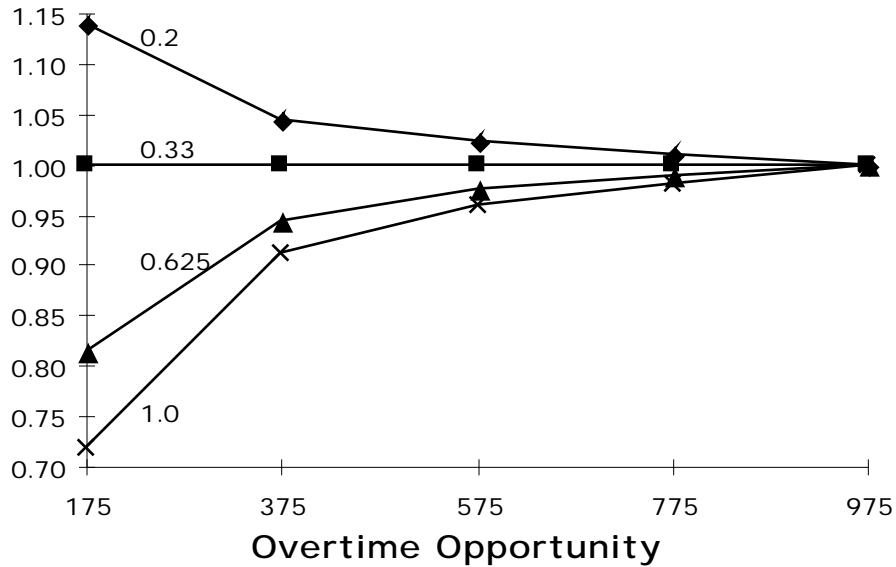


Figure 3.12 Critical overtime levels with ten opportunities. Cost to go = 9.3



**Figure 3.13** Normalized critical overtime levels with varying per unit overtime cost

We next look at the effect of varying the cost of one time unit of overtime. Four cost levels are considered: 0.33, the base case; 0.625, the cost at which the expected cost to produce a part on overtime is equal to the per unit shortage cost of 1.0; a low (0.2) and a high (1.0) case. Figure 3.13 shows the effect on the critical overtime levels. We see that increasing overtime costs decreases the critical numbers, and that the percent change decreases as we approach the end of the horizon.

We now turn our attention to changes in the machine reliability parameters MTBF and MTTR. We first decrease the MTTR to 7.5 and the MTBF to 12.5. These numbers are chosen such that the stand-alone availability (SAA) of the machine remains constant. The resulting confidence interval of machine output and critical overtime levels are plotted in Figure 3.14. We observe a considerable decrease in the width of the confidence interval, as we would expect since the MTTR has decreased; see the discussion in Chapter 2. Accordingly, there is a dramatic reduction in the cost to go. We also observe a reduction in the critical values,

particularly early in the horizon, due to the reduced variability (less uncertainty in the future).

In Figure 3.15 we show the result of increasing the MTTR to 30 and decreasing the MTBF to 50, again holding the SAA constant. This case is the exact opposite of the previous case. We observe a considerable increase in the width of the confidence interval and an equally substantial change in the cost to go. We also see a general pattern of increase in the critical values, particularly early in the horizon, due to the increased variability (more uncertainty in the future).

The next three charts show the impact of changes in machine utilization. We first decrease the utilization to 80% by decreasing the demand at each demand point to 50 and decreasing the lot sizes to 100. The results are shown in Figure 3.16. We see that this moderate reduction in utilization virtually eliminates the need for overtime: the critical overtime levels are all substantially less than the 5th percentile of the machine output distribution. Furthermore, cost to go has decreased to almost nothing.

In Figure 3.17 we plot the critical overtime levels for various machine utilizations (0.8, 0.9, 0.96, 1.0, 1.04) normalized such that the critical overtime levels for the base case (0.96) are 1.0. The different machine utilizations were achieved by scaling the size of the demand at each demand point, and keeping the ratio of demand to lot size at 1:2. We see that the critical numbers increase as the utilization of the machine increases. We see that the percent increase is greater for the earlier overtime opportunities, although we observe a significant difference between the critical numbers even at the last overtime opportunity in the horizon.

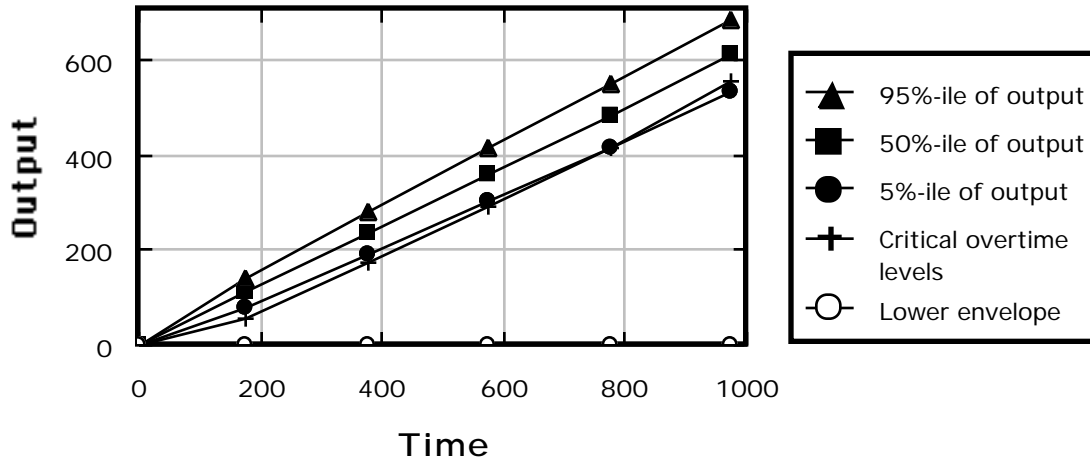


Figure 3.14 Critical overtime levels with MTTR decreased to 7.5. Cost to go = 4.0

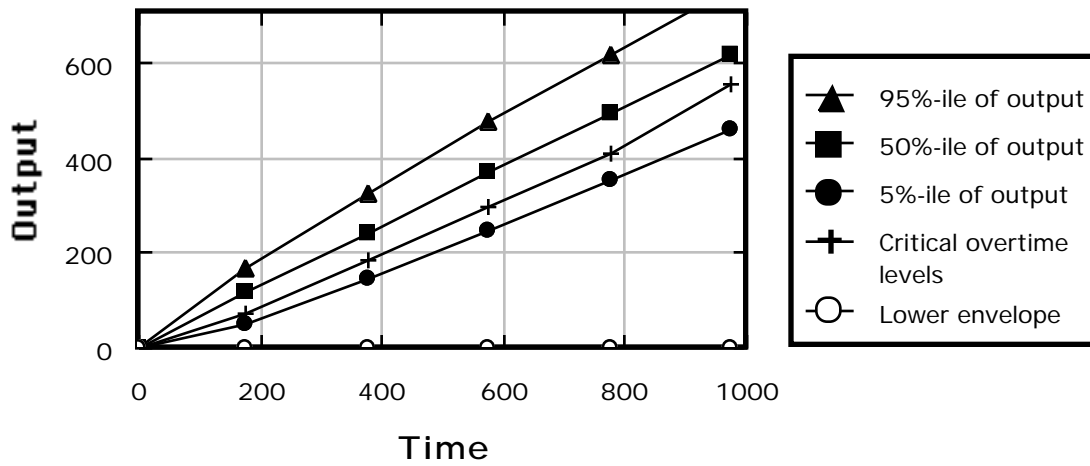


Figure 3.15 Critical overtime levels with MTTR increased to 30. Cost to go = 21.8

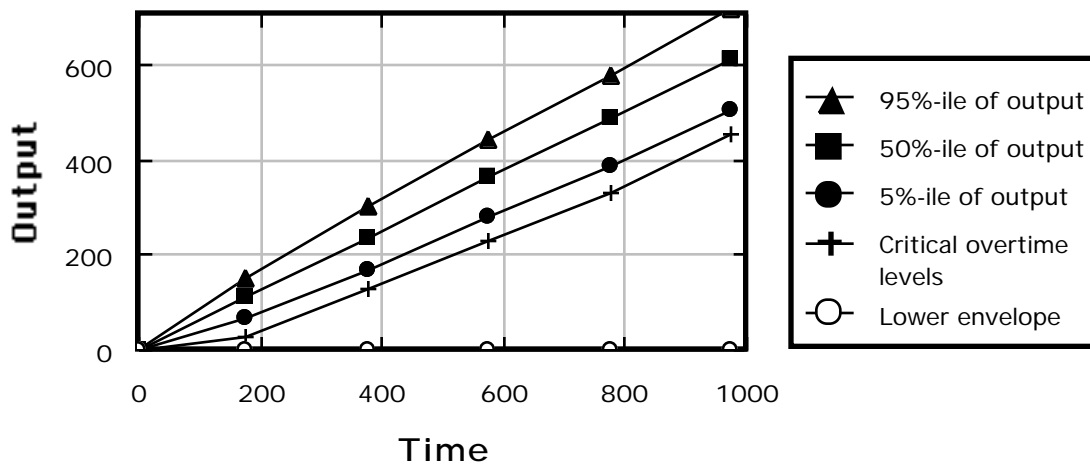
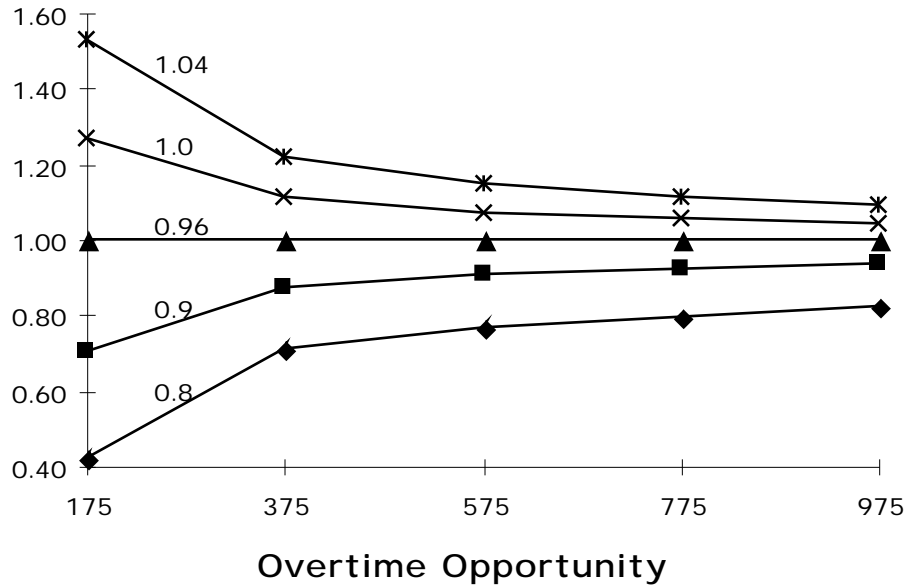
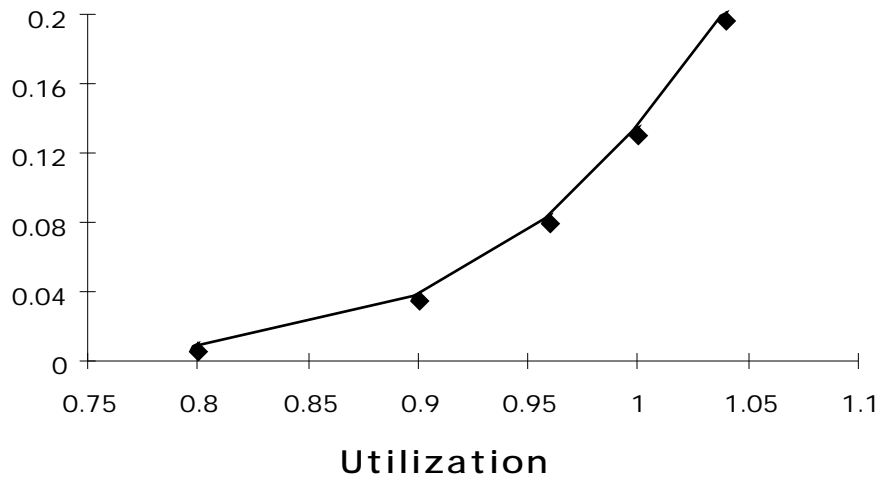


Figure 3.16 Critical overtime levels with utilization decreased to 80%. Cost to go = 0.6



**Figure 3.17** Normalized critical overtime levels with varying machine utilization

Figure 3.18 shows the expected cost to go per unit of total demand for various machine utilizations. We see that the expected cost to go per unit demand increases at a greater than linear rate in the region around full utilization. The point for utilization = 1.04 is not as high as might be initially expected. However, it is easy to see that as utilization is increased beyond 1.0, the probability that an additional unit of work will need to be produced on overtime rapidly approaches one. At utilizations beyond 1.0, the expected cost to go increases linearly (at the overtime cost rate). Thus, the superlinear increase in expected cost to go per part cannot be sustained.



**Figure 3.18** Expected cost per unit demand as a function of machine utilization

### 3.4 Properties of the dynamic programming solution

In this section we will prove certain important properties of the costs and the structure of the optimal policy of the dynamic program formulated in the previous section. This section can be omitted by the reader without loss of continuity.

It will be assumed throughout this section that machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts (i.e., the assumptions of Chapter 2 hold). Some additional notation is now introduced for the purposes of this section. First, let  $c_n(x, a; T)$  be the expected cost-to-go with  $n$  stages remaining if the current state is  $(x, a)$  and there are  $T$  time units available for production between stages  $n$  and  $n-1$ , and let  $c_{n-1}^*(x, a)$  be the optimal expected cost to go with  $n-1$  stages remaining if the current state is  $(x, a)$ . Then

$$c_n^*(x, a) = \min\{c_n(x, a; T), c_{op} + c_n(x, a; T+OT)\},$$

and

$$c_n(x, a; T) = c_n(x, a) + \int_{a=0}^1 \int_{x=0}^T c_{n-1}^*(x+x, a) \text{trans}(x, x; T | a) P_a(T) dx,$$

where  $\text{trans}(x_0, x_1; T | a)$  is the probability of transitioning from  $x_0$  to  $x_1$  in an interval of length  $T$ , conditional on transition from  $x_0$  to  $a$  in an interval of length  $T$ ;  $P_a(T)$  is the probability that the machine is in state  $a$  at time  $T$  if it is in state  $x_0$  at time zero; and  $c_n(x, a)$  is the immediate penalty cost function for stage  $n$ .

We will now show that the optimal expected cost-to-go at each stage as function of  $x$  is non-increasing.

**Theorem 1.** If  $c_{n-1}^*$  and  $c_n$  are continuous, non-negative and non-increasing as a function of  $x$ , then  $c_n$  is also a continuous, non-negative and non-increasing function of  $x$ .

**Proof.** Since  $c_n$ ,  $c_{n-1}^*$  and the transition probabilities are all non-negative,  $c_n$  is non-negative. Further,  $c_n$  is continuous because  $c_{n-1}$  and  $c_{n-1}^*$  are continuous. To show that  $c_n$  is non-increasing we must show that

$$-c_n(x, a; T) = -c_{n-1}(x) + \int_{a=0}^T \int_{x=0}^T \left\{ c_{n-1}^*(x+a, a) \text{trans}(x, x+a; T | a) P_a(T) \right\} dx$$

is non-positive.  $c_{n-1}(x)/c_{n-1}^*(x)$  is non-positive by assumption, so we need only to show that the integral is non-positive. Since machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, the above integral can be rewritten as

$$\int_{x=0}^T \frac{c_{n-1}^*(x+a)}{c_{n-1}^*(x)} f(x; T | a) P_a(T) dx,$$

where  $f(x; T | a) P_a(T)$  is the density of machine uptime as defined in Chapter 2. The integral is non-positive because  $c_{n-1}^*(x)/c_{n-1}^*(x+a)$  is non-positive, and  $f(x; T | a) P_a(T)$  is non-negative.

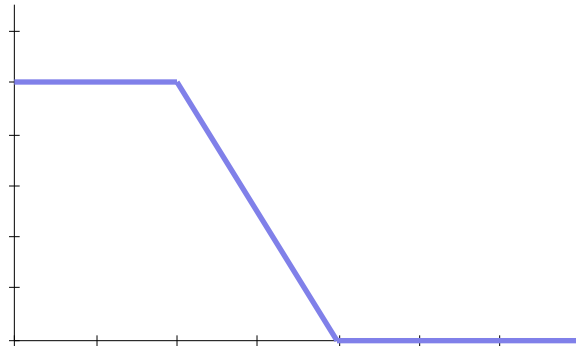
The argument if machine failures and repairs are not i.i.d. exponential or not the same across parts is more complex. However, it is difficult to imagine that this is

not true: if the expected cost to go  $c_n$  was not non-increasing then there would exist some value of  $x$  at which it is optimal to turn the machine off and stop producing. Since we know that there can not be a negative benefit to additional uptime, the expected cost to go can not increase when  $x$  is increased.

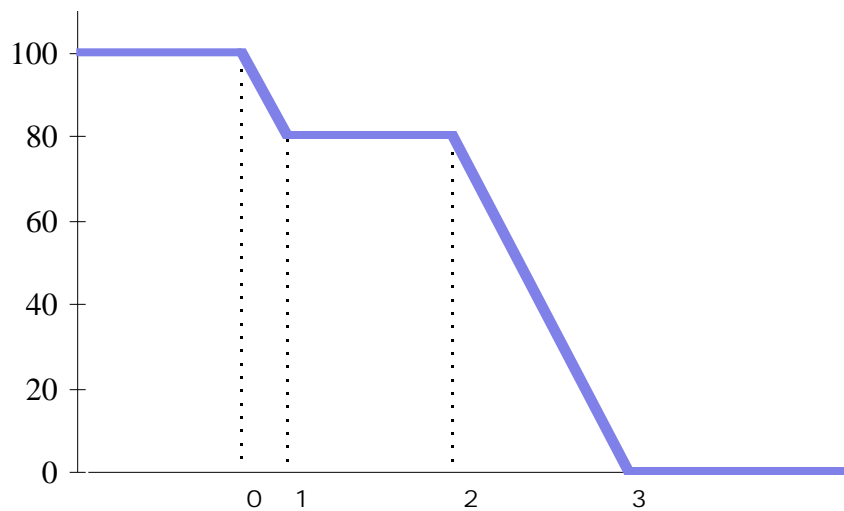
**Theorem 2.** If stage  $n$  represents overtime opportunity  $p$ ,  $co_p \geq 0$  and  $c_n(x, \tau; T)$  and  $c_n(x, \tau; T+OT)$  are continuous, non-negative and non-increasing as a function of  $x$ , then so is  $c_n^*(x, \tau) = \min\{c_n(x, \tau; T), co_p + c_n(x, \tau; T+OT)\}$ .

**Proof.** The minimum of two non-negative functions must also be non-negative. Similarly, the derivative of minimum of two functions with non-positive derivatives must also have a non-positive derivative. Note that where the derivative of the minimum does not exist, both one-sided derivatives are non-positive. Lastly, the minimum of two continuous functions is also continuous.

In Figure 3.19 we show our immediate penalty cost function  $c_n(x)$ . The function has this shape for the following reasons. If  $x$  is so low that we have not produced any parts to satisfy incremental demand at the demand point, then we will incur a penalty equal to the penalty cost rate times the quantity demanded. As  $x$  increases, this penalty will remain constant until  $x$  is such that we start producing the part that will satisfy demand at the demand point.  $c_n(x)$  then decreases at a linear rate as parts are produced, until the batch is completed or enough parts have been produced to satisfy all of the demand. As  $x$  increases beyond this point, the penalty cost does not decrease further. If there are multiple production runs for the part over the horizon, then there can be multiple regions of decrease separated by regions where the penalty cost is constant.



**Figure 3.19** Immediate penalty cost function



**Figure 3.20** Immediate penalty cost function with multiple production runs

To see this, let us consider a simple numerical example. Suppose there are two demand points of 100 units each for a part, two production runs of 100 units each, and suppose the initial inventory of the part is 20 units. If the per unit shortfall cost is one (for simplicity), then the penalty cost function at the second demand point might appear as shown in Figure 3.20. The value of  $t$  at which exactly 80 units of the first production run are completed is labeled as  $t_0$  in Figure 3.20. If by the time of the second demand point, at least 80 units of the first production run are not completed, none of the demand at second demand point will be satisfied, so the shortfall is 100

units. Between  $x_0$  and the point at which the first production run is completed (labeled as  $x_1$ ), the penalty cost function decreases linearly, since this production can be used to satisfy demand at the second demand point. From  $x_1$  until the second production run begins (labeled as  $x_2$ ), the shortfall is 80 units since no additional production occurs. Once the second production run begins, the penalty cost function decreases linearly until 80 units are produced, at which time all demand at the second demand point is satisfied. This point is labeled as  $x_3$ .

Irrespective of the number of production runs, the penalty cost function at any demand point is a continuous, non-negative and non-increasing function of  $x$ . As a result, if the terminal costs are continuous, non-negative and non-increasing, then by induction the optimal expected cost-to-go at each stage is a continuous, non-negative and non-increasing function of  $x$ .

The focus of the development that follows will be to characterize the form of the optimal overtime decisions. Before proceeding, we will require the following

**Definition 1.** A unidimensional function  $f$  is *weakly increasing\** if there exists some  $x_0$  such that  $f(x_0) = 0$ ,  $f(x) = 0$  for all  $x > x_0$ , and  $f(x) = 0$  for all  $x < x_0$ .

**Lemma 1.** If machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, then, ignoring impulses,  $f(t; T+OT | a) P_a(T+OT) - f(t; T | a) P_a(T)$  is a weakly increasing function of  $t$ .

---

\* The term weakly increasing and its definition are new.

The proof of this lemma is provided at the end of this section. We are now ready to state

**Theorem 3.**  $c_n(x, a; T)$  is a continuous and non-increasing function of  $T$  if machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts.

**Proof.** The portion of  $c_n(x, a; T)$  that is a function of  $T$  is the integral

$$c_n(x, a; T) = \int_{x=0}^T c_{n-1}^*(x + a, a) f(x; T | a) P_a(T) dx.$$

For any positive value of  $OT$ , consider the difference  $c_n(x, a; T+OT) - c_n(x, a; T)$ . This can be written as

$$\int_{x=0}^{T+OT} c_{n-1}^*(x + a, a) [f(x; T+OT | a) P_a(T+OT) - f(x; T | a) P_a(T)] dx.$$

This integral is a zero-sum weighted average, since

$$\int_{x=0}^{T+OT} [f(x; T+OT | a) P_a(T+OT) - f(x; T | a) P_a(T)] dx = 0.$$

From Theorem 2,  $c_{n-1}^*(x + a, a)$  is a non-increasing function of  $x$ . From Lemma 1, we know the expression in brackets is weakly increasing. Thus, our zero-sum weighted average gives negative weight to larger values of  $c_{n-1}^*(x + a, a)$  and positive weight to smaller values of  $c_{n-1}^*(x + a, a)$  (since  $c_{n-1}^*(x + a, a)$  is non-increasing). Therefore, the zero-sum weighted average of  $c_{n-1}^*(x + a, a)$  must be negative, so  $c_n(x, a; T)$  is a non-

increasing function of  $T$ . The continuity of  $c_n(x, a; T)$  with respect to  $T$  follows from the continuity of  $f(x; T | a) P_a(T)$  with respect to  $T$ .

To facilitate a discussion of the optimal overtime decisions, we first require

**Definition 2.** If stage  $n$  represents overtime opportunity  $p$ , then the largest value of  $x$  such that  $c_n(x, a; T) = co_p + c_n(x, a; T+OT)$  is the *critical overtime level* for a given  $a$  at stage  $n$ . Similarly, the smallest value of  $x$  such that  $c_n(x, a; T) = co_p + c_n(x, a; T+OT)$  is the *lower envelope* for a given  $a$  at stage  $n$ .

Note that the critical overtime level and lower envelope need not exist. We can now state

**Theorem 4.** If both the critical overtime level and lower envelope exist, then the optimal policy will not purchase overtime for any value of  $x$  above the critical overtime level, or any value of  $x$  below the lower envelope.

**Proof.** We know that for sufficiently large  $x$ ,  $c_n(x, a; T)$  and  $c_n(x, a; T+OT)$  are zero since no stockouts will occur and thus there will be no penalty costs. As a result, for sufficiently large  $x$ , the difference between the cost of purchasing overtime and not purchasing overtime is  $co_p + c_n(x, a; T+OT) - c_n(x, a; T) = co_p$ . By definition of the critical overtime level, we know that the costs are equal at that point, and are not equal again. Since the cost of purchasing overtime is eventually greater (by an amount  $co_p$ ), it must be that the cost of not purchasing overtime is less for all  $x$  greater than the critical overtime level. Using analogous logic, one can prove the result for the lower envelope.

For most cases, the optimal policy is a two critical number policy: run overtime if and only if the value of  $x$  is between the lower envelope and the critical overtime level. This need not always be true, and extreme cases can be constructed where it is not true. These cases can occur when the lower envelope conditional on  $a = 0$  is larger than the critical overtime level conditional on  $a = 1$ . For example, suppose that the failure and repair rates are so low that the probability that the machine fails twice between two stages is very small. In this situation there could be a value of  $x$  between the two critical numbers such that if the machine fails over the upcoming interval, it is not optimal to run overtime because no demand will be satisfied even with the additional overtime, and if that the machine does not fail over the interval, it is not optimal to run overtime because all demand will most likely be satisfied.

Based on the above reasoning, we suspected such a case might occur with very high MTTR and low SAA, and overtime opportunities that are small relative to the size of the demands. To test this hypothesis, the computer program described in Section 3.3 was used to find the optimal overtime decisions. The parameter settings used were  $MTTR = MTBF = 100$  ( $SAA = 50\%$ ), with demand of 25 parts at intervals of 100 time units, and overtime opportunities of length 10 time units, placed 15 time units before each demand point. The critical overtime levels and lower envelope for these parameters are shown in Figure 3.21. A two critical number policy is not optimal for the ninth overtime opportunity if the machine is failed at the decision point. The critical overtime level is at  $x = 249$ , and the lower envelope is at  $x = 137$ . However, it is not optimal to run overtime between  $x = 201$  and  $x = 214$ . As the MTBF and MTTR are increased further, other decision points lose their two critical number structure.

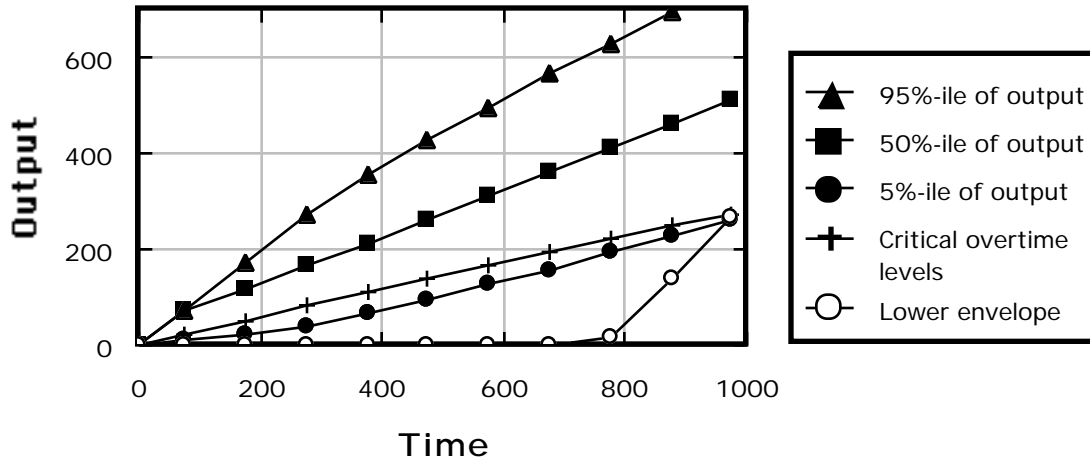


Figure 3.21 Critical overtime levels when machine is failed. Two critical number policy not optimal.

**Theorem 5.** If it exists, the critical overtime level at overtime opportunity  $p$  is non-increasing as a function of  $co_p$ , the cost of the overtime block. Similarly, the lower envelope at overtime opportunity  $p$  is non-decreasing as a function of  $co_p$ .

**Proof.** We have already established that  $c_n(\cdot, \cdot; T)$  is a non-increasing function of  $T$ , so that  $c_n(\cdot, \cdot; T) \geq c_n(\cdot, \cdot; T+OT)$  and thus  $\Delta(\cdot) = co_p + c_n(\cdot, \cdot; T+OT) - c_n(\cdot, \cdot; T) \leq co_p$ . Furthermore,  $\Delta(\cdot)$  is equal to  $co_p$  for sufficiently large  $\cdot$ , and also for sufficiently small  $\cdot$ . Therefore,  $\Delta(\cdot)$  is initially decreasing. We expect  $\Delta(\cdot)$  to appear something like that shown in Figure 3.22. Such a function will intersect the x-axis in an even number of places, or not at all. The points at which the function crosses the axis are the critical numbers. We now show that the rightmost (leftmost) point at which this function first crosses the x-axis is therefore non-increasing (non-decreasing) as a function of  $co_p$ .

Let  $L(y)$  be the set  $\{ \cdot : \Delta(\cdot) = y \}$ . Let  $L^-(y)$  ( $L^+(y)$ ) be the smallest (largest) element in the set  $L(y)$ . Since  $\Delta(\cdot)$  is initially decreasing,  $L^-(y)$  is initially increasing as  $y$  decreases. Since  $\Delta(\cdot)$  must eventually increase back to  $co_p$ , at some point  $\Delta(\cdot)$  will reach a local

minimum; let us call the value of  $x$  at which this happens  $x_L$ . After  $x_L$ ,  $f(x)$  will start to increase, but these values of  $x$  cannot be part of the set  $L(y)$  since these  $y$  values were achieved at lower values of  $x$ . At some point  $f(x)$  may reach a local maximum and then start to decrease again, as shown in Figure 3.22. If  $y$  decreases lower than  $f(x_L)$ , then while  $f(x)$  decreases, these values of  $x$  will be part of the set  $L(y)$  until another local minimum is reached. Irrespective of the number of local minima and maxima,  $L(y)$  is increasing in  $y$ . We can analogously show that  $L^+(y)$  is decreasing in  $y$ . Note that an increase (decrease) in  $c_{op}$  effectively shifts the entire function  $f(x)$  upward (downward) relative to the  $x$ -axis. Thus  $L(y)$  and  $L^+(y)$  give the lower envelope and critical overtime levels, and the result is proven.

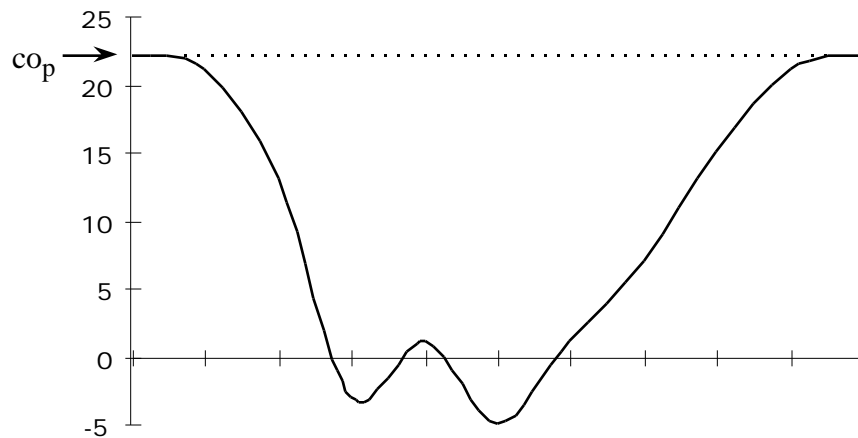


Figure 3.22 Example of  $f(x)$ , the increased cost as a result of purchasing overtime

As promised, we conclude this section with the following

**Lemma 1.** If machine failures are i.i.d. exponential and the same across parts, and repairs are i.i.d. exponential and the same across parts, then, ignoring impulses,  $f(t; T+OT | a) P_a(T+OT) - f(t; T | a) P_a(T)$  is a weakly increasing function of  $t$ .

Our proof will require the following two definitions.

**Definition 2.** A differentiable uni-dimensional function  $f$  is *strictly pseudoconvex* over  $[a,b]$  if and only if for  $x_1, x_2 \in [a,b], x_1 < x_2$ , such that  $f(x_1) = f(x_2)$ , the following two conditions are satisfied:

- (i)  $df(x_1)/dt < 0$  if  $x_2 > x_1$ ,
- (ii)  $df(x_1)/dt > 0$  if  $x_1 > x_2$ .

**Definition 3.** A uni-dimensional function  $f$  is *strictly pseudoconcave* over  $[a,b]$  if  $-f$  is strictly pseudoconvex over  $[a,b]$ .

**Proof of Lemma 1.** We will use the notation of the uptime densities derived in Chapter 2. We begin with the case  $\mu = 1, a = 0$ . After simplification, the difference is

$$f(t; T+OT | 10) P_{10}(T+OT) - f(t; T | 10) P_{10}(T) = e^{-t-\mu(T-t)} \left[ e^{-\mu OT} I_0(2\sqrt{x + \mu t OT}) - I_0(2\sqrt{x}) \right],$$

where

$$x = \mu t (T-t).$$

This expression is valid for  $0 \leq t \leq T$ . For  $T < t \leq T+OT$ ,  $P_{10}(T) f(t; T | 10)$  is zero and  $P_{10}(T+OT) f(t; T+OT | 10)$  is positive so the difference is positive. For the difference to be weakly increasing we must show that there exists some  $a, 0 \leq a \leq T$ , such that the difference is non-positive for  $0 \leq t \leq a$ , zero at  $t = a$ , and non-negative for  $a \leq t \leq T$ .

Since  $e^{-t-\mu(T-t)}$  is non-negative for all  $t$ , it does not affect the sign of  $P_{10}(T+OT) f(t; T+OT | 10) - P_{10}(T) f(t; T | 10)$ . We will therefore limit our attention to the expression in square brackets since it determines the sign, which we can rewrite as

$$f(t) = k I_0\left(2\sqrt{\mu t(T-t) + \mu tOT}\right) - I_0\left(2\sqrt{\mu t(T-t)}\right)$$

where

$$k = e^{-\mu OT}.$$

We now note a number of properties of  $f(t)$  that will be of subsequent importance.

Let  $f_1 = k I_0(2\sqrt{\mu t(T-t) + \mu tOT})$  and let  $f_2 = I_0(2\sqrt{\mu t(T-t)})$ .

i)  $0 < k < 1$ .

ii)  $I_0(t)$  is non-negative, convex and strictly increasing for  $t \geq 0$ , which follows immediately from its first and second derivatives (Abramowitz and Stegun, 1964).

iii)  $f_2$  is strictly increasing as a function of  $t$  up to  $T/2$ , strictly decreasing after  $T/2$ , and symmetric about  $T/2$  over  $[0, T/2]$ .  $f_1$  is strictly increasing as a function of  $t$  up to  $(T+OT)/2$ , strictly decreasing after  $(T+OT)/2$ , and symmetric about  $(T+OT)/2$  over  $[0, (T+OT)/2]$ . These properties follow from (ii).

iv)  $f_2$  is strictly pseudoconcave over  $[0, T]$  and  $f_1$  is strictly pseudoconcave over  $[0, T+OT]$ . This follows from (ii) and (iii).

v) At  $t = 0$ ,  $f(0) < 0$  since  $0 < k < 1$  from (i).

Since  $f_1(0) < 0$ , we must show that  $f_1(t)$  crosses the x-axis at most once. Let  $t_0$  be the smallest value of  $t$  at which  $f_1(t) = 0$ ,  $0 < t_0 < T$ . We now show that  $f_1(t) > 0$  for  $t > t_0$ , i.e.,  $f_1(t)$  not cross the axis again. We consider two separate cases.

First suppose  $t_0 > T/2$ . This implies that  $f_1(T/2) < 0$ . An example of this case is shown in Figure 3.23. By definition of  $t_0$ ,  $f_1 < f_2$  for  $t < t_0$ . This implies that  $f_1$  increases more slowly up to  $T/2$ . Therefore, due to the symmetry of each function,  $f_1$  decreases more slowly. Because of the symmetry and strict pseudoconcavity of each function, and since  $f_1$  decreases more slowly it must be greater than  $f_2$  for  $t > t_0$ .

Now consider the case  $t_0 < T/2$ . An example of this case is shown in Figure 3.24. Since  $I_0(t)$  is convex increasing, the difference  $k [I_0(2\sqrt{\mu t(T-t)} + \mu tOT) - I_0(2\sqrt{\mu t(T-t)})]$  is increasing. Since it is zero at  $t = t_0$ , it must be positive for  $t > t_0$ . We can therefore conclude that  $f_1(T/2) > 0$ . Since the mode of  $f_1$  is greater and to the right of the mode of  $f_2$ ,  $f_1$  must remain to the right (and therefore, above)  $f_2$  due to the symmetry and strict pseudoconcavity of the two functions. Thus  $f_1 > f_2$  for  $t > t_0$ .

We have assumed that  $f_1 < f_2$  for  $t < t_0$ , and shown that  $f_1 > f_2$  for  $t > t_0$ . Therefore  $P_{10}(T+OT) f(t; T+OT | 10) - P_{10}(T) f(t; T | 10)$  must be weakly increasing. The proof for the case  $\mu = 0$ ,  $a = 1$  follows from the same arguments since  $f(t; T | 10)$  and  $f(t; T | 01)$  are nearly identical.

The cases  $\mu = 1$ ,  $a = 1$  and  $\mu = 0$ ,  $a = 0$  are also very similar. We can write the two differences as

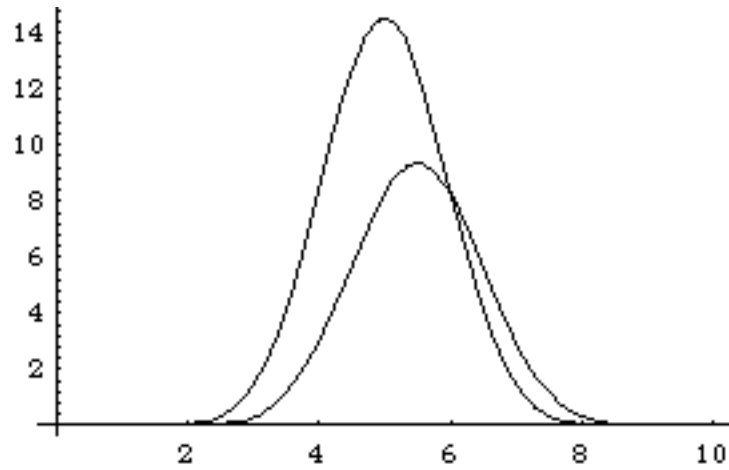


Figure 3.23 Example of case  $\mu > T/2$

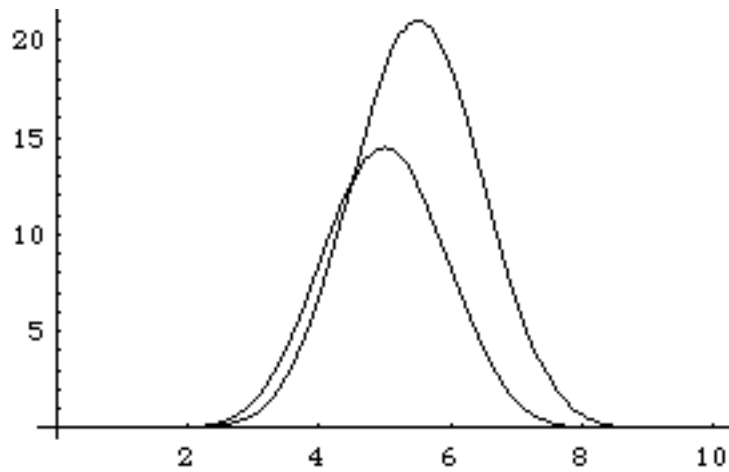


Figure 3.24 Example of case  $\mu = T/2$

$$f(t; T+OT | 11) P_{11}(T+OT) - f(t; T | 11) P_{11}(T) =$$

$$\mu e^{-t-\mu(T-t)} e^{-\mu OT} t \frac{I_1(2\sqrt{x + \mu t OT})}{\sqrt{x + \mu t OT}} - t \frac{I_1(2\sqrt{x})}{\sqrt{x}}$$

$$+ u_0(T+OT-t)e^{-(T+OT)} - u_0(T-t)e^{-T},$$

$$f(t; T+OT | 00) P_{00}(T+OT) - f(t; T | 00) P_{00}(T) =$$

$$\mu e^{-t-\mu(T-t)} e^{-\mu OT} (T+OT-t) \frac{I_1(2\sqrt{x + \mu t OT})}{\sqrt{x + \mu t OT}} - (T-t) \frac{I_1(2\sqrt{x})}{\sqrt{x}}$$

$$+ u_0(t)e^{-\mu(T+OT)} - u_0(t)e^{-\mu T}.$$

As before, the quantities that multiply the expressions in square brackets are non-negative and therefore can be ignored. We rewrite the two expressions in square brackets as

$$k_{11}(t) = k_{11} t \frac{I_1(2\sqrt{x + \mu tOT})}{\sqrt{x + \mu tOT}} - t \frac{I_1(2\sqrt{x})}{\sqrt{x}}$$

and

$$k_{00}(t) = k_{00} (T + OT - t) \frac{I_1(2\sqrt{x + \mu tOT})}{\sqrt{x + \mu tOT}} - (T - t) \frac{I_1(2\sqrt{x})}{\sqrt{x}}$$

where

$$k_{11} = k_{00} = e^{-\mu OT}.$$

We can now see that  $k_{11}(t)$  and  $k_{00}(t)$  have the same characteristics as  $f_1(t)$ . First,  $k_{11}$  and  $k_{00}$  are both positive and less than one. Further, it is easy to show that  $k_{11} t I_1(2\sqrt{\mu t(T-t) + \mu tOT}) / \sqrt{\mu t(T-t) + \mu tOT}$  and  $t I_1(2\sqrt{\mu t(T-t)}) / \sqrt{\mu t(T-t)}$  share the same properties as  $f_1$  and  $f_2$  described above with the exception of property (v). The same is true for  $k_{00} (T+OT-t) I_1(2\sqrt{\mu t(T-t) + \mu tOT}) / \sqrt{\mu t(T-t) + \mu tOT}$  and  $(T-t) I_1(2\sqrt{\mu t(T-t)}) / \sqrt{\mu t(T-t)}$ .

Property (v) deals with the behavior of the difference at  $t = 0$ . In this case, the differences are zero at  $t = 0$ . However, we know that

$$I_1(t) \sim \frac{1}{2}t \quad (\text{for } t \text{ small, } t > 0)$$

from Equation 9.6.7 of Abramowitz and Stegun (1964), so

$$t \frac{I_1(2\sqrt{x})}{\sqrt{x}} \sim t \quad (\text{for } t \text{ small, } t > 0).$$

This means that for  $t$  small,  $f_{11}(t)$  is approximately  $k_{11} t - t$  which is negative since  $0 < k_{11} < 1$ .

The case  $\alpha = 0, a = 0$  does not exhibit a strict property of this type. In this case  $f_{00}(t)$  is approximately  $k_{11} (T+OT-t) - (T-t)$ . If  $OT > (1-k_{00}) T / k_{00}$  then  $f_{00}(t) > 0$  for sufficiently small positive  $t$ . However, if  $f_{00}(t)$  is positive for all positive  $t$  in a neighborhood of zero, then  $f_{00}(t) > 0$  for  $0 < t < T/2$ . This follows from the fact that  $I_1(t)/t$  is increasing and convex in  $t$ , so  $f_{00}(t)$  can be seen to be increasing. Therefore, this is a special case of  $t < T/2$ , where  $\alpha = 0$ . The argument is therefore unchanged, except that  $f_{00}(t) > 0$  for  $0 < t < T$ . This completes the proof for all four cases.

Although we have proven that  $f(t; T+OT | a) P_a(T+OT) - f(t; T | a) P_a(T)$  is weakly increasing in  $t$  only for the case of i.i.d. exponential repairs and i.i.d. exponential failures, we expect this result to hold for a much broader class of failure and repair distributions. For example, we expect that if the uptime distribution is Normal with mean and variance proportional to  $T$ , then the result would still hold. This conjecture is based on the fact that  $f(t; T | a)$  is asymptotically Normal in  $T$  (Takács 1957a, Takács 1957b).

### 3.5 A computational refinement

In this section we describe a method that can be used to reduce the computational effort of the dynamic program when the reliability of the machine (in terms of failure and repair rates) is the same across all parts. This section can be omitted by the reader without loss of continuity.

Recall from Section 3.2 that the dynamic programming algorithm we have described requires  $O(s^2 M)$  multiplications and additions, and the computation of  $O(s^2 M)$  transition probabilities. Since the determination of the transition probabilities requires significantly more computational effort than vector multiplication (see the Appendix to Chapter 2), the time to compute the transition probabilities will dictate the running time of the algorithm.

The key observation is to recognize that when the reliability of the machine (in terms of failure and repair rates) is the same across all parts, a single transition probability can be reused several times. We now describe this in detail.

Suppose we wish to compute the expected cost of some state  $(t_1, i_1, j_1)$  and wish to compute the expected cost

$$c_n(x, a; T) = c_n(x, a) + \int_{a=0}^1 \int_{x=0}^T c_{n-1}^*(x+x, a) \text{trans}(x, x; T | a) P_a(T) dx$$

where, as in the previous section,  $c_n(x, a; T)$  is the expected cost-to-go with  $n$  stages remaining if the current state is  $(x, a)$  and there are  $T$  time units available for production between stages  $n$  and  $n-1$ ;  $c_{n-1}^*(x, a)$  is the optimal cost to go with  $n-1$

stages remaining if the current state is  $(i, a)$ ;  $\text{trans}(i_0, i_1; T | a)$  is the probability of transitioning from  $i_0$  to  $i_1$  in an interval of length  $T$ , conditional on transition from  $i_0$  to  $a$  in an interval of length  $T$ ;  $P_a(T)$  is the probability that the machine is in state  $a$  at time  $T$  if it is in state  $i_0$  at time zero; and  $c_n(\cdot)$  is the immediate penalty cost function for stage  $n$ .

If the machine reliability is the same across parts, we can rewrite  $\text{trans}(i, i+x; T | a)$  as  $f(x; T | a)$  since a transition from  $i$  to  $i+x$  units implies the same amount of uptime under the same failure process irrespective of the value of  $i$ , assuming that there is not a machine changeover between  $i$  and  $i+x$ . If a transition from  $i$  to  $i+x$  implies  $s$  units of setup time (and therefore  $x-s$  units of machine uptime), then we can rewrite  $\text{trans}(i, i+x; T | a)$  as  $f(x-s; T-s | a)$ .

Let us consider a simple numerical example to illustrate how transition probabilities can be reused. Suppose the machine is currently set up to produce part 1, and we plan to produce 100 units of part 1, incur a 10 minute changeover, then produce 100 units of part 2. For simplicity, suppose the production rate is one part per minute. Lastly, suppose  $T$ , the time between the current stage and the next, is 60 minutes. Note that  $\text{trans}(i, i+x; 60 | a) = \text{trans}(0, x; 60 | a)$  for  $0 \leq x \leq 40$ . Note also that  $\text{trans}(i, i+x; 60 | a) = \text{trans}(110, 110+x; 60 | a)$  for  $110 \leq i < 210$ , and further that  $\text{trans}(110, 110+x; 60 | a) = \text{trans}(0, x; 60 | a)$  for  $110 \leq i < 210$ .

For  $40 < i < 110$ , it is possible that in the time between the two stages, production of the first part is completed and a changeover occurs. If  $i+x \leq 100$  then no changeover occurs, so  $\text{trans}(i, i+x; 60 | a) = \text{trans}(0, x; 60 | a)$ . If  $i \geq 100$  and  $i+x \leq 110$  then the changeover is started and completed, so  $\text{trans}(i, i+x; 60-10 | a) = \text{trans}(110-x, 110; 60-10 | a)$ . If  $i \geq 100$  and  $100 < i+x < 110$  then the changeover is

started but not completed, and in this case each transition probability will be unique. However, these transition probabilities can be reused when  $100 < x < 110$  by noting that  $110 - x$  is the number of minutes of the changeover that are completed, so that  $\text{trans}(x, x+x; 60-(110-x) | a) = \text{trans}(100+(110-x)-x, 100+(110-x); 60-(110-x) | a)$ .

In total, we must compute  $\text{trans}(0, x; 60 | a)$  at  $0 \leq x \leq 60$  (61 values),  $\text{trans}(110-x, 110; 60-10 | a)$  at  $0 \leq x \leq 50$  (51 values), and  $\text{trans}(x, x+x; 60 | a)$  at  $40 < x < 100$  and  $100 < x+x < 110$  ( $59 + 58 + \dots + 50 = 545$  values), for a total of 657 transition probabilities. This is a vast reduction from the  $60 \times 210 = 12,600$  transition probabilities that would be computed if the algorithm were implemented without reuse.

Although such reuse does not reduce the computational complexity of the algorithm, for most problems this technique will greatly reduce the computational time required to run the algorithm.

Lastly, we note that if the time between two stages is equal to the time between two other stages, the transition probabilities can also be reused.

### 3.6 Static optimal solutions

In Section 3.3 we described an algorithm to determine when it is optimal to run overtime. The solution obtained by this algorithm is *dynamic* since the optimal policy is a function of the state space when the decisions must be made. In contrast, a *static* solution is one in which all decisions are made at a given point in time, and are not a function of the state of the system at future points in time. In this section we will show how to determine the static optimal policy, in which all decisions must be made at the beginning of the horizon and can not be changed over the course of the horizon. We will briefly examine whether or not such static solutions are competitive with the dynamic solutions discussed elsewhere in this chapter.

#### Determining static optimal solutions

In Section 3.2 we described a calculus-based approach for evaluating the cost of a given production plan. The evaluation involved the computation of

$$\sum_{j=1}^M |A_j| L_{aj} + \left( D_j - I_{JK_j}(0) - Q_{a-1,j} \right)^+ \times G_{A_j(a)} \left( 0; T_{A_j(a),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}} \right) - G_{A_j(a-1)} \left( \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}}; T_{A_j(a-1),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-2)}}{P_{A_j(a-2)}} \right),$$

where

$$L_{aj} = \int_0^{Q_{A_j(a)}} \left( D_j - I_{JK_j}(0) - Q_{a-1,j} - x \right)^+ g_{A_j(a)} \left( \frac{x}{P_{A_j(a)}}; T_{A_j(a),j}, \frac{Q_1}{P_1}, \dots, \frac{Q_{A_j(a-1)}}{P_{A_j(a-1)}} \right) dx,$$

and  $T_{ij} = TD_j - S_1 - \dots - S_i$ . We refer the reader to Section 3.2 for an explanation of the notation and an interpretation of these expressions.

The above evaluation procedure assumes that no overtime is purchased over the horizon. If instead we wish to evaluate the expected cost of a production plan under the assumption that the  $p^{\text{th}}$  overtime opportunity is purchased, we simply replace  $T_{ij}$  in the above expression with  $T_{ij} + OT_p$  for all  $j$  such that  $TD_j > TO_p$ , and re-evaluate the expected cost of the production plan, adding  $co_p$  to the cost.

In general, to find the optimal static policy when there are  $N_{OT}$  different overtime opportunities, we could evaluate each of the  $2^{N_{OT}}$  different possible combinations of running or not running overtime at each opportunity. Each evaluation can be performed by numerical integration or by Laplace transform inversion using the results of Chapter 2.

### **An improved algorithm**

In this section we describe approaches to simplify computation of the optimal static policy. These approaches only work if the marginal benefit of additional overtime is decreasing. Although this is not true in general, it is true for a variety of realistic numerical examples that we have explored.

We have seen that purchasing overtime opportunity  $p$  replaces  $T_{ij}$  with  $T_{ij} + OT_p$  for all  $j$  such that  $TD_j > TO_p$ . One can visualize this as “shifting” each demand point after the overtime opportunity to the right by  $OT_p$  time units. Before the results of this section can be utilized, the total expected cost function must be shown to be decreasing and convex as each demand point is shifted to the right. This can be accomplished with  $O(M)$  evaluations of the total expected cost function.

The algorithm we propose begins by evaluating the expected cost if no overtime is purchased. We then restrict attention to the  $N_{OT}$  different plans in which we choose only one overtime opportunity.

If the expected cost rises as a result of choosing one of the  $N_{OT}$  opportunities, then we can safely ignore that opportunity since it will have an even smaller impact on reducing expected cost if combined with other opportunities. By ignoring such an opportunity, we can eliminate  $2^{N_{OT}-1}$  of the possible combinations in which that opportunity is chosen.

The next step of the algorithm is to consider the different plans in which we choose any two overtime opportunities. Before we evaluate any additional plans, we can first compute lower bounds on the expected cost of any plan in which we choose two overtime opportunities as follows. First we calculate the benefit of choosing one of the opportunities by subtracting the expected cost of purchasing that opportunity from the expected cost if no overtime is purchased. We then do the same for the other opportunity under consideration. We then add the sum of the benefits to the expected cost if no overtime is purchased. When the marginal benefit of additional overtime is decreasing, this total will be a lower bound on the expected cost of purchasing both overtime opportunities. An example of this is given in the next subsection. If this lower bound is higher than any of the expected cost of any of the already computed plans, then the plan can not be optimal. For the combinations of two opportunities that produce a lower bound that is below the lowest expected cost of any already computed opportunity, we should evaluate their expected cost. Whenever we can eliminate a combination of two opportunities (since they will

result in an increase in expected cost), we can eliminate the  $2^{N_{OT}-2}$  possible combinations in which those two opportunities are chosen together.

In general, at the  $i^{\text{th}}$  step of the algorithm we evaluate the expected costs of the plans in which we choose  $i$  of the  $N_{OT}$  overtime opportunities. The algorithm terminates when we have eliminated all possibilities or have reached the  $N_{OT}^{\text{th}}$  step of the algorithm. In the worst case, we must evaluate every possible combination. The number of evaluations is then

$$\binom{N_{OT}}{0} + \binom{N_{OT}}{1} + \binom{N_{OT}}{2} + \dots + \binom{N_{OT}}{N_{OT}}$$

which is equal to  $2^{N_{OT}}$ . As a result, this procedure can not be worse than the fully enumerative procedure described earlier, except that we will do some additional work in computing the lower bounds. These bounds are extremely simple to compute, however, and will not affect the total running time of the algorithm in any substantive way.

We now briefly examine how the above problem can be viewed as a combinatorial optimization problem. Let  $S$  and  $T$  be index sets of the overtime opportunities such that  $S \subseteq T$ , let  $a$  be the index of an opportunity such that  $a \in T$ , and let  $v(\cdot)$  be the expected total cost of any subset of overtime opportunities. Then the fact that the marginal benefit of additional overtime is decreasing implies that  $v(T \cup \{a\}) - v(T) \leq v(S \cup \{a\}) - v(S)$ . Accordingly, the function  $v$  is submodular. This is a useful observation because a submodular function can be minimized in polynomial time. See Nemhauser and Wolsey (1988). Polynomial submodular function

minimization algorithms are quite complicated, and thus if  $N_{OT}$  is not large the above procedure, although not polynomial, may be preferred.

Lastly, we consider a simple special case, where the opportunities are each of the same length, and the opportunities earlier in the horizon are no more expensive than those that occur later in the horizon. In this case, we can see that the earliest opportunities are the most preferable since they afford the greatest protection against stockout. As a result, the best combination when purchasing  $n$  opportunities will be to purchase the first  $n$  opportunities. Therefore, we need only to evaluate  $N_{OT} + 1$  different combinations, where each combination considers purchasing the first  $n$  opportunities,  $n = 0, \dots, N_{OT}$ .

### **Comparison of static and dynamic optimum**

We now briefly examine whether a static solution is competitive with the dynamic solution. To address this question, we consider the following simple example. The production plan involves three parts built one time each over a horizon of 300 time units. We consider a short time horizon since we expect this to be favorable to a static solution. For simplicity we consider a symmetric problem, that is, where all the parts have the same parameters. The parts have demands of 30 units each at 100 time unit intervals and lot sizes of 60 units. The  $MTBF = 20$  and  $MTTR = 15$  time units, for an SAA of 57%. The production speeds are assumed to be one, and we ignore setup times (i.e., assume that they are zero), so that the expected utilization of the machine is 105%. There are three overtime opportunities of length 10 time units located 15 time units before each demand point. The overtime cost is 3.5 per time unit and the stockout cost is 10 per unit. This data is summarized in Table 3.2. As in the base case experiment of Section 3.3, we set the terminal costs to the expected cost of the amount of overtime required to complete any unfinished

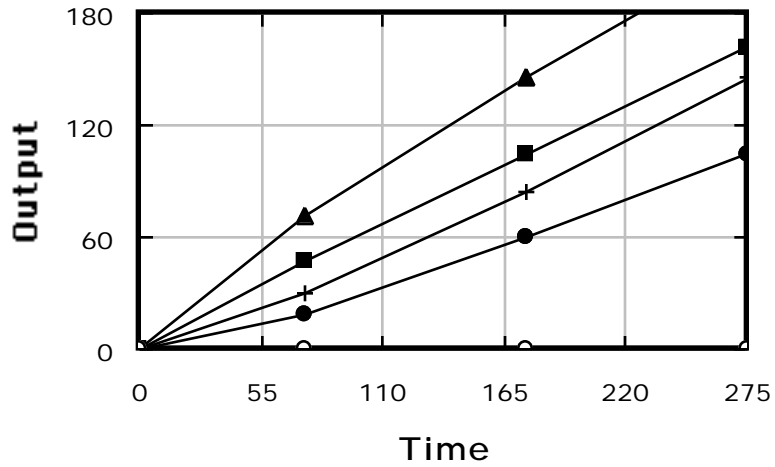
portion of the production plan. For the base case, the terminal costs are set to  $3.5 \times (3 \times 30 - ) / 0.57 + 3.5 \times (1 - ) / \mu$ . This experiment is very similar to the base case experiment of Section 3.3, except with a shorter horizon and slightly higher failure rate (and thus slightly higher utilization).

<u>Demand points</u>			<u>Overtime Opportunities</u>		
Part	Time	Quantity	#	Time	Length
1	100	30	1	75	10
2	200	30	2	175	10
3	300	30	3	275	10
Horizon length = 300					
Production batch size = 60			MTBF = 20		
Production rate = 1			MTTR = 15		
Utilization = 105%			SAA = 57%		
Backorder cost = 10			OT Cost = 3.5		

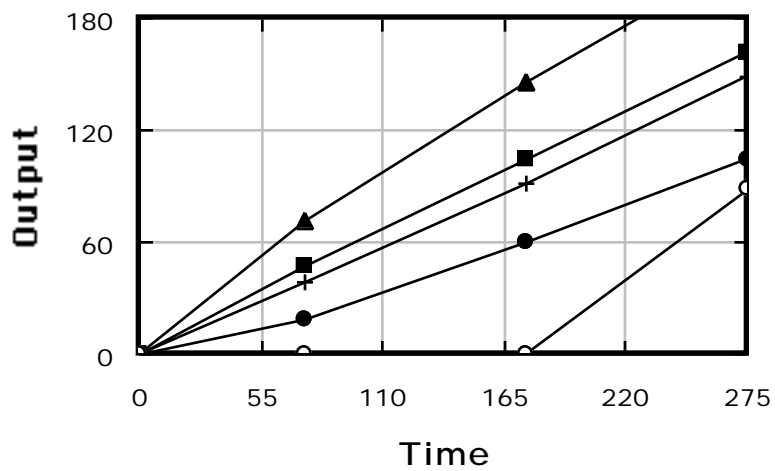
**Table 3.2** Data for experiment

The confidence intervals of machine output, critical overtime levels, and lower envelope are shown in Figure 3.25 for the case where the machine is working, and Figure 3.26 for the case where the machine is failed. We note that although the lower envelope is positive at the last decision point (at time 275) when the machine is failed, we see that the envelope is below the 5-%ile of the machine output distribution.

The optimal dynamic solution has an expected cost of 130 if the machine is initially working, and 183 if the machine is initially failed. We expect this large difference in expected costs between the working and failed cases due to the high utilization of the machine and the size of the MTTR relative to the length of the horizon.



**Figure 3.25** Critical overtime levels and confidence interval of output (machine working).



**Figure 3.26** Critical overtime levels and confidence interval of output (machine failed).

Overtime Opportunities			Expected cost	
1	2	3	Machine up	Machine down
0	0	0	198	280
0	0	1	203	278
0	1	0	197	269
0	1	1	206	271
1	0	0	196	265
1	0	1	205	267
1	1	0	201	260
1	1	1	214	267

**Table 3.3** Expected cost of static policies.

In Table 3.3 we show the expected cost of the  $2^3 = 8$  different static policies. The ones and zeroes in the first three columns indicate whether or not that overtime opportunity was purchased (1 = yes, 0 = no). We see that the optimal static policy is to purchase overtime opportunity #1 only if the machine is working at time zero, and purchase the first two if the machine is failed at time zero.

We note that as expected, the best policy if only one overtime opportunity is chosen is (1,0,0) and the best policy if two overtime opportunities are chosen is (1,1,0). This is consistent with the principle that overtime earlier in the horizon has greater value.

Let us ignore the fact that this problem has the special structure where the opportunities are of the same length and cost, and examine the lower bounding procedure that we described in the previous subsection for the more general problem. In general, one must examine the total cost function to ensure that the benefit of additional overtime is decreasing before applying the lower bounds. Although we have not done this, it will be evident that the benefit of additional overtime is decreasing for this problem because we have enumerated all the possible solutions.

The bounds are reported in Table 3.4. For the case (0,1,1) for example, the lower bound is computed from the cost of (0,0,1) over (0,0,0) [ $203 - 198 = 5$ ], plus the cost of (0,1,0) over (0,0,0) [ $197 - 198 = -1$ ], plus the cost of (0,0,0) [198], for a total of 202. Since the actual expected cost was 206, we report a gap of size 4. Table 3.4 lists four different bounds for the case (1,1,1) since it can be computed in four different ways. The first and weakest is to sum the benefits of (0,0,1), (0,1,0) and (1,0,0). The other

three ways are to sum (0,1,1) and (1,0,0); (1,0,1) and (0,1,0); and (1,1,0) and (0,0,1). Of course, we would have concluded immediately that no policy that chooses overtime opportunity 3 can be optimal, so we would have eliminated policies (0,1,1), (1,0,1) and (1,1,1) without needing to evaluate them.

Overtime Opportunities			Expected cost	
1	2	3	Lower bound	Gap
0	1	1	202	4
1	0	1	201	4
1	1	0	195	6
1	1	1	200	14
1	1	1	204	10
1	1	1	204	10
1	1	1	206	8

**Table 3.4** Lower bounds on expected cost of static policies.

We conclude with the observation that the expected cost of the best static policy is over 50% higher than the cost of the dynamic optimal policy in the case where the machine is working at time zero, and 70% higher in the case where the machine is failed at time zero. Based on this limited evidence, it should be clear that there are benefits to the dynamic optimization that we propose even over short time intervals and moderate variability. Over longer time horizons or under greater production variability (e.g., higher MTTR for a fixed SAA – see Section 2.8), the superiority of dynamic optimization will be even more pronounced.

We do not mean to imply that static optimization can not perform well in certain circumstances. For example, if the machine utilization is very low, the expected amount of overtime purchased may be very low. In an extreme case, the static optimal policy might not purchase overtime, which could be quite competitive (in terms of expected cost) with dynamic optimization. It is important to realize, however, that such cases are not very interesting.

### **3.7 Extensions**

In this section we describe a variety of different extensions to the basic model of Section 3.3. These extensions do not change our basic methodology; each involves the solution of a dynamic program by a backward recursion scheme. However, we will see that the structure of these dynamic programs and algorithms to solve them will differ substantially from our basic model.

#### **Early overtime authorization**

In Section 3.3 we described a model for determining whether or not to purchase overtime opportunities. This model assumes that the overtime opportunities are fixed in length and occur at certain fixed points in time. In some real-world contexts, the decision as to whether or not to run overtime must be made in advance of the point in time that the overtime actually begins. For example, a certain union agreement might require that the decision regarding whether or not overtime is run at the end of the day must be made by 10:00 AM on that same day. We now describe how to incorporate such an extension into our model.

In this section we will show several diagrams such as the one in Figure 3.27, which is intended to represent the dynamic programming logic described in Section 3.3. The circles represent the possible discretized states. This diagram does not distinguish between the two possible machine states, working or failed. The columns of circles represent the stages of the dynamic program (numbered in reverse chronological order). In this diagram time flows from left to right, and the transitions are made from left to right.

The lines between stages represent the possible transitions, each with an associated probability. Two types of lines are shown: solid black and gray. The gray lines that leave a state are intended to represent the transitions if an overtime opportunity is purchased at that state. No gray lines should leave a state if overtime opportunities are not permissible in that state.

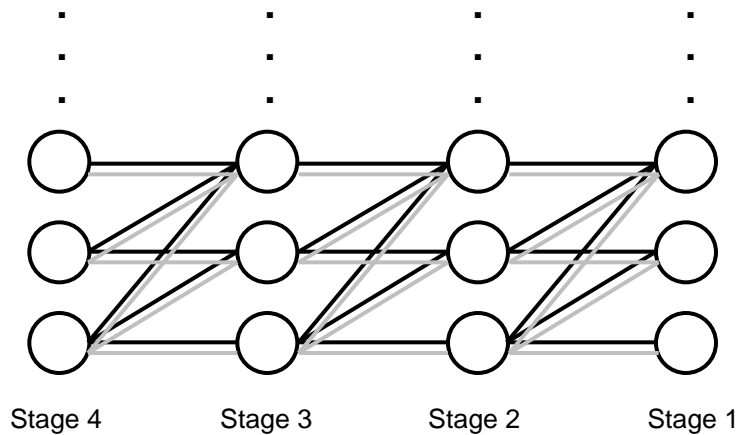
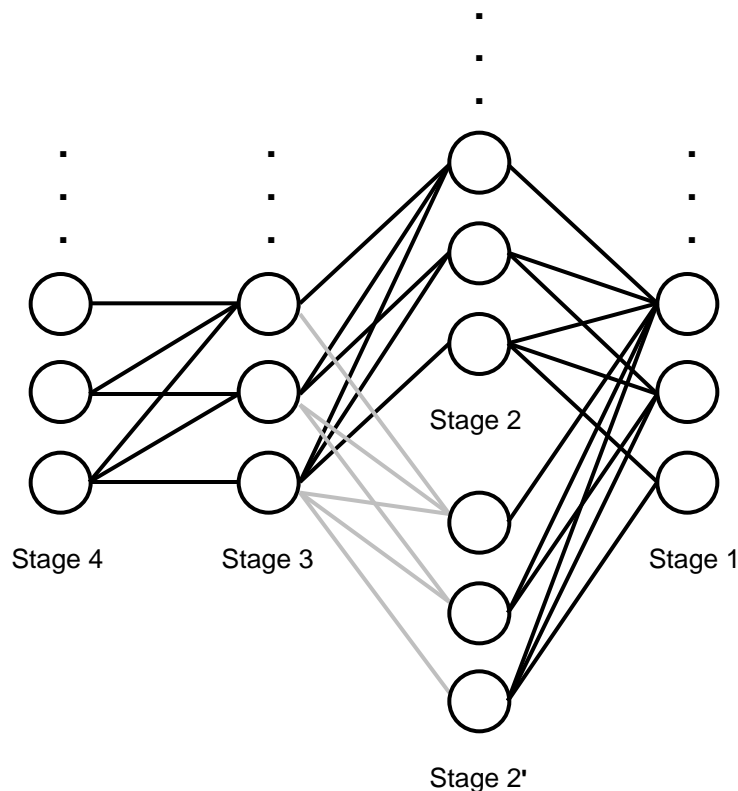


Figure 3.27 Stages and transitions in dynamic programming algorithm

Now consider the effect of requiring that a decision regarding overtime at the  $p^{\text{th}}$  opportunity must be made at time  $TO_p - x$  ( $x > 0$ ), instead of  $TO_p$ . Suppose that the stage that corresponds to the  $p^{\text{th}}$  overtime opportunity is stage  $n$ . We must first change the “time” of stage  $n$  to be  $TO_p - x$  and then reindex the stages so that they are again in reverse chronological order. If no reindexing is required, then the dynamic program can be solved as before. The decision regarding overtime will be made  $x$  time units earlier, so that the time between stages  $n+1$  and  $n$  will be  $x$  time units shorter and the time between stages  $n$  and  $n-1$  will be  $x$  time units longer. This will affect the transition probabilities, and accordingly, the expected cost to go that is computed at these stages.

If changing the time of stage  $n$  disrupts the chronological order of the stages, then further modification to the dynamic program is required. Suppose that after the stages are reindexed in reverse chronological order, the stage corresponding to the  $p^{\text{th}}$  overtime opportunity is  $m$ , where  $m > n$ . Then for each stage  $m-1, m-2, \dots, n$ , we create a duplicate set of states that we will denote by stage  $(m-1)', (m-2)', \dots, n'$ . These duplicate stages are incorporated into the model as follows. If we decide to purchase overtime at a state in stage  $m$ , then we transition to the duplicate stage  $(m-1)'$  instead of stage  $m-1$ . This is depicted in Figure 3.28 for the case  $n = 2$  and  $m = 3$ .



**Figure 3.28** Modified stages and transitions for early overtime authorization

The transition probabilities from stage  $m$  to  $(m-1)'$  are the same as those from stage  $m$  to  $m-1$ . Similarly, the transitions between stages  $(m-1)'$  and  $(m-2)'$ ,  $(m-2)'$  and

$(m-3)', \dots, (n-1)'$  and  $n'$  are the same as the transitions between stages  $m-1$  and  $m-2$ ,  $m-2$  and  $m-3$ ,  $\dots$ ,  $n-1$  and  $n$ . The transition probabilities differ only in the transition from stage  $n'$  to  $n-1$  (versus the transition from stage  $n$  to  $n-1$ ). When transitioning from stage  $n'$ , we add  $OT_p$  to the time available for production. The net result of these changes is simple: a transition to the duplicate set of states (those whose stage we have denoted with a prime) represents a commitment to purchase the  $p^{\text{th}}$  overtime opportunity. We do not see the benefit of this purchase until the transition into the  $n-1^{\text{st}}$  stage.

Since decisions must be made earlier (i.e., with less information), the total expected cost will increase. We have seen this effect in Theorem 3 of Section 3.4. Further, we expect the critical overtime levels to decrease, as seen in Section 3.3 and Figure 3.11 in an experiment where the overtime opportunities were moved earlier in the horizon.

### **Overtime opportunities of variable size**

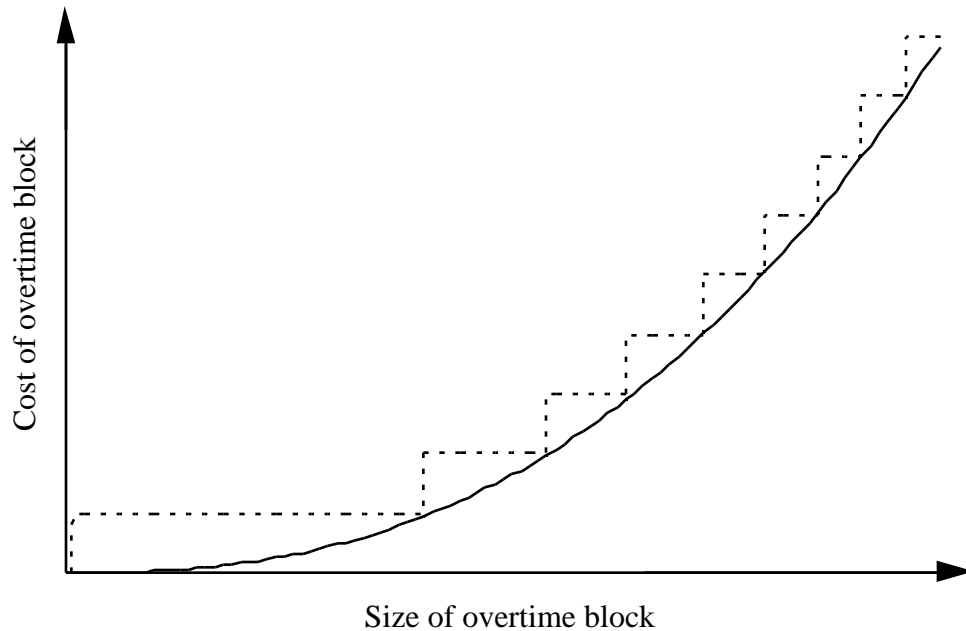
The model of Section 3.3 assumed that the overtime opportunities are fixed in length. In this and in the following subsection we show two ways to extend our model to incorporate overtime opportunities of variable size.

In this subsection we describe an extension to the model of Section 3.3 in which overtime can be dynamically purchased in a series of discrete blocks. We refer to this extension as dynamic purchasing since after a block of overtime is purchased, the state of the system is observed before a decision must be made whether or not to purchase additional overtime. In contrast, the extension of the next subsection might be called static purchasing, since the quantity of overtime to be purchased is chosen and all the overtime is performed without an opportunity for recourse.

We now assume that the cost of overtime at the  $p^{\text{th}}$  overtime opportunity  $c_p(t)$  is increasing and convex in  $t$ , the amount of overtime purchased. With this assumption, we can modify our dynamic program to permit the decision maker to purchase overtime in a series of discrete blocks, where the size of the blocks are determined by the places where the “steps” occur. Based on the current system state, the decision maker can stop running overtime at any of the discrete points. Thus, we are incorporating a continuous choice of overtime quantity into the dynamic program by approximating the cost function  $c_p(t)$  as a step function. The discretization can have any number of steps, of any length and size. Figure 3.29 shows a discretization with equal size cost increments. This particular choice of discretization results in a very large minimum purchase. Some care should be taken to choose an appropriate discretization, although real-world circumstances, such as union contracts or other agreements with workers may dictate the appropriate discretization.

We now describe the required modifications to the dynamic program. Previously we used a single stage of the dynamic program to represent the decision of whether or not to purchase a fixed size block of overtime at a particular opportunity. We now model the overtime opportunity as a series of stages, where each stage represents a discrete block. At a stage, the decision maker has the opportunity to purchase the discrete block of overtime. If the block of overtime is purchased, the overtime is performed and the decision maker observes the output of the machine and the state of the machine at the end of the block of overtime before deciding whether or not to purchase the next block. If the decision maker does not purchase the next block of overtime, the overtime opportunity is over. This is because the overtime blocks will be increasing in marginal cost, so that if it is not optimal to

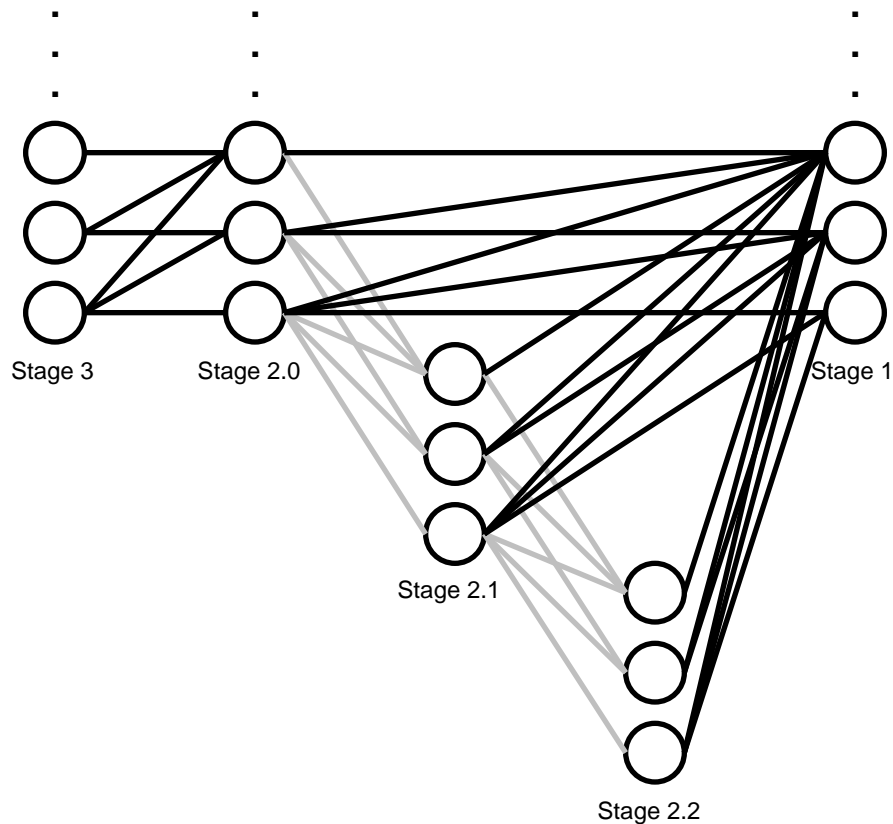
purchase a block, it will not be optimal to purchase an even larger quantity of overtime at a higher per time unit cost. This follows from Theorem 5 of Section 3.4, where we showed that the critical overtime levels are non-increasing as a function of the cost of the overtime block.



**Figure 3.29** Discrete approximation of convex cost function

Figure 3.30 shows an example of how this modification to the dynamic program is performed. In the example shown, the overtime decision corresponding to Stage 2 is broken into three discrete choices. We have labeled the corresponding stages Stage 2.0, 2.1 and 2.2. At Stage 2.0, the first block of overtime can be purchased, or not. If the block is not purchased, then a transition occurs to Stage 1. If the block is purchased (represented by a gray line), we transition to Stage 2.1 and the cost of the block is incurred immediately. The transition probabilities are determined by the random output of the machine, where the available time for production is equal to the size of the overtime block. At Stage 2.1, we can purchase the second block and

transition to Stage 2.2, or not purchase the block and transition to Stage 1. Lastly, at Stage 2.2 we transition to Stage 1 whether or not we purchase the last overtime block. If we do purchase the block, the cost is immediately incurred and the additional time for production is taken into account in the transition probabilities.



**Figure 3.30** Modified stages and transitions for variable size overtime opportunities

In this instance, to solve the dynamic program we would again start at the end of the horizon and work backwards computing the optimal cost to go at each stage. When we reach a stage where there is a variable sized overtime opportunity (such as Stage 2.0-2.2 in Figure 3.30), the dynamic program can still be solved by backward recursion. In the case of the example in Figure 3.30, once the cost to go has been computed for Stage 1, the cost to go is computed for Stage 2.2 in the usual way. Once

this is finished, the cost to go for Stage 2.1 can be computed for either action (purchase the second overtime block or not), and the optimal decision determined for each state. Once this is finished, the same can be accomplished at Stage 2.0, and then the dynamic programming recursion proceeds as before to Stage 3 and continues to the beginning of the horizon in this fashion.

Replacing a fixed sized opportunity with a variable sized opportunity composed of smaller overtime increments can not result in an increase in total expected cost, since the set of actions available to the decision maker has been expanded at no additional cost. Recall that the computational complexity of the dynamic programming algorithm is linear in the number of stages, so that the computational effort will increase linearly with the number of steps in the discretization of the variable sized opportunity.

### **Choosing among a set of overtime opportunities**

In the previous subsection we looked at an extension to the model of Section 3.3 in which overtime could be dynamically purchased in a series of discrete blocks. In the extension of this subsection, the quantity of overtime to be purchased is chosen and all the overtime is performed without an opportunity for recourse.

The extension presented in this subsection is of interest for two reasons. First, we no longer require the restriction of the previous subsection that the cost of overtime at the  $p^{\text{th}}$  overtime opportunity  $c_p(t)$  be increasing and convex in  $t$ . Second, the static purchasing scenario may be an accurate representation of reality. We have seen real-world environments in which the quantity of overtime purchased must be decided in advance of the point at which overtime begins, although there is flexibility in terms of how much overtime is purchased.

Let us first consider a variable-sized overtime opportunity at time zero. For example, this would correspond to an opportunity over the weekend before any production begins Monday AM. The dynamic programming algorithm can easily facilitate evaluation of such an opportunity with minimal modification.

Consider Figure 3.31, shown below, a modification of Figure 3.1 from Section 3.2. In this diagram we plot time on the horizontal axis and  $x$  on the vertical axis. The white circles represent the possible values of  $x$  that can be reached, while the shaded circles represent values of  $x$  that are not achievable even if the machine does not fail. The lines between the circles represent the possible transitions. We have not shown every possible transition, only those from  $x = 0$  (shown as solid lines) and the transitions that would result if the machine did not fail at all (shown as a dashed line).

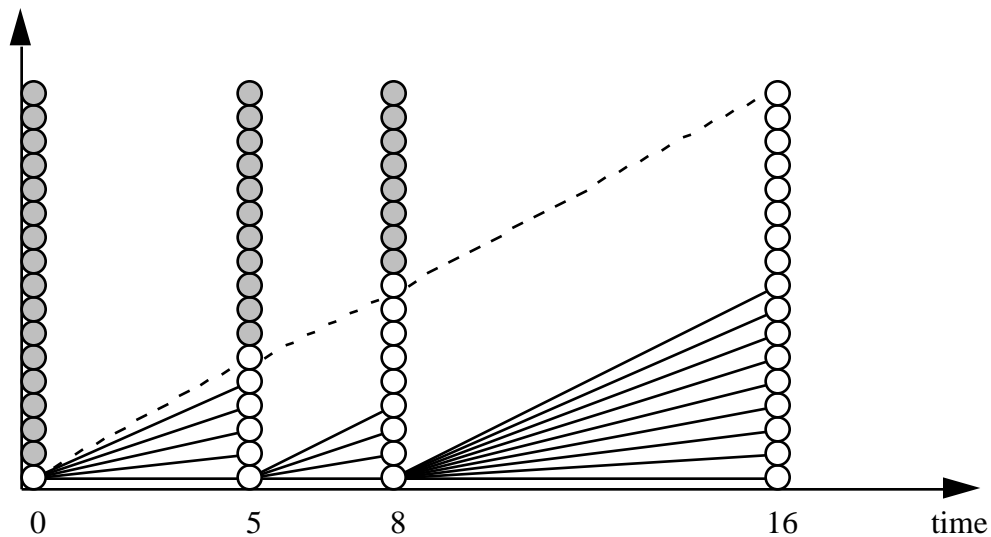


Figure 3.31 Additional states needed to evaluate variable sized overtime opportunity at time zero

The first step to evaluate an overtime opportunity of variable size at time zero is to solve the dynamic program as before, but instead of limiting attention to those states shown in Figure 3.31 as white circles, compute the expected cost to go at the shaded circles as well. Once the dynamic program has been solved, the expected cost to go vector for time zero tells us the benefit that would result if, instead of starting at  $x = 0$ , we could start at some other value of  $x$ . Denote this vector for a given value of  $x$  by  $c_N(x, t)$ .

Denote the cost of purchasing  $t$  time units of overtime by  $c(t)$ . Then the optimal amount of overtime to purchase at time zero is found by solving

$$\underset{0 \leq t \leq t_{\max}}{\text{minimize}} \quad c(t) + \sum_{a=0}^1 c_N \left( \min(t, S_1) + \int_0^{(t-S_1)^+} x f(x; (t-S_1)^+ | a) dx, a \right) P_a(t),$$

where  $t$  is the amount of overtime purchased,  $x$  is the initial state of the machine,  $a$  is the state of the machine when the overtime is completed,  $S_1$  is the setup time required before production can begin,  $\min(t, S_1)$  is the amount of the setup that is completed on overtime, the integral is the expected uptime of the machine over an interval of length  $(t - S_1)^+$  with initial machine state  $x$ , so  $\min(t, S_1)$  plus the integral is the expected value of  $x$  at the end of the overtime period, and  $P_a(t)$  is the probability that the machine is in state  $a$ , given that  $t$  time units earlier it is in state  $x$ .

$t_{\max}$  will typically be a constraint on the amount of overtime that is available before time zero, although if such a constraint does not exist, then it should be set to the largest value of  $t$  that is achievable over the horizon. Since a simple closed form expression has been found for the above integral (given by equations (18) and (19) of

Chapter 2), the optimal value of  $t$  can be found with very little effort by simple enumeration.

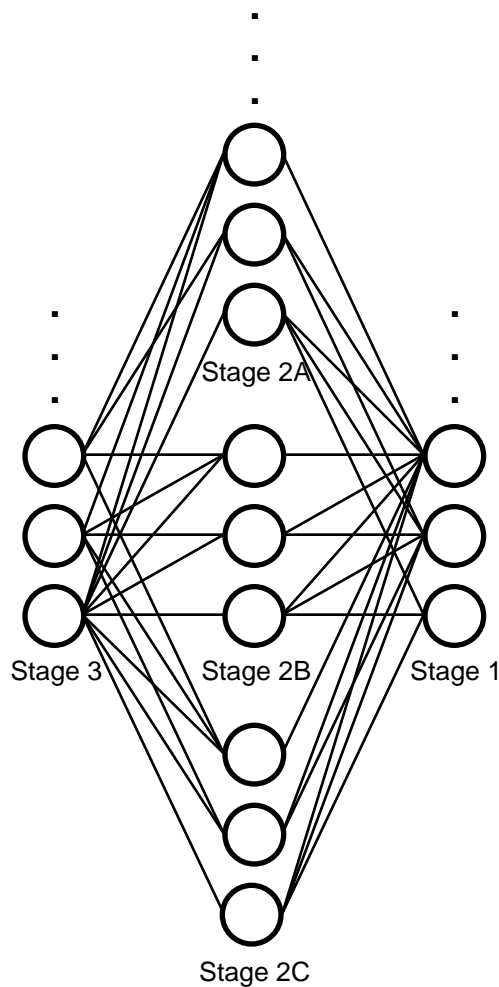
The development above has assumed that  $t_{\max}$  is such that production of the first batch can not be completed even if all  $t_{\max}$  time units are purchased. If this is not the case then the above minimization must also take into account the additional setup times, and machine reliabilities across different parts, if they differ.

We now turn our attention to choosing among a set of overtime opportunities in the middle of the horizon. These overtime opportunities could be of any length and any cost. In Section 3.3 we used a single stage of the dynamic program to represent the decision of whether or not to purchase a fixed size block of overtime at a particular opportunity. To incorporate several different overtime opportunities that are available at a single point in time, we create a stage for each opportunity and place these stages in parallel.

An example of this is shown in Figure 3.32. For simplicity of the diagram we have not drawn the additional gray lines that represent transitions when overtime is purchased. In this example we are replacing the overtime opportunity at Stage 3 with three different overtime alternatives, where each alternative has a different length and cost of overtime (where typically one of the alternatives is to not run overtime and not to incur any overtime cost). We take the subsequent stage, Stage 2 in this example, and replace it with three stages in parallel, which we have labeled 2A, 2B, and 2C. A transition from stage 3 to Stage 2A represents a choice of the overtime alternative "A". Accordingly, the available machine time that is available between Stages 3 and 2A reflects the amount of overtime purchased, and the

immediate cost at Stage 2A is set to reflect the purchase of overtime alternative “A”. The transitions and costs from Stages 2A-2C to Stage 1 are unchanged.

The addition of alternatives broadens the set of actions that are available to the decision maker at no additional cost, so the total expected cost can not increase. Each additional alternative adds one stage to the dynamic program. Recall that the computational complexity of the dynamic programming algorithm is linear in the number of stages, so that the computational effort will increase linearly with the number of alternatives presented.



**Figure 3.32** Modified stages and transitions for choosing among a set of overtime opportunities

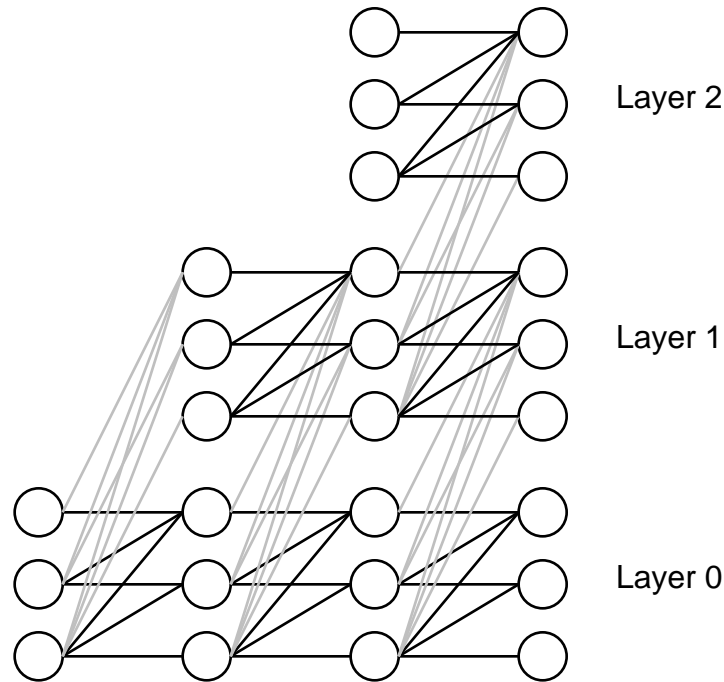
### **Constraining the number of overtime opportunities used**

In the development thus far, we have assumed that there is no restriction on the number of overtime opportunities that can be purchased. In some real-world situations however, there may be such constraints, for example, where only two out of any three consecutive weekends can be used for overtime. We now show how to accommodate such a constraint into our model.

The basic idea is to make copies of all the states and stages in the dynamic program. We will call such a copy a *layer*. As before, the initial stage corresponds to the beginning of the horizon. We start out in Layer 0. The dynamic program is structured as before, where transitions occur to the next stage (within the same layer). The difference is that if we choose to run overtime, we transition to the next stage, but in one layer higher. This is depicted in Figure 3.33. The number of layers is equal to the maximum number of times that we are permitted to run overtime over the horizon, plus one. The layer number indicates how many times we have run overtime thus far. In the topmost layer, we do not permit overtime to be run, thereby enforcing the constraint.

The dynamic programming algorithm proceeds very much like before. It starts with the topmost layer and performs the backward recursion the first stage is reached. This can be done because the cost to go at any stage in Layer 2 is not a function of the other layers. We then move one layer downward, and perform the backward recursion starting with the last stage. The cost to go in this layer is only a function of the cost to go at the topmost layer. This continues one layer at a time until the first stage of Layer 0 is reached. Note that we do not need every stage in every layer, since we can only transition up one layer per stage. Therefore, on the  $n^{\text{th}}$  layer the stages

up to and including the stage that represents the  $n^{\text{th}}$  overtime opportunity can be omitted.



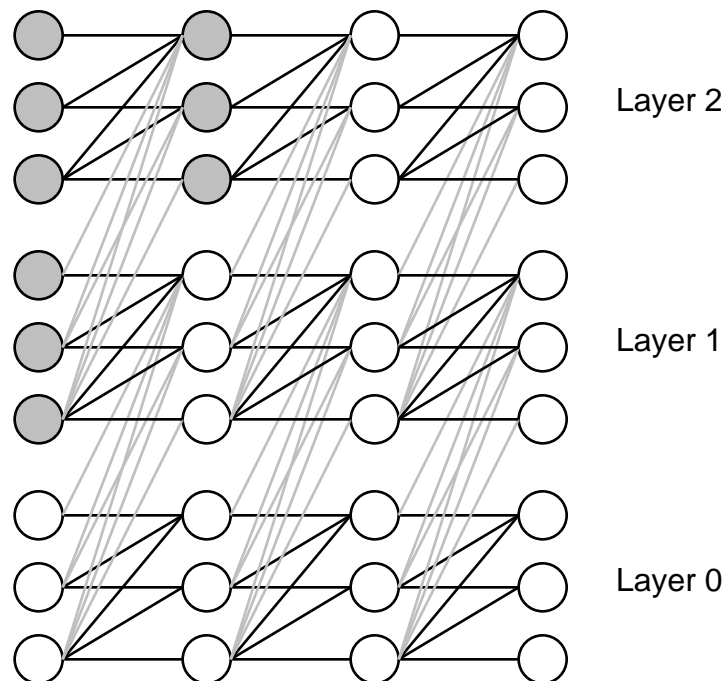
**Figure 3.33** Modified stages and transitions for choosing among a set of overtime opportunities

Although the above procedure at first appears to entail a significant amount of additional computation, this is in fact not the case. Since the transition probabilities are the same between any two successive stages irrespective of which layer we are evaluating, the number of transition probabilities that need to be computed does not increase as a result of the addition of layers. Recall that the computation of the transition probabilities will dictate the running time of the algorithm, so this extension requires very limited additional computational effort.

As a result, we observe that we are able to obtain sensitivity information on the number of overtime opportunities that are permitted. In particular, by computing the expected cost to go at a few additional stages we can evaluate the impact of

reducing the number of overtime opportunities that are available. In Figure 3.34 the additional stages that need to be evaluated are shaded in gray. In this example there are two layers so we permit two overtime opportunities. By computing the expected cost to go at the first stage on Layer 1, we will have determined the increase in total cost that would result if only one overtime opportunity were available. Similarly, the expected cost to go at the first stage on Layer 2 tells us the increase in total cost that would result no overtime opportunity were available.

Since the addition of layers is not computationally expensive, it is therefore quite practical to add Layers -1, -2, ... below Layer 0, that tell us the benefit that results if we had one, two, ... extra overtime opportunities.



**Figure 3.34** Evaluation of a decrease in the number of overtime opportunities permitted

Of course, we have only considered the most simple type of constraint that can be accommodated. For example, if the length of the horizon were two weeks, one

could place one constraint on the number of overtime opportunities permitted in the first week, and a second constraint on the number of overtime opportunities permitted in the second week. Another possibility is only to constrain some of the overtime opportunities, e.g., to place a constraint on the number of times that overtime can be worked on the weekend. Limitless other possibilities exist. The ones that we have mentioned are reasonably straightforward.

An entirely different type of constraint that can be accommodated is a constraint on the amount of *time* that can be consumed, e.g., no more than eight hours of overtime per week. Such a constraint might arise from human resource issues, or might be a result of necessary machine downtime for activities such as preventative maintenance.

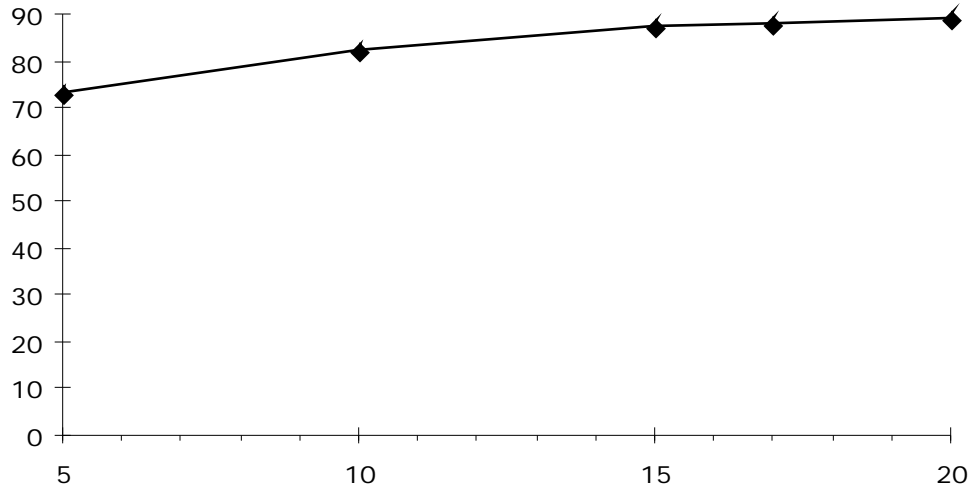
The general methodology for incorporating such a constraint is very similar to the one that we have just described. The layers now represent the amount of overtime (in terms of time) that has been consumed, instead of the number of times that overtime has been worked. An appropriate discretization must be chosen; suppose this is 15 minutes. If an overtime opportunity that is 60 minutes in length is undertaken, then the transition moves not one layer higher as before, but now four layers higher. Otherwise the algorithm is unchanged.

### **3.8 Steady-state Analysis**

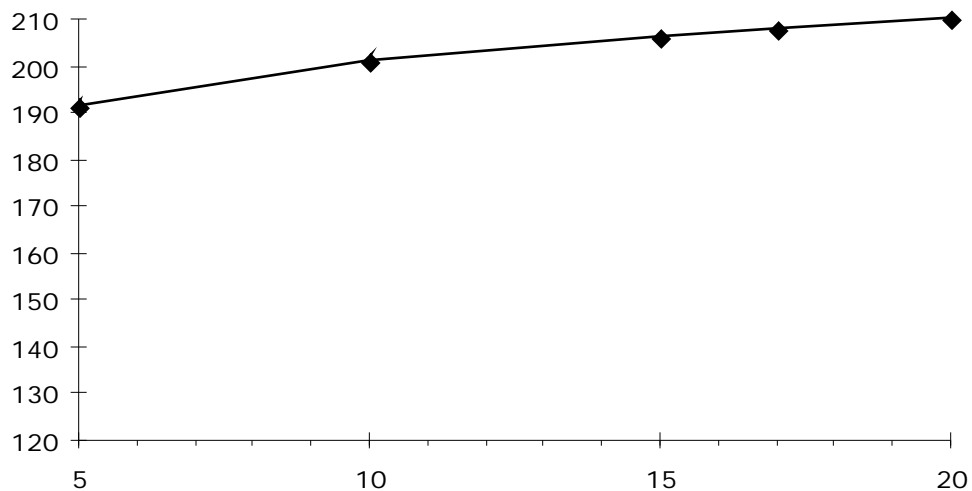
In this section we examine the impact of the finite horizon assumption that we have made in the preceding sections. We show empirically how the critical overtime levels are affected by increasing the length of the horizon, and briefly examine the factors that influence the rate at which the steady state is attained.

We use the base case described in Table 3.1 of Section 3.3. We first suppose that the horizon were twice as long, where the demand points and overtime opportunities in the second half of the horizon are identical to those in the first half. This doubles the number of demand points to 10. Figure 3.35 shows the impact of gradually increasing the length of the horizon (while adding demand points and overtime opportunities) on the critical overtime level at the first overtime opportunity. We see that the length of the horizon initially has a noticeable effect, although this effect rapidly diminishes and a steady-state is achieved. This indicates that the horizon length of 1000 minutes (16.66 hours) in the base case was too short for accurate decision making. However, we see that by the time there are 15 demand points (which corresponds to a horizon length of 50 hours), the steady-state critical overtime level is essentially achieved.

In Figure 3.36 we show the results of the same experiment, except now we consider the critical overtime level at the second demand point. The convergence to a steady-state value is slightly slower and the percent difference between the critical overtime level with five demand points and the steady-state value is slightly larger. Not surprisingly, we observe in general that the critical overtime levels further out in the horizon are more affected by the length of the horizon.



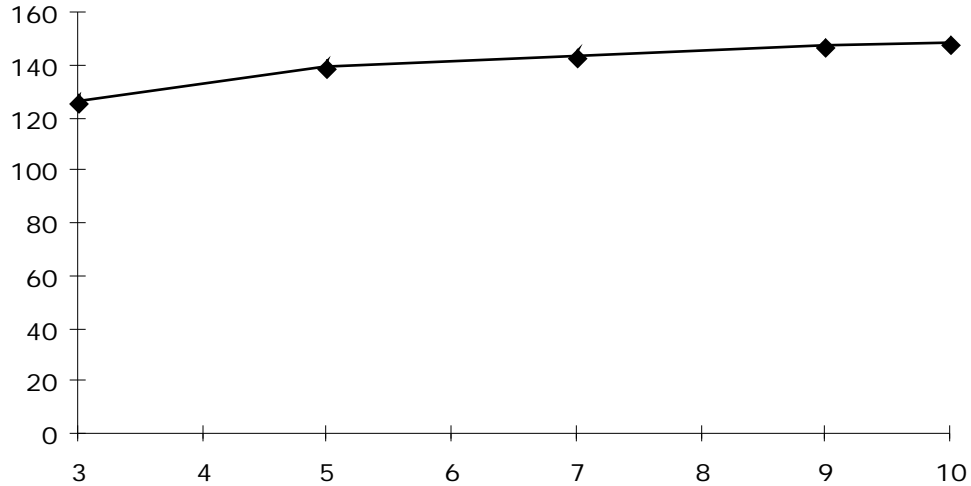
**Figure 3.35** Critical overtime level at the first decision point with varied number of demand points



**Figure 3.36** Critical overtime level at the second decision point with varied number of demand points

We now wish to demonstrate that it is the length of the horizon and not the number of demand points that determines the deviation of the critical overtime levels from their steady-state values. This is done by taking the base case data and doubling the production batch size to 240, doubling the demand quantities to 120, placing the demand points 400 units apart, and placing the overtime opportunities 50 units before each demand point. Figure 3.37 shows the impact of varying the

number of demand points on the first critical overtime level. We see that steady state is essentially achieved with half of the number of demand points.



**Figure 3.37** Critical overtime level at the first decision point all data doubled

We conclude by mentioning the results of two experiments we have not included here. The first experiment was the opposite of the previous experiment, which showed that by halving the data values, the number of demand points required to reach steady state is twice as large. We have also observed that the rate at which convergence is achieved is accelerated when the cost of overtime is increased.

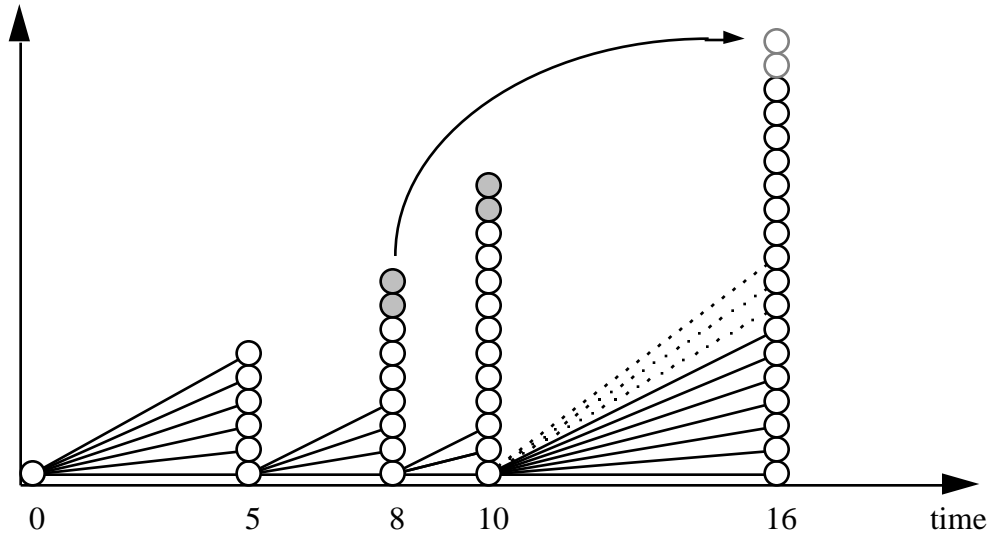
### 3.9 Rescheduling and sensitivity analysis

In this section we describe how to modify the dynamic programming algorithm discussed earlier in this chapter to obtain information regarding the sensitivity of the model to changes in the inputs.

The first and possibly most important question we address is how to use the algorithm to determine when and where rescheduling is beneficial. We consider a special type of rescheduling that we will call “cutting short”. Cutting short means shifting a portion of a batch that was intended to be built now to the next batch that was planned for that same part. The rationale for this type of rescheduling is that many facilities will produce batches in excess of the immediate requirements, motivated by economic lot sizing concerns. As a result, the size of the batch can often be reduced without risk of stockout at the next demand point. Cutting short is therefore desirable when there is an imbalance between the intended production quantities and demand, either due to the non-stationarity of the demand, or because of above-average levels of downtime in the immediate past. One could therefore think of cutting short as rebalancing. In one particular plant, we have observed that real-world schedulers frequently employed cutting short as a method of alleviating short term capacity problems.

Figure 3.38 is a modification of Figure 3.5 that depicts what a cutting short strategy might look like in terms of its impact on the state space. Here we are cutting short by two units at time 8, resulting in the “removal” of the two shaded states, and shifting this additional work to time 16 (shown as dotted circles). Note that since the state variable  $x$  measures cumulative output, the cumulative requirements for

each stage between times 8 and 16 (which is only time 10 in this example) decrease by two units.



**Figure 3.38** State space representation with rescheduling

Computing this information is computationally inexpensive. The dynamic program must be re-solved using the modified production plan, but the transition probabilities do not need to be recomputed if the machine reliability is the same across all parts<sup>\*</sup>. Recall that the computational effort is largely determined by the computation of the transition probabilities, so this extension requires very little additional effort. Determining the marginal benefits of shifting production between production runs requires one problem to be solved for each pair of production runs of the same part. Thus, if each part were produced twice over the horizon, there would be  $N$  different problems to be solved. Note that when these  $N$  different problems are solved by backward recursion, there is much replication of effort

---

<sup>\*</sup> If machine reliability is part dependent and we are only shifting one part (in order to evaluate marginal benefits), then although the transition probabilities will change as a result of cutting short, the effect will be very minor, and using the old transition probabilities will be an excellent approximation.

which can be avoided if an algorithm is coded that reuses the computational results from previous problems.

An alternative to computing marginal benefits is to restrict attention to cutting short in quantities equal to the lot size minus the sum of the demands before the next production run. Perhaps the ideal is to have an interactive tool that could be placed in the hands of a human scheduler, who could then explore a richer set of alternatives.

Although we have framed the above discussion in terms of “cutting short”, all of the same ideas can be applied to the rescheduling strategy of “getting ahead”, in which the target lot size for the next run of a part is increased to alleviate the load on the machine for the next production run. This may not be possible in some environments because of unavailability of excess raw materials. It may also be undesirable if raw materials are in limited supply and shared across parts. Lastly, it may be unappealing in certain factories where the culture is such that excess inventories are viewed as wasteful. Indeed, such a strategy is counter to quality concerns and the just-in-time philosophy. Nevertheless, overbuilding and carrying excess inventory during times in which there is excess capacity may be an alternative worth considering.

The information that is obtained from a cut short or get ahead sensitivity analysis can also be used to estimate the shadow prices of the lengths of the overtime opportunities. One less unit that must be produced (e.g., by cutting short) is nearly equivalent to the benefit of one additional time unit of overtime, multiplied by the machine’s stand-alone availability. In this way, one can estimate the marginal

benefit of an additional unit of overtime. The marginal net benefit is obtained by subtracting the marginal cost of the additional time unit of overtime.

We now turn our attention to other sensitivity analyses. First, note that any change in the cost parameters (per time unit cost of overtime or per unit backorder cost) affects only the immediate costs at each stage, and therefore does not affect the transition probabilities. Once again, this means that the dynamic program can be re-solved very quickly as the cost parameters are varied.

The second observation that we make is that since we do not assume that there is any relationship between the demand quantities and production schedule, the demand quantities can be varied and affect only the immediate costs at each stage. Once again, this means that the dynamic program can be re-solved very quickly for different demand quantities. One potential benefit of this in addition to sensitivity analysis is to analyze different demand scenarios if the demand quantities are uncertain. This is discussed briefly in the next section.

### **3.10 Models with stochastic demand**

The purpose of this section is to make some progress toward understanding the impact of the addition of demand variability to our models. Up to this point in the chapter we have assumed that the timing and quantity of demand is known with certainty over some short horizon. While this may be an acceptable assumption for some environments, for others it will be highly unrealistic. The addition of demand variability causes the model to become much more difficult. We therefore restrict our attention to two special cases. The first incorporates stochastic demands in the special case where only one part is produced on the machine. The second special case we consider assumes that the demand for all parts occurs at the same point in time, there is only one such point over the horizon, and the precise demand quantity is not known until the last moment.

#### **3.10.1 Single part**

In the case where there is only a single part built on the machine, the state variable can be replaced with a new state variable that represents the inventory level of the part. We will effectively abandon the notion of a schedule, assuming that the machine produces at full capacity, so that the only control available is the quantity of overtime to purchase. We retain the same stages as the previous formulation, one stage for each demand point and one stage for each overtime opportunity, except now we insert an additional stage after each stage that represents a demand point. Like the stages that represent the demand points, there are no decisions made at these additional stages. The transitions between a stage that represents a demand point and the new stage that we have inserted after it are governed by the demand distribution. Since the state variable  $x$  has been replaced with a state variable

representing inventory level, the transitions between these two stages represent the fulfillment of demand.

The cost of stockouts are assessed after the demand is filled. Within this structure, we can capture either lost sales or backorders as a result of stockout. In either case we require the state space to include negative values for the inventory level. In the lost sales case, we treat the negative inventory valued states as though they were state zero (no inventory on hand) when computing the transition probabilities to the next stage. The negative states are necessary to help assess the appropriate lost sales costs. In the backorder case, the negative states have a physical interpretation that affects the transition probabilities to the next stage. Note that we can also assess a per unit inventory holding cost in the positive states if desired.

The only remaining detail is how to value inventory (or how to penalize a backlog) at the end of the horizon. As before, we can enter an arbitrary terminal costs that are a function of the state.

The model is then solved with a backwards dynamic programming recursion similar to the one described earlier in this chapter. The algorithm for computing expected cost to go and optimal overtime decisions is unchanged.

If inventory holding costs are assessed, it may be desirable to insert additional stages where the decision maker has the option to turn the machine off and stop producing. The state of the machine (on or off) could be carried as an additional state variable, and would approximately double the computational effort required.

The problem is much more difficult when there are multiple parts produced on the same machine, each with stochastic demand. An extension of the above model to the case of multiple parts would require an additional state variable for each additional part, plus the state variable to keep track of progress relative to the schedule. The addition of state variables increases the complexity of the dynamic programming algorithm exponentially, and therefore rapidly becomes unrealistic. We do not explore the well developed theory of state space reduction; the interested reader is referred to Larson (1968). We do note, however, that the model discussed earlier in this chapter could be used to evaluate a number of different demand scenarios. Although this is no substitute for a model that can accommodate stochastic demands, it may be of great assistance in helping a decision maker to understand the impact of demand uncertainty on the optimal overtime decisions.

### **3.10.2 Single demand point**

The model of this subsection will differ from the model discussed at the beginning of the chapter in three fundamental ways. First, we assume that there is only a single demand point for all parts, and that it occurs at the end of the horizon. The second major difference is that the demand for each part is now a random variable with a known distribution function, where the uncertainty in the demand quantity is not resolved until the demand point. Lastly, we assume a fixed production sequence as before, but we now find optimal production quantities, and later, overtime levels as well.

Initially we restrict ourselves to finding optimal set of production quantities. An alternative approach might be to find an optimal set of run times, where the available machine time is partitioned among the different parts. Although each of these policy types has its own merit, the best policy is a mixture of the two: a

dynamic policy in which the decision to stop production is based on both the realized output of the machine *and* the remaining time for production. We briefly explore an approximate dynamic policy of this type at the end of this subsection.

### Brief literature review

The classic single demand point model is the “newsboy” model (Lee and Nahmias, 1993), which has been and continues to be extensively studied. Some extensions include multiple items (Evans, 1967; Smith et al., 1980), uncertain replenishment (Rose, 1992), and multiple time periods for production (Bitran et. al, 1986; Matsuo, 1990). However, each of these models is fundamentally different from the one that we consider here. For example, Rose assumes that demand is deterministic, none of the authors except Rose address machine unreliability, and there does not appear to be any paper that addresses the option to purchase additional capacity (overtime).

### Formulation

The mathematical structure of our model will closely parallel that of the classic newsboy model, which we now briefly describe. Let  $x$  denote the current inventory level,  $c$  the unit purchase price,  $h$  the cost per unit of inventory remaining at the end of the period,  $p$  the unit shortage cost and  $g(\cdot)$  the PDF of demand. The problem is then to choose an order-up-to quantity  $y$  to minimize the expected purchase, holding and shortage costs. Mathematically, we can state the problem as

$$C^*(x) = \min_{y \geq x} c (y - x) + p \int_y^\infty (t - y) g(t) dt + h \int_0^y (y - t) g(t) dt.$$

The problem is solved by finding the value of  $y$  such that  $C(x)/y$  is zero. To find this partial derivative, we need to employ Leibnitz’s rule

$$\frac{1}{y} \int_{p(y)}^{q(y)} f(x,y) dx = \frac{1}{p(y)} \int_{p(y)}^{q(y)} \frac{f(x,y)}{y} dx + \frac{q(y)}{y} f(q(y),y) - \frac{p(y)}{y} f(p(y),y)$$

(Beyer, 1987). We will use this extensively in our analysis. From this rule it is easy to see that the optimal solution  $y^*$  to the newsboy model occurs at the point where  $G(y^*) = (p - c) / (p + h)$ , unless this implies  $y^* < x$ , in which case it is optimal not to order.

We now extend this basic single part model to our multiple part, unreliable production process model, for now ignoring overtime opportunities. The problem is to find the optimal order-up-to levels to minimize the sum of purchasing, holding and shortage costs over all parts. Let  $y$ ,  $x$ ,  $c$ ,  $p$ ,  $h$  and  $g(\cdot)$  retain the same meanings as above, except now we add a subscript  $i$ , for each part  $i = 1, \dots, N$ . We assume without loss of generality that the parts are indexed in the order in which they will be produced. Denote the setup time for each part as  $S_i$ . If we are already setup to produce part 1, then we set  $S_1 = 0$ . We assume for simplicity that each part is produced at the same rate when the machine is working ( $P_i = 1 \quad i$ ).

Let  $T$  denote the amount of time available for production, and the time available after setups as  $T_i = T - S_1 - \dots - S_i$ . As before, the CDF  $F(t; T)$  is the probability that in  $T$  units of time, the cumulative output of the machine is at most  $t$  parts. This distribution was discussed in detail in Chapter 2, although our results will not depend on the form of this distribution.

We can now write the problem as

$$C^*(x) = \min_{y_1, x_1, \dots, y_N, x_N} C(y, x)$$

$$\text{where } C(y, x) = \sum_{i=1}^N C_i(y, x),$$

$$\begin{aligned} \text{and } C_i(y, x) = & c_i \int_{x_i}^{y_i} (t - x_i) f_{y_j - x_j + t - x_i; T_i}^{i-1} dt \\ & + c_i (y_i - x_i) \bar{F}_{y_j - x_j; T_i}^i \\ & + p_i \int_{x_i}^{y_i} (t - u) g_i(t) dt f_{y_j - x_j + u - x_i; T_i}^{i-1} du \\ & + p_i \bar{F}_{y_j - x_j; T_i}^i \int_{y_i} (t - y_i) g_i(t) dt \\ & + p_i F_{y_j - x_j; T_i}^{i-1} \int_{x_i} (t - x_i) g_i(t) dt \\ & + h_i \int_{x_i}^{y_i} (u - t) g_i(t) dt f_{y_j - x_j + u - x_i; T_i}^{i-1} du \\ & + h_i \bar{F}_{y_j - x_j; T_i}^i \int_0^{y_i} (y_i - t) g_i(t) dt \\ & + h_i F_{y_j - x_j; T_i}^{i-1} \int_0^{x_i} (x_i - t) g_i(t) dt \end{aligned}$$

where the summations from 1 to i-1 are taken to be null at i = 1.

Each  $C_i(y, x)$  represents the expected purchasing, holding and shortage costs incurred for part i given a set of order-up-to levels  $y_i$ . We have written  $C_i(y, x)$  as the sum of eight terms. The first two terms express the expected purchasing cost, where the first term is the expected purchasing cost if the realized uptime of the machine is such that the available supply of the  $i^{\text{th}}$  part is between the values of 0 and  $y_i - x_i$  and the second term is the expected purchasing cost if the realized uptime of the machine is

such that the available supply of the  $i^{\text{th}}$  part is the desired value  $y_i - x_i$ . There is no purchasing cost if the available supply of the  $i^{\text{th}}$  part is not greater than zero. The next three terms represent the expected shortage costs. The first of these terms is the expected shortage cost if the available supply is between 0 and  $y_i - x_i$ , the second term is the expected shortage cost if the available supply is  $y_i - x_i$ , and the third is the expected shortage cost if the available supply is 0. Similarly, the last three terms represent the expected holding costs, where the first of these terms is the expected holding cost if the available supply of the  $i^{\text{th}}$  part is between 0 and  $y_i - x_i$ , the second term is the expected holding cost if the available supply is  $y_i - x_i$ , and the third is the expected holding cost if the available supply is 0.

### Properties of the objective function

To obtain the optimal order quantities we wish to show that the total cost function is convex with respect to the order quantities. If this is so, we can find minimizing order quantities by finding where the partial derivative of the total cost function is zero. We now discuss each of these properties in turn.

We begin with the first order optimality condition for  $y_N$ , using Leibnitz's rule to obtain

$$\frac{\partial}{\partial y_N} C(y, x) = (c_N - p_N \bar{G}_N(y_N) + h_N \bar{G}_N(y_N)) \bar{F} \prod_{j=1}^N y_j - x_j; T_N .$$

When written as the product of two terms as we have done, this derivative has a nice interpretation. The first term is the derivative of the cost function for the classical newsboy problem. This term is multiplied by the probability that we can complete our production plan in the time available.

Because of this structure, the first order optimality condition is reduced to  $G_N(y_N) = (p_N - c_N) / (p_N + h_N)$ , the solution to the classical newsboy problem. As before, it is easy to show that if this implies  $y_N < x_N$ , then the optimal  $y_N$  is  $x_N$ . The optimal  $y_N$  should not be dependent on the other  $y_i$ , because once we have produced parts 1, ..., N-1, all we can do is try to minimize the costs for part N. The optimal  $y_N$  should not be dependent on the machine's reliability, because the best thing to do is attempt to achieve the optimal order-up-to quantity exactly.

We now turn to the more difficult task of taking the partial derivative of the total cost function  $C(y, x)$  with respect to  $y_i$  for  $i < N$ . After simplification, the result is

$$(1) \quad \frac{\partial C(y, x)}{\partial y_i} = \bar{F}_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ + \sum_{k=i+1}^N (p_k - c_k) F_{j=1}^k y_j - x_j; T_k - F_{j=1}^{k-1} y_j - x_j; T_k \\ - \sum_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} G_k(u) f_{j=1}^{k-1} y_j - x_j + u - x_k; T_k \, du$$

where the summations from  $i+1$  to  $N$  are taken to be null at  $i = N$ . This expression is easier to interpret if we rewrite it as

$$\frac{\partial C(y, x)}{\partial y_i} = \bar{F}_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ + \sum_{k=i+1}^N (p_k - c_k) F_{j=1}^k y_j - x_j; T_k - F_{j=1}^{k-1} y_j - x_j; T_k \\ + \sum_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} \bar{G}_k(u) f_{j=1}^{k-1} y_j - x_j + u - x_k; T_k \, du$$

$$- \sum_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} f_{y_j - x_j + u - x_k; T_k}^{k-1} du,$$

and then simplify to obtain

$$\begin{aligned} \frac{\partial C(y, x)}{\partial y_i} = & \bar{F}_{y_j - x_j; T_i}^i (c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i)) \\ & - \sum_{k=i+1}^N (c_k + h_k) F_{y_j - x_j; T_k}^k - F_{y_j - x_j; T_k}^{k-1} \\ & + \sum_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} \bar{G}_k(u) f_{y_j - x_j + u - x_k; T_k}^{k-1} du. \end{aligned}$$

The first term is analogous to  $C(y, x)/y_N$  discussed above. The second two terms give the impact of the choice of  $y_i$  on the parts  $k = i+1, \dots, N$ . The first of these terms represents the marginal cost of machine time. The expression in square brackets is the probability that machine output is insufficient to produce up to  $y_k$  but sufficient to start production of part  $k$ . As this probability increases, total cost decreases at rate  $c_k + h_k$ , assuming that the units built are not sold. The final term is the marginal cost of lost sales. The integral represents the expected sales given that machine output is greater than zero but less than  $y_k$ . As this increases, shortage costs are accrued at a rate  $p_k$  and holding costs, which have already been charged in the second term, are avoided at a rate  $h_k$ .

It can be seen from this first order condition that as  $T$  tends to infinity, the optimal  $y_i$  each approach their "newsboy point"  $y_i^N$ , that is, the point where  $G(y_i) = (p_i - c_i) / (p_i + h_i)$ . It should also be evident that the optimal  $y_i$  are never greater than  $y_i^N$ , their respective newsboy points. We now argue this formally by induction. We have already shown that the optimal  $y_N$  is  $y_N^N$ , the newsboy point for part  $N$ . Suppose that

we have shown that the optimal  $y_k$  are not greater than  $y_k^N$  for  $k = i+1, \dots, N$ . We will now show that the optimal  $y_i$  is also less than or equal to  $y_i^N$ . We first require the following result:

$$\begin{aligned} \frac{C(y, x)}{y_i} &= \bar{F} \prod_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ &+ \prod_{k=i+1}^N (p_k - c_k) \prod_{j=1}^k y_j - x_j; T_k - \prod_{j=1}^{k-1} y_j - x_j; T_k \\ &- \prod_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} G_k(u) \prod_{j=1}^{k-1} y_j - x_j + u - x_k; T_k \, du \end{aligned}$$

(from equation (1))

$$\begin{aligned} &\bar{F} \prod_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ &+ \prod_{k=i+1}^N (p_k - c_k) \prod_{j=1}^k y_j - x_j; T_k - \prod_{j=1}^{k-1} y_j - x_j; T_k \\ &- \prod_{k=i+1}^N (p_k + h_k) \int_{x_k}^{y_k} G_k(y_k) \prod_{j=1}^{k-1} y_j - x_j + u - x_k; T_k \, du \end{aligned}$$

(because  $G_k(\cdot)$  is non-decreasing)

$$\begin{aligned} &= \bar{F} \prod_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ &- \prod_{k=i+1}^N \left( c_k - p_k \bar{G}_k(y_k) + h_k G_k(y_k) \right) \prod_{j=1}^k y_j - x_j; T_k - \prod_{j=1}^{k-1} y_j - x_j; T_k \\ &\bar{F} \prod_{j=1}^i y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \end{aligned}$$

(because  $G_k(y_k) = (p_k - c_k) / (p_k + h_k)$  for  $k = i+1, \dots, N$ , by the induction hypothesis).

Using this result, it immediately follows that for any  $y_i > y_i^N$ ,  $C(y, x) / y_i$  is positive. Therefore, if  $x_i < y_i^N$ , the optimal  $y_i$  lies between  $x_i$  and  $y_i^N$ . If  $x_i = y_i^N$ , then it is optimal not to produce (the optimal  $y_i$  equals  $x_i$ ).

We now show that if  $x_k \leq y_k \leq y_k^N$  for  $k = i, i+1, \dots, N$ , then  $\partial^2 C(y, x) / \partial y_i^2$  is non-negative. To show this, we once again use Leibnitz's rule to take the second partial derivative with respect to  $y_i$  to obtain

$$\begin{aligned} \frac{\partial^2}{\partial y_i^2} C(y, x) = & \bar{F} \prod_{j=1}^i (y_j - x_j; T_i) \left( p_i g_i(y_i) + h_i g_i(y_i) \right) \\ & + -f \prod_{j=1}^i (y_j - x_j; T_i) \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\ & + \prod_{k=i+1}^N \left( p_k - c_k \right) \prod_{j=1}^k (y_j - x_j; T_k) - f \prod_{j=1}^{k-1} (y_j - x_j; T_k) \\ & - \left( p_k + h_k \right) \int_{x_k}^{y_k} G_k(u) \frac{1}{y_i} \prod_{j=1}^{k-1} (y_j - x_j + u - x_k; T_k) du . \end{aligned}$$

We now show that this second partial derivative is non-negative. We have written the second partial derivative as the sum of three (square bracketed) terms. The first term can be seen to be non-negative by inspection. The second square bracketed term is non-negative if  $-c_i + p_i \bar{G}_i(y_i) - h_i G_i(y_i)$  is non-negative, which is true if  $G_i(y_i) \leq (p_i - c_i) / (p_i + h_i)$ , which is always true for  $y_i < y_i^N$ . Showing that the third bracketed term is non-negative is slightly more difficult. We note that for each  $k$ ,

$$\begin{aligned} & \left( p_k - c_k \right) \prod_{j=1}^k (y_j - x_j; T_k) - f \prod_{j=1}^{k-1} (y_j - x_j; T_k) \\ & - \left( p_k + h_k \right) \int_{x_k}^{y_k} G_k(u) \frac{1}{y_i} \prod_{j=1}^{k-1} (y_j - x_j + u - x_k; T_k) du \\ & \left( p_k - c_k \right) \prod_{j=1}^k (y_j - x_j; T_k) - f \prod_{j=1}^{k-1} (y_j - x_j; T_k) \\ & - \left( p_k + h_k \right) \int_{x_k}^{y_k} G_k(y_k) \frac{1}{y_i} \prod_{j=1}^{k-1} (y_j - x_j + u - x_k; T_k) du \end{aligned}$$

(because  $G_k(\cdot)$  is non-decreasing)

$$\begin{aligned}
& (p_k - c_k) \int_{j=1}^k f(y_j - x_j; T_k) - \int_{j=1}^{k-1} f(y_j - x_j; T_k) \\
& - (p_k + h_k) \int_{x_k}^{y_k} \frac{(p_k - c_k)}{(p_k + h_k)} \frac{1}{y_i} \int_{j=1}^{k-1} f(y_j - x_j + u - x_k; T_k) du \\
& \hspace{15em} (\text{because } G_k(y_k) = (p_k - c_k) / (p_k + h_k)) \\
& = (p_k - c_k) \int_{j=1}^k f(y_j - x_j; T_k) - \int_{j=1}^{k-1} f(y_j - x_j; T_k) \\
& - (p_k - c_k) \int_{j=1}^k f(y_j - x_j; T_k) - \int_{j=1}^{k-1} f(y_j - x_j; T_k) \\
& = 0. \text{ Q.E.D.}
\end{aligned}$$

Given the other  $y_j, j \neq i$ , this result allows us to find the optimal  $y_i$  by determining if  $y_i \in [x_i, y_i^N]$  such that  $C(y, x) / y_i = 0$ . If such a  $y_i$  exists then it is optimal, otherwise, the optimal policy is not to order. Since  $C(y, x) / y_i$  is a non-decreasing function of  $y_i$  over the range  $[x_i, y_i^N]$  when  $y_k = y_k^N$  for  $k = i+1, \dots, N$ , the optimal  $y_i$  can be found by simple binary search.

Given the above results, after we have found  $y_N$  we can find the other  $y_i$  by solving the above problem as a  $N-1$  dimensional unconstrained minimization problem on the interval  $x_i \leq y_i \leq y_i^N, i = 1, \dots, N-1$ . For an excellent discussion of algorithms to solve such problems, see Bazaraa et al. (1993). An alternative approach is presented on the next few pages.

### Solution algorithm

The difficulty in finding the optimal production quantities is that the first order condition tells us that  $N-1$  of the  $y_i$  are mutually dependent. We now describe a solution procedure that exploits the special structure of these dependencies. In particular, consider the difference

$$\begin{aligned}
\hat{C}_{i+1} &= \frac{C(y, x)}{y_{i+1}} - \frac{C(y, x)}{y_i} \\
&= \bar{F}^{i+1}_{j=1} y_j - x_j; T_{i+1} \left( c_{i+1} - p_{i+1} \bar{G}_{i+1}(y_{i+1}) + h_{i+1} G_{i+1}(y_{i+1}) \right) \\
&\quad - \bar{F}^i_{j=1} y_j - x_j; T_i \left( c_i - p_i \bar{G}_i(y_i) + h_i G_i(y_i) \right) \\
&\quad - (p_{i+1} - c_{i+1}) F^{i+1}_{j=1} y_j - x_j; T_{i+1} - F^i_{j=1} y_j - x_j; T_{i+1} \\
&\quad + (p_{i+1} + h_{i+1}) \int_{x_{i+1}}^{y_{i+1}} G_{i+1}(u) f^i_{j=1} y_j - x_j + u - x_{i+1}; T_{i+1} du.
\end{aligned}$$

Note that if  $y_i$  is optimal,  $C(y, x)/y_i$  is zero, so that  $\hat{C}_{i+1} = C(y, x)/y_{i+1}$ . The reason that this is significant is because  $\hat{C}_{i+1}$  is a function only of  $y_1, \dots, y_i$ . Therefore if the optimal  $y_1$  is known then  $\hat{C}_2$  can be used to find the optimal  $y_2$ , and then  $\hat{C}_3$  can be used to find the optimal  $y_3$ , and so forth.

Since the optimal  $y_1$  is not known, we must use a search technique to find it. We now prove three important properties that will be helpful in this regard.

Let the production quantities that result from the above procedure be denoted by  $\hat{y}_i$ .

We first show that  $\hat{y}_N = y_N^N$  iff  $C(y, x)/y_1 = 0$ . Observe that  $\hat{C}_N$  is exactly equal to

$C(y, x)/y_N - C(y, x)/y_{N-1}$ , and thus  $\hat{y}_N = y_N^N$  iff  $C(y, x)/y_{N-1} = 0$ . Further, for any  $i$ ,  $\hat{C}_{i+1} = C(y, x)/y_{i+1}$  iff  $C(y, x)/y_i = 0$ . Therefore,  $\hat{y}_N = y_N^N$  iff  $C(y, x)/y_1 = 0$ .

The second property is that if the guess for the optimal value of  $y_1$  is too large,  $\hat{y}_N > y_N^N$ . We have shown above that if  $x_k = y_k = y_i^N$  for  $k = i, i+1, \dots, N$ , then  $\partial^2 C(y, x)/y_i^2$

0. Accordingly, if the guess for the optimal value of  $y_1$  is too large,  $C(y, x)/y_1 > 0$ ,

so that in order for  $\hat{C}_2 = 0$ ,  $\hat{y}_2$  must be chosen such that  $C(y, x)/y_2 > 0$ , so that  $\hat{y}_2$  will be greater than the optimal  $y_2$ . Repeating this argument, we see that each  $\hat{y}_i$  will be greater than the optimal  $y_i$ , and thus  $\hat{y}_N > y_N^N$ . By analogous reasoning we can conclude that if the guess for the optimal value of  $y_1$  is too small,  $\hat{y}_N < y_N^N$ .

The third and final property that we wish to show is that  $\hat{C}_{i+1}$  is an increasing function of  $y_{i+1}$ . This property is particularly important, as it allows us to find  $\hat{y}_{i+1}$  by simple binary search. To prove this, we take the partial derivative of  $\hat{C}_{i+1}$  with respect to  $y_{i+1}$  and simplify to obtain

$$\frac{\partial \hat{C}_{i+1}}{\partial y_{i+1}} = \bar{F} \sum_{j=1}^{i+1} y_j - x_j; T_{i+1} (p_{i+1}g_{i+1}(y_{i+1}) + h_{i+1}g_{i+1}(y_{i+1}))$$

which is clearly non-negative since each term is non-negative, and thus the result is proven.

Using these properties, we are now ready to state the following

**Algorithm:**

1. Pre-processing. Compute the  $y_i^N$ . If any  $x_i > y_i^N$  then the optimal  $\hat{y}_i = x_i$  and it is optimal not to produce this part. Remove all such parts from the list of parts to be produced over the horizon.
2. Initialization. Set  $\hat{y}_1 = y_1^N$ . Set  $U = y_1^N$  and  $L = x_1$ .
3. Main loop. For each  $i = 2, \dots, N$ , find the  $\hat{y}_i$  such that  $\hat{C}_i = 0$ . If any  $\hat{y}_i > y_i^N$  then  $\hat{y}_i$  is too large. Set  $U = \hat{y}_i$ ,  $\hat{y}_i = (U + L) / 2$ , and repeat Step 3.
4. Optimality test. If  $|\hat{y}_N - y_N^N| < \epsilon$  then the  $\hat{y}_i$  are optimal. Stop.

5. **Adjustment step.** If  $\hat{y}_N > y_N^N$  then  $\hat{y}_1$  is too large. Set  $U = \hat{y}_1$ ,  $\hat{y}_1 = (U + L) / 2$ , and go to Step 3. If  $\hat{y}_N < y_N^N$  then  $\hat{y}_1$  is too small. Set  $L = \hat{y}_1$ ,  $\hat{y}_1 = (U + L) / 2$ , and go to Step 3.

The algorithm essentially performs a binary search on the guess for the optimal  $y_1$  by maintaining an upper and lower bound (U and L) on the optimal value. The algorithm terminates when the current value of  $\hat{y}_N$  is within some small positive of  $y_N^N$ .

Because the properties that we have proven above are valid only if  $x_i \leq y_i \leq y_i^N$  for  $i = 1, \dots, N$ , we must take care to ensure that this remains true throughout the algorithm. We perform the test in Step 2 to ensure that we do not proceed if any  $y_i > y_i^N$ . We set  $L = x_i$  so that  $\hat{y}_1 \geq x_i$ . Lastly, in a pre-processing step we remove a part  $i$  from consideration if  $x_i > y_i^N$ . We can do this because, for any such part, the optimal  $\hat{y}_i$  is  $x_i$ , and it is thus optimal not to produce that part. Since the part would not be produced, it has no effect on the other parts.

### Dynamic rescheduling

In the development above we have discussed how to determine a set of production quantities to minimize expected total cost. Of course, as the plan is implemented, the reliability of the machine may be much higher or much lower than expected. As a result, if we were given the opportunity to do so, we might adjust the production plan based on what actually happens as time rolls forward.

We now consider how to dynamically update the optimal policy based on the realized output of the machine. One approach would be to repeatedly solve the model as fast as possible with constantly updated information from the factory floor.

Such an approach places a great demand on both computational resources and information systems. We instead propose a simpler method that would allow someone on the shop floor to determine when to stop production of the current part based on the current inventory level. We now describe the optimal dynamic policy for the current part, assuming that the decision maker will follow a static optimal policy for all subsequent parts. In this way, the dynamic solution obtained is only an approximation to the true dynamic optimal policy.

Suppose it is currently time zero, and for any particular future point in time we would like to determine the amount of completed production at or above which it is optimal to stop producing the current part and switch to the next part. We find these critical inventory levels as follows. We feed inputs into the model as if it is now some future point in time. The model is then used to find the optimal production plan as we vary the inventory level of part 1. We have found the critical inventory level when we have found the lowest inventory level such that the optimal decision is not to produce. We then know that at this future point in time if we are at or above this level then we should stop producing part 1. In terms of the mathematical model, this equates to finding the smallest  $x_1$  such that the optimal  $y_1$  is equal to  $x_1$ . If we can do this, then we can trace out a curve that shows this critical inventory level over time. The optimal dynamic operating policy is therefore to produce until the inventory level crosses the curve. Once this happens and production is switched to the next part, the model should be solved again to find the critical inventory level as a function of time for the next part.

Two important details have been omitted from the above discussion. The first involves the existence of such a critical inventory level. Recall that we are only interested in the *lowest*  $x_1$  such that at the optimum  $y_1 - x_1 = 0$ , so the only question

we must answer is whether or not such an  $x_1$  exists. But this is clearly so, since if we set  $x_1 = G_1^{-1}((p_1 - c_1) / (p_1 + h_1))$ , we know  $y_1 = G_1^{-1}((p_1 - c_1) / (p_1 + h_1))$  and since we must constrain  $y_1$  to be at least  $x_1$ ,  $y_1 = x_1$ .

The second detail that needs to be resolved is how to compute the critical inventory level for a future point in time. We now describe a method based on the solution procedure outlined above. Suppose the point in time is  $t_1$  and the current time is  $t_0$ . Then the first step is to update the horizon length by replacing  $T$  with  $T - (t_1 - t_0)$ , set  $x_1 = 0$ , and then solve for the optimal production quantities. We then search over  $x_1$ , at each iteration finding the optimal  $y_1$ , until we identify the *lowest*  $x_1$  such that at the optimum,  $y_1 = x_1$ .

### Impact of overtime opportunities

In the development above we purposely omitted any discussion of how to make optimal overtime decisions. Suppose now that there are  $p = 1, \dots, N_{OT}$  opportunities over the horizon to run overtime, and for simplicity assume that they are each of duration  $OT$  at cost  $c_p$ . In the development above we computed optimal production quantities ignoring overtime opportunities. This is equivalent to assuming that we choose not to run overtime, and the resulting expected cost is the expected cost of this strategy.

Suppose instead that we decide that we are going to run overtime once. To evaluate the expected cost of this strategy we simply replace  $T$  by  $T + OT$  and find the optimal production quantities to compute the minimum expected cost, and then add  $c_p$ . Note that unlike the previous models in this chapter, it does not matter *when* we run overtime, since all overtime opportunities occur before the demand point. Because of this simple fact, we can find the optimal policy by finding the optimal

production quantities  $N_{OT} + 1$  times, with  $T$  taking on the values  $T, T + OT, T + 2 OT, \dots, T + N_{OT} OT$ .

Of course, we expect that the total cost function will be convex in  $OT$  if overtime costs are convex in  $OT$ . If this is true, then the optimal overtime level can be found by a more efficient search procedure. We leave this as a conjecture for now.

### Extension to different machine speeds

For notational convenience, up to this point we have ignored the possibility that the machine operates at different speeds when producing different parts. If the speeds are different, then the requirements on the machine need to be expressed in common units, such as time, instead of parts. This can be accommodated easily, replacing all expressions such as

$$F \sum_{j=1}^i y_j - x_j; T_i \quad \text{and} \quad f \sum_{j=1}^i y_j - x_j; T_i$$

with

$$F \sum_{j=1}^i \frac{y_j - x_j}{P_j}; T_i \quad \text{and} \quad f \sum_{j=1}^i \frac{y_j - x_j}{P_j}; T_i ,$$

where  $P_j$  is the speed at which the machine produces part  $j$  when it is working. Our solution procedure for finding the optimal  $y_i$  is also unchanged.

### References for Chapter 3

- Adshead, N. S. and D. H. R. Price. "Overtime Decision Rule Experiments With a Model of a Real Shop". European Journal of Operational Research, 39(3), pp. 274-283, 1989.
- Bazaraa, Mokhtar S., Hanif D. Sherali and C. M. Shetty. Nonlinear Programming: Theory and Algorithms, 2nd edition. New York: John Wiley and Sons, Inc., 1993.
- Bellman, Richard E. Dynamic Programming. Princeton, NJ: Princeton University Press, 1957.
- Bellman, Richard E. and Stuart E. Dreyfus. Applied Dynamic Programming. Princeton, NJ: Princeton University Press, 1962.
- Bertsekas, Dimitri P. Dynamic Programming: Deterministic and Stochastic Models. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- Beyer, William H., ed. CRC Handbook of Mathematical Sciences, 6th edition. Boca Raton, Florida: CRC Press, 1987.
- Birge, J., J. B. G. Frenk, J. Mittenthal and A. H. G. Rinnooy Kan. "Single-Machine Scheduling Subject to Stochastic Breakdowns". Naval Research Logistics, 37, pp. 661-677, 1990.
- Birge, John R. and Kevin D. Glazebrook. "Assessing the Effects of Machine Breakdowns in Stochastic Scheduling". Operations Research Letters, 7(6), pp. 267-271, 1988.
- Bitran, Gabriel R., Elizabeth A. Haas and Hirofumi Matsuo. "Production Planning of Style Goods with High Setup Costs and Forecast Revisions". Operations Research, 34(2), pp. 226-236, 1986.
- Bitran, Gabriel R. and Devanath Tirupati. "Approximations for Networks of Queues with Overtime". Management Science, 37(3), pp. 282-300, 1991.
- Brooke, Lindsay. "Stamping the Ram". Automotive Industries, September 1993, pp. 77-78.
- Denardo, Eric V. Dynamic Programming: Models and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1982.

- Evans, Richard V. "Inventory Control of a Multiproduct System with a Limited Production Resource". Naval Research Logistics Quarterly, 14(2), pp. 173-184, 1967.
- Federgruen, Awi and Linda Green. "Queueing Systems with Service Interruptions." Operations Research, 34(5), pp. 752-768, 1986.
- Federgruen, Awi and Linda Green. "Queueing Systems with Service Interruptions II". Naval Research Logistics, 35(3), pp. 345-358, 1989.
- Gelders, L. and P. R. Kleindorfer. "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: Part I. Theory". Operations Research, 22(1), pp. 46-60, 1974.
- Gelders, L. and P. R. Kleindorfer. "Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop: II – Computation and Structure". Operations Research, 23(2), pp. 312-324, 1975.
- Gittins, J. C. "Bandit Processes and Dynamic Allocation Indices". Journal of the Royal Statistical Society, Series B, 41, pp. 148-177, 1979.
- Glazebrook, K. D. "Scheduling Stochastic Jobs on a Single Machine Subject to Breakdowns". Naval Research Logistics Quarterly, 31, pp. 251-264, 1984.
- Groenevelt, Harry, Liliane Pintelon and Abraham Seidmann. "Production Batching with Machine Breakdowns and Safety Stocks". Operations Research, 40(5), pp. 959-971, 1992 (a).
- Groenevelt, Harry, Liliane Pintelon and Abraham Seidmann. "Production Lot Sizing with Machine Breakdowns". Management Science, 38(1), pp. 104-123, 1992 (b).
- Kletter, David B. Determining Production Lot Sizes and Safety Stocks for an Automobile Stamping Plant. S. M. Thesis, MIT, June 1994.
- Larson, Robert E. State Increment Dynamic Programming. New York: American Elsevier Pub. Co., 1968.
- Lee, Hau L. and Stephen Nahmias. "Single-Product, Single-Location Models", Chapter 1 in Graves, S. C., A. H. G. Rinnooy Kan and P. H. Zipkin, eds., Logistics of Production and Inventory, Amsterdam: North-Holland, 1993.
- Matsuo, Hirofumi. "The Weighted Total Tardiness Problem with Fixed Shipping Times and Overtime Utilization". Operations Research, 36(2), pp. 293-307, 1988.

- Matsuo, Hirofumi. "A Stochastic Sequencing Problem for Style Goods with Forecast Revisions and Hierarchical Structure". Management Science, 36(3), pp. 332-347, 1990.
- Nemhauser, George L. and Laurence A. Wolsey. Integer and Combinatorial Optimization. New York: John Wiley & Sons, Inc., 1988.
- Pinedo, Michael. Scheduling: Theory, Algorithms, and Systems. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- Pinedo, Michael and Elias Rammouz. "A Note on Stochastic Scheduling on a Single Machine Subject to Breakdown and Repair". Probability in the Engineering and Informational Sciences, 2, pp. 41-49, 1988.
- Reiman, Martin I. and Lawrence M. Wein. "Dynamic Scheduling of a Two-Class Queue with Setups". Working Paper No. 3692-94-MSA. Cambridge, MA: Alfred P. Sloan School of Management, MIT, 1994.
- Rose, John S. "The Newsboy with Known Demand and Uncertain Replenishment: Applications to Quality Control and Container Fill". Operations Research Letters, 11(2), pp. 111-117, 1992.
- Sengupta, Bhaskar. "A Queue with Service Interruptions in an Alternating Random Environment". Operations Research, 38(2), pp. 308-318, 1990.
- Sethi, Suresh P. and Qing Zhang. "Hierarchical Production and Setup Scheduling in a Single Machine System", Chapter 8 in Hierarchical Decision Making in Stochastic Manufacturing Systems. Boston, MA: Birkhäuser, 1994.
- Smith, Stephen A., John C. Chambers and Eli Shlifer. "Optimal Inventories Based on Job Completion Rate for Repairs Requiring Multiple Items". Management Science, 26(8), pp. 849-852.