PART I

QUANTITY-BASED RM

Chapter 2

SINGLE-RESOURCE CAPACITY CONTROL

2.1 Introduction

In this chapter, we examine the problem of quantity-based revenue management for a single resource; specifically, optimally allocating capacity of a resource to different classes of demand. Two prototypical examples are controlling the sale of different fare classes on a single flight leg of an airline and the sale of hotel rooms for a given date at different rate classes. This is to be contrasted with the multiple-resource—or network—problems of Chapter 3, in which customers require a bundle of different resources (such as two connecting flights or a sequence of nights at the same hotel). In reality, many quantity-based RM problems are network RM problems, but in practice, they are still frequently solved as a collection of single-resource problems (treating the resources independently). For this reason, it is important to study single-resource RM models. Moreover, single-resource models are useful as building blocks in heuristics for the network case.

We assume that the firm sells its capacity in n distinct classes[1] that require the same resource. In the airline and hotel context, these classes represent different discount levels with differentiated sale conditions and restrictions. In the early parts of this chapter, we assume that these products appeal to distinct and mutually exclusive segments of the market: the conditions of sale segment the market perfectly into n segments—one for each class. Customers in each segment are eligible for or can afford only the class corresponding to their segment. Later in

[1]In the case of airlines, these are called *fare classes.* Terms like *rate products, rate classes, revenue classes, booking classes* and *fare products* are also used. We shall use the generic term *class* in this chapter.

the chapter, we look at models that do not assume that customers are perfectly segmented, but instead that they choose among the n classes.

The units of capacity are assumed homogeneous, and customers demand a single unit of capacity for the resource. The central problem of the chapter is how to optimally allocate the capacity of the resource to the various classes. This allocation must be done dynamically as demand materializes and with considerable uncertainty about the quantity or composition of future demand. The remainder of the chapter focuses on various models and methods for making these capacity-allocation decisions.

2.1.1 Types of Controls

In the travel industry, reservation systems provide different mechanisms for controlling availability. These mechanisms are usually deeply embedded in the software logic of the reservation system and, as a result, can be quite expensive and difficult to change. Therefore, the control mechanisms chosen for a given implementation are often dictated by the reservation system. The details of reservations systems and the constraints they impose are discussed in greater detail in Chapters 10 and 11. Here, we focus on the control mechanisms themselves.

2.1.1.1 Booking Limits

Booking limits are controls that limit the amount of capacity that can be sold to any particular class at a given point in time. For example, a booking limit of 18 on class 2 indicates that at most 18 units of capacity can be sold to customers in class 2. Beyond this limit, the class would be "closed" to additional class 2 customers. This limit of 18 may be less than the physical capacity. For example, we might want to protect capacity for future demand from class 1 customers.

Booking limits are either *partitioned* or *nested*: A *partitioned booking limit* divides the available capacity into separate blocks (or *buckets*) — one for each class—that can be sold only to the designated class. For example, with 30 units to sell, a partitioned booking limit may set a booking limit of 12 units for class 1, 10 units for class 2, and 8 units for class 3. If the 12 units of class 1 capacity are used up, class 1 would be closed regardless of how much capacity is available in the remaining buckets. This could be undesirable if class 1 has higher revenues than do classes 2 and 3 and the units allocated to class 1 are sold out.

With a *nested booking limit,* the capacity available to different classes overlaps in a hierarchical manner—with higher-ranked classes having access to all the capacity reserved for lower-ranked classes (and perhaps more). Let the nested booking limit for class j be denoted b_j. Then

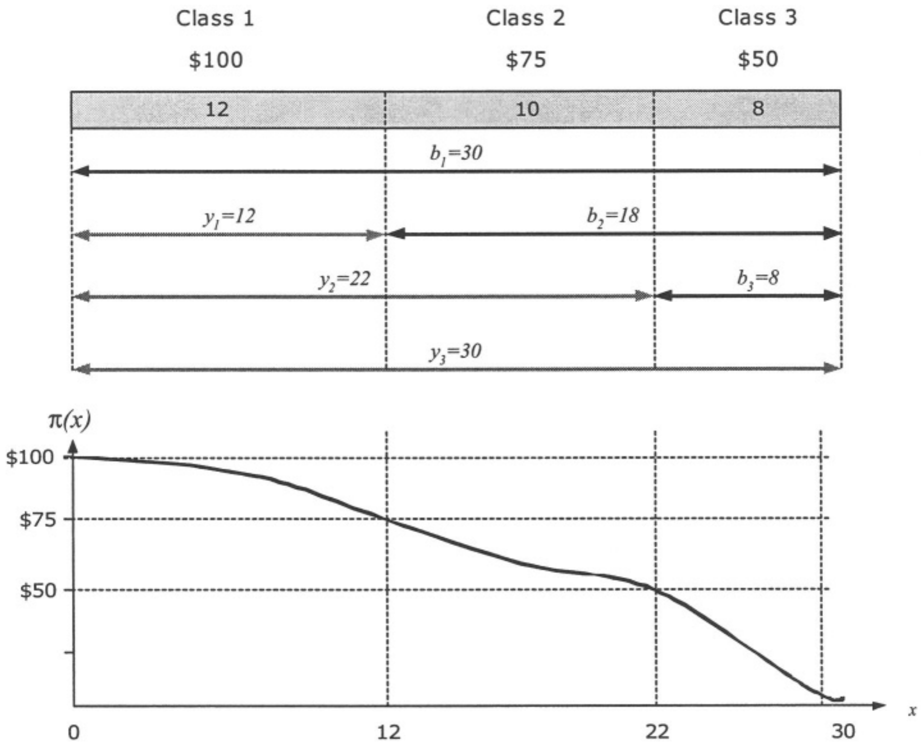


Figure 2.1. **The relationship between booking limits b_j, protection levels y_j, and bid prices $\pi(x)$.**

b_j is the maximum number of units of capacity we are willing to sell to classes j and lower. So in Figure 2.1, the nested booking limit on class 1 and lower (all classes) would be $b_1 = 30$ (the entire capacity), the nested booking limit on classes 2 and 3 combined would be $b_2 = 18$, and the nested booking limit on class 3 alone would be $b_3 = 8$. We would accept at most 30 bookings for classes 1, 2, and 3, at most 18 for classes 2 and 3 combined, and at most 8 for class 3 customers. Effectively, this logic simply allows any capacity "left over" after selling to low classes to become available for sale to higher classes.

Nesting booking limits in this way avoids the problem of capacity being simultaneously unavailable for a high class yet available for lower classes. Most reservations systems that use booking-limit controls quite sensibly use nested rather than partitioned booking limits for this reason.

2.1.1.2 Protection Levels

A *protection level* specifies an amount of capacity to reserve (protect) for a particular class or set of classes. Again, protection levels can be *nested* or *partitioned.* A partitioned protection level is trivially equivalent to a partitioned booking limit; a booking limit of 18 on class 2 sales is equivalent to protecting 18 units of capacity for class 2.

In the nested case, protection levels are again defined for sets of classes—ordered in a hierarchical manner according to class order. Suppose class 1 is the highest class, class 2 the second highest, and so on. Then the protection level j, denoted y_j, is defined as the amount of capacity to save for classes $j, j-1, \ldots, 1$ combined—that is, for classes j and higher (in terms of class order). Continuing our example, we might set a protection level of 12 for class 1 (meaning 12 units of capacity would be protected for sale only to class 1), a protection level of 22 for classes 1 and 2 combined, and a protection level of 30 for classes 1, 2, and 3 combined. (Though frequently no protection level is specified for this last case since it is clear that all the capacity is available to at least one of the classes.)

Figure 2.1 shows the relationship between protection levels and booking limits. The booking limit for class j, b_j is simply the capacity minus the protection level for classes $j-1$ and higher. That is,

$$b_j = C - y_{j-1}, \quad j = 2, \ldots, n,$$

where C is the capacity. For convenience, we define $b_1 = C$ (the highest class has a booking limit equal to the capacity) and $y_n = C$ (all classes combined have a protection level equal to capacity).

2.1.1.3 Standard Versus Theft Nesting

The standard process for using booking limits or nested protection levels proceeds as follows. Starting with C units of capacity, we begin receiving bookings. A bookings for class j is accepted provided (1) there is capacity remaining and (2) the total number of requests accepted for class j to date is less than the booking limit b_j (equivalently, the current capacity remaining is more than the protection level y_{j-1} for classes higher than j). This is called *standard nesting,* and it is the most natural and common way to implement nested-capacity controls.

Another alternative, which is less prevalent though still encountered occasionally in practice, is called *theft nesting.* In theft nesting, a booking in class j not only reduces the allocation for class j but also "steals" from the allocation of all lower classes. So when we accept a request for class j, not only is the class j allocation reduced by one but so are the allocations for classes $j+1, j+2, \ldots, n$. This is equivalent to keeping y_j

units of capacity protected for *future* demand from class j and higher. In other words, even though we just accepted a request for class j, under theft nesting we continue to reserve y_j units for class j and higher, and to do so requires reducing the allocation for classes $j+1, j+2, \ldots, n$. Under standard nesting, in contrast, when we accept a request from class j we effectively reduce by one the capacity we protect for future demand from class j and higher.

The rationale for standard nesting is that the capacity protected for, say, class 1 is based on a forecast of future demand for class 1. Once we observe some demand for class 1, we then *reduce* our estimate of future demand—and hence the capacity we protect for class 1. Standard nesting does this by reducing the capacity protected for future class 1 demand on a one-for-one basis after each arriving request is accepted (and similarly for other classes as well). To illustrate, suppose in our example demand for class 1 is deterministic and equal to the protection level $y_1 = 12$. Then if we receive 5 requests for class 1, we know for certain that future demand for class 1 will be only 7 and hence that it makes sense to reduce the capacity we protect for future demand from 12 to 7, which is precisely what standard nesting does. Theft nesting, in contrast, intuitively corresponds to an assumption of "memorylessness" in demand. In other words, it assumes the demand to date for class 1 does not affect our estimate of *future* demand for class 1. Therefore, we continue to protect y_1 units of capacity for class 1 (and hence must reduce the allocation for classes $2, 3, \ldots, n$).

The two forms of nesting are in fact equivalent if demand arrives strictly in low-to-high class order; that is, the demand for class n arrives first, followed by the demand for class $n-1$, and so on.[2] This is what the standard (static) single-resource models assume, so for these static models, the distinction is not important. However, in practice demand rarely arrives in low-to-high order, and the choice of standard versus theft nesting matters. With mixed order of arrivals, theft nesting protects more capacity for higher classes (equivalently, allocates less capacity to lower classes). Again, however, standard nesting is the norm in RM practice.

2.1.1.4 Bid Prices

What distinguishes bid-price controls from both booking limits and protection levels is that they are revenue-based rather than class-based controls. Specifically, a bid-price control sets a threshold price (which

[2]It is easy to convince oneself of this fact by tracing out the accept/deny decisions under both forms of nesting, and doing so is an instructive exercise.

may depend on variables such as the remaining capacity or time), such that a request is accepted if its revenue exceeds the threshold price and rejected if its revenue is less than the threshold price. Bid-price controls are, in principle, simpler than booking-limit or protection-level controls because they require only storing a single threshold value at any point in time—rather than a set of capacity numbers, one for each class. But to be effective, bid prices must be updated after each sale—and possibly also with time as well—and this typically requires storing a table of bid price values indexed by the current available capacity, current time, or both.

Figure 2.1 shows how bid prices can be used to implement the same nested-allocation policy as booking limits and protection levels. The bid price $\pi(x)$ is plotted as a function of the remaining capacity x. When there are 12 or fewer units remaining, the bid price is over \$75 but less than \$100, so only class 1 demand is accepted. With 13 to 22 units remaining, the bid price is over \$50 but less than \$75 so only classes 1 and 2 are accepted. With more than 22 units of capacity available, the bid price drops below \$50 so all three classes are accepted.

Bid-price control is criticized by some as being "unsafe"—the argument being that having a threshold price as the only control means that the RM system will sell an unlimited amount of capacity to any class whose revenues exceed the bid price threshold. But this is true only if the bid price is not updated. As shown in Figure 2.1, if the bid price is a function of the current remaining capacity, then it performs exactly like a booking limit or protection level, closing off capacity to successively higher classes as capacity is consumed. Without this ability to make bid prices a function of capacity, however, a simple static threshold is indeed a somewhat dangerous form of control.

One potential advantage of bid-price controls is their ability to discriminate based on revenue rather than class. Often (see Section 10.1.3.1) a number of products with different prices are booked in a single class. RM systems then use an average price as the price associated with a class. However, if actual revenue information is available for each request, then a bid-price control can selectively accept only the higher revenue requests in a class, whereas a control based on class designation alone can only accept or reject all requests of a class. Of course, if the exact revenue is not observable at the time of reservation, then this advantage is lost.

2.1.2 Displacement Cost

While the mathematics of optimal capacity controls can become complex, the overriding logic is simple. First, capacity should be allocated to a request if and only if its revenue is greater than the value of the capacity required to satisfy it. Second, the value of capacity should be measured by its (expected) *displacement cost*—or *opportunity cost*—which is the expected loss in future revenue from using the capacity now rather than reserving it for future use.

Theoretically, the displacement-cost idea is captured by using a *value function,* $V(x)$, that measures the optimal expected revenue as a function of the remaining capacity x. The displacement cost is then the difference between the value function at x and the value function at $x - 1$, or $V(x) - V(x-1)$. Much of the theoretical analysis of the capacity controls boils down to analyzing this value function. But conceptually, the logic is simply to compare revenues to displacement costs to make the accept or deny decision.

2.2 Static Models

In this section, we examine one of the first models for quantity-based RM, the so-called *static*[3] single-resource models.

The static model makes several assumptions that are worth examining in some detail. The first is that demand for the different classes arrives in nonoverlapping intervals in the order of increasing prices of the classes.[4] In reality, demand for the different classes may overlap in time. However, the nonoverlapping-intervals assumption is a reasonable approximation (for example, advance-purchase discount demand typically arrives before full-fare coach demand in the airline case). Moreover, the optimal controls that emerge from the model can be applied—at least heuristically—even where demand comes in arbitrary order (using either bid prices or the nesting policies, for example). As for the strict low-before-high assumption, this represents something of a worst-case scenario; for instance, if high-revenue demand arrives before low-revenue demand, the problem is trivial because we simply accept demand first come, first serve.

The second main assumption is that the demands for different classes are independent random variables. Largely, this assumption is made for analytical convenience because to deal with dependence in the demand structure would require introducing complex state variables on the history of observed demand. We can make some justification of the

[3]The term *static* here is somewhat of a misnomer because demand does arrive sequentially over time, albeit in stages ordered from low-revenue to high-revenue demand. However, this term is now standard and helps distinguish this class of models from *dynamic* models that allow arbitrary arrival orders.

[4]Robinson [445] generalizes the static model to the case where demand from each class arrives in nonoverlapping intervals but the order is not necessarily from low to high revenue.

assumption by appealing to the forecast inputs to the model. That is, to the extent that there are systematic factors affecting *all* demand classes (such as seasonalities), these are often reflected in the forecast and become part of the *explained* variation in demand in the forecasting model (for example, as the differences in the forecasted means and variance on different days). The randomness in the single-resource model is then only the residual, unexplained variation in demand. So, for example, the fact that demand for all classes may increase on peak flights does not in itself cause problems provided the increase is predicted by the forecasting method. Still, one has to worry about possible residual dependence in the *unexplained variation* in demand, and this is a potential weakness of the independence assumption.

A third assumption is that demand for a given class does not depend on the capacity controls; in particular, it does not depend on the availability of other classes. Its only justification is if the multiple restrictions associated with each class are so well designed that customers in a high revenue class will not buy down to a lower class and if the prices are so well separated that customers in a lower class will not buy up to a higher class if the lower class is closed. However, neither is really true in practice. There is considerable porousness (imperfect segmentation) in the design of the restrictions, and the price differences between the classes are rarely that dispersed. The assumption that demand does not depend on the capacity controls is therefore a weakness, though in Section 2.6 we look at models that handle imperfect segmentation.

Fourth, the static model suppresses many details about the demand and control process within each of the periods. This creates a potential source of confusion when relating these models to actual RM systems. In particular, the static model assumes an aggregate quantity of demand arrives in a single stage and the decision is simply how much of this demand to accept. Yet in a real reservation system, we typically observe demand sequentially over time, or it may come in batch downloads. The control decision has to be made knowing only the demand observed to date and is usually implemented in the form of prespecified controls uploaded to the reservation system. These details are essentially ignored in the static model. However, fortunately (and perhaps surprisingly), the form of the optimal control is not sensitive to this assumption and can be applied quite independently of how the demand is realized within a period (all at once, sequentially, or in batches). The simplicity and robustness of the optimal control is in fact a central result of the theory for this class of models.

A fifth assumption of the model is that either there are no groups, or if there are group bookings, they can be partially accepted. Group bookings cause significant methodological problems, and these are discussed in Section 2.4.

Finally, the static models assume risk-neutrality. This is a reasonable assumption in practice, since a firm implementing RM typically makes such decisions for a large number of products sold repeatedly (for example, daily flights, daily hotel room stays, and so on). Maximizing the average revenue, therefore, is what matters in the end. While we do not cover this case here, some researchers have recently analyzed the single-resource problem with risk-averse decision makers (Feng and Xiao [187]).

We start with the simple two-class model in order to build some basic intuition and then examine the more general n-class case.

2.2.1 Littlewood's Two-Class Model

The earliest single-resource model for quantity-based RM is due to Littlewood [347]. The model assumes two product classes, with associated prices $p_1 > p_2$. The capacity is C, and we assume there are no cancellations or overbooking. Demand for class j is denoted D_j, and its distribution is denoted by $F_j(\cdot)$. Demand for class 2 arrives first. The problem is to decide how much class 2 demand to accept before seeing the realization of class 1 demand.

The two-class problem is similar to the classic *newsboy problem* in inventory theory, and the optimal decision can be derived informally using a simple marginal analysis: Suppose that we have x units of capacity remaining and we receive a request from class 2. If we accept the request, we collect revenues of p_2. If we do not accept it, we will sell unit x (the marginal unit) at p_1 if and only if demand for class 1 is x or higher. That is, if and only if $D_1 \geq x$. Thus, the expected gain from reserving the x^{th} unit for class 1 (the *expected marginal value*) is $p_1 P(D_1 \geq x)$. Therefore, it makes sense to accept a class 2 request as long as its price exceeds this marginal value, or equivalently, if and only if

$$p_2 \geq p_1 P(D_1 \geq x). \tag{2.1}$$

Note the right-hand side of (2.1) is decreasing in x. Therefore, there will be an optimal protection level, denoted y_1^*, such that we accept class 2 if the remaining capacity exceeds y_1^* and reject it if the remaining capacity is y_1^* or less. Formally, y_1^* satisfies

$$p_2 < p_1 P(D_1 \geq y_1^*) \quad \text{and} \quad p_2 \geq p_1 P(D_1 \geq y_1^* + 1).$$

If a continuous distribution $F_1(x)$ is used to model demand (as is often the case), then the optimal protection level y_1^* is given by the simpler expressions

$$p_2 = p_1 P(D_1 > y_1^*), \quad \text{equivalently,} \quad y_1^* = F_1^{-1}(1 - \frac{p_2}{p_1}), \tag{2.2}$$

which is known as *Littlewood's rule.* Setting a protection level of y_1^* for class 1 according to Littlewood's rule is an optimal policy. Equivalently, setting a booking limit of $b_2^* = c - y_1^*$ on class 2 demand is optimal. Alternatively, we can use a bid-price control with the bid price set at $\pi(x) = p_1 P(D_1 > x)$.

We omit a rigorous proof of Littlewood's rule since it is a special case of a more general result proved below. However, to gain some insight into it, consider the following example:

Example 2.1 Suppose D_1 is normally distributed with mean μ and standard deviation σ. Then by Littlewood's rule, $F_1(y_1^*) = 1 - p_2/p_1$, which implies the optimal protection level can be expressed as

$$y_1^* = \mu + z\sigma,$$

where $z = \Phi^{-1}(1 - p_2/p_1)$ and $\Phi(\cdot)^{-1}$ denotes the inverse of the standard normal c.d.f. Thus, we reserve enough capacity to meet the mean demand for class 1, μ, plus or minus a factor that depends both on the revenue ratio and the demand variation σ. If $p_2/p_1 > 0.5$, the optimal protection level is less than the mean demand; and if $p_2/p_1 < 0.5$, it is greater than the mean demand. In general, the lower the ratio p_2/p_1, the more capacity we reserve for class 1. This makes intuitive sense because we should be willing to take very low prices only when the chances of selling at a high price are lower.

2.2.2 n-Class Models

We next consider the general case of $n > 2$ classes. Again, we assume that demand for the n classes arrives in n stages, one for each class, with classes arriving in increasing order of their revenue values. Let the classes be indexed so that $p_1 > p_2 > \cdots > p_n$. Hence, class n (the lowest price) demand arrives in the first stage (stage n), followed by class $n-1$ demand in stage $n-1$, and so on, with the highest price class (class 1) arriving in the last stage (stage 1). Since there is a one-to-one correspondence between stages and classes, we index both by j. Demand and capacity are most often assumed to be discrete, but occasionally we model them as continuous variables when it helps simplify the analysis and optimality conditions.

2.2.2.1 Dynamic Programming Formulation

This problem can be formulated as a dynamic program in the stages (equivalently, classes), with the remaining capacity x being the state variable. At the start of each stage j, the demand $D_j, D_{j-1}, \ldots, D_1$ has not been realized. Within stage j, the model assumes that the following sequence of events occurs:

1. The realization of the demand D_j occurs, and we observe its value.
2. We decide on a quantity u of this demand to accept. The amount accepted must be less than the capacity remaining, so $u \leq x$. The optimal control u^* is therefore a function of the stage j, the capacity x, and the demand D_j, $u^* = u^*(j, x, D_j)$, though we often suppress this explicit dependence on j, x and D_j in what follows.
3. The revenue $p_j u$ is collected, and we proceed to the start of stage $j-1$ with a remaining capacity of $x - u$.

This sequence of events is assumed for analytical convenience; we derive the optimal control u^* "as if" the decision on the amount to accept is made *after* knowing the value of demand D_j. In reality, of course, demand arrives sequentially over time, and the control decision has to be made *before* observing all the demand D_j. However, it turns out that optimal decisions do not use the prior knowledge of D_j as we show below. Hence, the assumption that D_j is known is not restrictive.

Let $V_j(x)$ denote the value function at the start of stage j. Once the value D_j is observed, the value of u is chosen to maximize the current stage j revenue plus the revenue to go, or

$$p_j u + V_{j-1}(x - u),$$

subject to the constraint $0 \leq u \leq \min\{D_j, x\}$. The value function entering stage j, $V_j(x)$, is then the expected value of this optimization with respect to the demand D_j. Hence, the Bellman equation is[5]

$$V_j(x) = E\left[\max_{0 \leq u \leq \min\{D_j, x\}} \{p_j u + V_{j-1}(x-u)\}\right], \qquad (2.3)$$

with boundary conditions

$$V_0(x) = 0, \quad x = 0, 1, \ldots, C.$$

The values u^* that maximize the right-hand side of (2.3) for each j and x form an optimal control policy for this model.

[5]Readers familiar with dynamic programming may notice that this Bellman equation is of the form E[max{·}] and not max E[·] as in many standard texts. The relationship between these two forms is explained in detail in Appendix D. But essentially, the max E[·] form can be recovered by considering the demand D_j to be a state variable along with x. While the two forms can be shown to be equivalent, the E[max{·}] is simpler to work with in many RM problems. In our case, this leads to the modeling assumption that we optimize "as if" we observed D_j.

2.2.2.2 Optimal Policy: Discrete Demand and Capacity

We first consider the case where demand and capacity are discrete. To analyze the form of the optimal control in this case, define

$$\Delta V_j(x) \equiv V_j(x) - V_j(x-1).$$

$\Delta V_j(x)$ is the *expected marginal value of capacity* at stage j—the expected incremental value of the x^{th} unit of capacity. A key result concerns how these marginal values change with capacity x and the stage j (See Appendix 2.A for proof.):

PROPOSITION 2.1 *The marginal values $\Delta V_j(x)$ of the value function $V_j(x)$ defined by (2.3) satisfy $\forall x, j$:*
(i) $\Delta V_j(x+1) \leq \Delta V_j(x)$
(ii) $\Delta V_{j+1}(x) \geq \Delta V_j(x)$.

That is, at a given stage j the marginal value is decreasing in the remaining capacity, and at a given capacity level x the marginal value increases in the number of stages remaining. These two properties are intuitive and greatly simplify the control. To see this, consider the optimization problem at stage $j+1$. From (2.3) and the definition of $\Delta V_j(x)$, we can write

$$V_{j+1}(x) = V_j(x) + E\left[\max_{0\leq u\leq \min\{D_{j+1},x\}}\left\{\sum_{z=1}^{u}(p_{j+1} - \Delta V_j(x+1-z))\right\}\right],$$

where we take the summation above to be empty if $u = 0$. Since $\Delta V_j(x)$ is decreasing in x by Proposition 2.1(i), it follows that the terms in the sum $p_{j+1} - \Delta V_j(x+1-z)$ are decreasing in z. Thus, it is optimal to increase u (keep adding terms) until the terms $p_{j+1} - \Delta V_j(x+1-z)$ become negative or the upper bound $\min\{D_{j+1}, x\}$ is reached, whichever comes first.

The resulting optimal control can be expressed in terms of optimal protection levels y_j^* for $j, j-1, \ldots, 1$ (class j and higher in the revenue order) by

$$y_j^* \equiv \max\{x : p_{j+1} < \Delta V_j(x)\}, \quad j = 1, \ldots, n-1. \tag{2.4}$$

(Recall the optimal protection level $y_n^* \equiv C$ by convention.) The optimal control at stage $j+1$ is then

$$u^*(j+1, x, D_{j+1}) = \min\{(x - y_j^*)^+, D_{j+1}\}, \tag{2.5}$$

where the notation $z^+ = \max\{0, x\}$ denotes the positive part of z. The quantity $(x - y_j^*)^+$ is the remaining capacity in excess of the protection

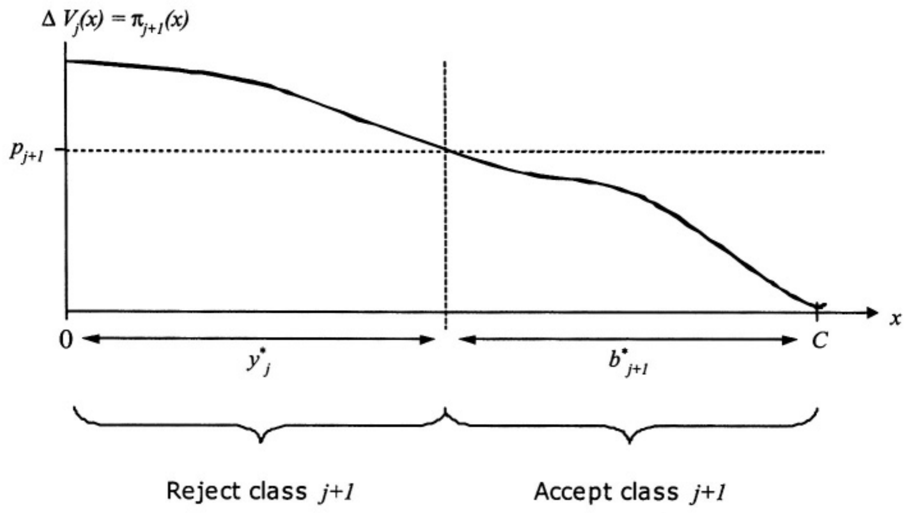


Figure 2.2. The optimal protection level y_j^* in the static model.

level, which is the maximum capacity we are willing to sell to class $j+1$. The situation is shown in Figure 2.2.

In practice, we can simply post the protection level y_j^* in a reservation system and accept requests first come, first serve until the capacity threshold y_j^* is reached or the stage ends, whichever comes first. Thus, the optimal protection-level control at stage $j+1$ requires no information about the demand D_{j+1}, yet it produces the same optimal decision "as if" we knew D_{j+1} exactly at the start of stage $j+1$. The reason for this is that knowledge of D_{j+1} does not affect the future value of capacity, $V_j(x)$. Deciding to accept or reject each request simply involves comparing current revenues to the marginal cost of capacity, and this comparison does not depend on how many stage-$(j+1)$ requests there are in total.

Proposition 2.1(ii) implies the nested protection structure

$$y_1^* \leq y_2^* \leq \cdots \leq y_n^*.$$

This fact is easily seen from Figure 2.2. If p_{j+1} increases with j and the curve $\Delta V_j(x)$ decreases with j, then the optimal protection level y_j^* will shift to the left (decrease). Together, this ordering produces the nested protection-level structure.

One can also use booking limits in place of protection levels to achieve the same control. Optimal nested booking limits are defined by

$$b_j^* \equiv C - y_{j-1}^*, \quad j = 2, \ldots, n, \tag{2.6}$$

with $b_1^* \equiv C$. The optimal control in stage $j+1$ is then to accept

$$u^*(j+1, x, D_{j+1}) = \min\{(b_{j+1} - (C - x))^+, D_{j+1}\}.$$

Note that $C - x$ is the total capacity sold prior to stage $j+1$ and b_{j+1} is the booking limit for class $j+1$, so $(b_{j+1} - (C-x))^+$ is the remaining capacity available for class $j+1$. The optimal booking limit is also shown in Figure 2.2.

Finally, the optimal control can also be implemented through a table of bid prices. Indeed, if we define the stage $j+1$ bid price by

$$\pi_{j+1}(x) \equiv \Delta V_j(x), \tag{2.7}$$

then the optimal control is

$$u^*(j+1, x, D_{j+1}) = \begin{cases} 0 & \text{if } p_{j+1} < \pi_{j+1}(x) \\ \max\{z : p_{j+1} \geq \pi_{j+1}(x - z)\} & \text{otherwise.} \end{cases}$$

In words, we accept the z^{th} request in stage $j+1$ if the price p_{j+1} exceeds the bid price value $\pi_{j+1}(x - z)$ of the z^{th} unit of capacity that is allocated. In practice, we can store a table of bid prices and process requests by sequentially comparing the price of each product to the table values corresponding to the remaining capacity.

We summarize these results in the following theorem:

THEOREM 2.1 *For the static model defined by (2.3), the optimal control can be achieved using either*
(i) Nested protection levels defined by (2.4),
(ii) Nested booking limits defined by (2.6), or
(iii) Bid price tables defined by (2.7).

How to compute these various policies is discussed in Section 2.2.3.

2.2.2.3 Optimality Conditions for Continuous Demand

Next, consider the case where capacity is continuous and demand at each stage has a continuous distribution. In this case, the dynamic program is still given by (2.3); however D_j, x, and u are now continuous quantities. The analysis of the dynamic program is slightly more complex than it is in the discrete-demand case, but many of the details are quite similar. Hence, we only briefly describe the key differences.

The main change is that the marginal value $\Delta V_j(x)$ is now replaced by the derivative of $V_j(x)$ with respect to x, $\frac{\partial}{\partial x} V_j(x)$. This derivative is still interpreted as the marginal expected value of capacity. And an argument nearly identical to that in the proof of Proposition 2.1 shows that the marginal value $\frac{\partial}{\partial x} V_j(x)$ is decreasing in x (equivalently, $V_j(x)$ is concave in x).

Therefore, the optimal control in stage $j+1$ is to keep increasing u (keep accepting demand) as long as

$$p_{j+1} \geq \frac{\partial}{\partial x} V_j(x-u)$$

and to stop accepting once this condition is violated or the demand D_{j+1} is exhausted, whichever comes first. Again, this decision rule can be implemented with optimal protection levels, defined by

$$y_j^* \equiv \max\{x : p_{j+1} < \frac{\partial}{\partial x} V_j(x)\}, \quad j = 1, \ldots, n-1.$$

One of the chief virtues of the continuous model is that it leads to simplified expressions for the optimal vector of protection levels $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)$. We state the basic result without proof (see Brumelle and McGill [91] for a proof).

First, for an arbitrary vector of protection levels $\mathbf{y}$ and vector of demands $\mathbf{D} = (D_1, \ldots, D_n)$, define the following $n-1$ *fill events*

$$\begin{aligned} B_j(\mathbf{y}, \mathbf{D}) \quad \equiv \quad & \{D_1 > y_1, D_1 + D_2 > y_2, \\ & \ldots, D_1 + \cdots + D_j > y_j\}, \quad j = 1, \ldots, n-1. \end{aligned} \tag{2.8}$$

$B_j(\mathbf{y}, \mathbf{D})$ is the event that demand to come in stages $1, 2, \ldots, j$ exceeds the corresponding protection levels. A necessary and sufficient condition for $\mathbf{y}^*$ to be an optimal vector of protection levels is that it satisfy the $n-1$ equations

$$P(B_j(\mathbf{y}^*, \mathbf{D})) = \frac{p_{j+1}}{p_1}, \quad j = 1, 2, \ldots, n-1. \tag{2.9}$$

That is, the j^{th} fill event should occur with probability equal to the ratio of class $(j+1)$ revenue to class 1 revenue. As it should, this reduces to Littlewood's rule (2.2) in the $n = 2$ case, since $P(B_1(\mathbf{y}^*, \mathbf{D})) = P(D_1 > y_1^*) = p_2/p_1$.

Note that

$$B_j(\mathbf{y}, \mathbf{D}) = B_{j-1}(\mathbf{y}, \mathbf{D}) \cap \{D_1 + \cdots + D_j > y_j\},$$

so the event $B_j(\mathbf{y}, \mathbf{D})$ can occur only if $B_{j-1}(\mathbf{y}, \mathbf{D})$ occurs. Also, if $y_j = y_{j-1}$ then $B_j(\mathbf{y}, \mathbf{D}) = B_{j-1}(\mathbf{y}, \mathbf{D})$. Thus, if $p_j < p_{j-1}$, we must have $y_j^* > y_{j-1}^*$ to satisfy (2.9). Thus, the optimal protection levels are strictly increasing in j if the revenues are strictly decreasing in j.

2.2.3 Computational Approaches

At first glance it may appear that the optimal nested allocations are difficult to compute. However, computing these values is in fact quite easy and efficient algorithmically. There are two basic approaches: dynamic programming and Monte Carlo integration.

2.2.3.1 Dynamic Programming

The first approach is based on using the dynamic programming recursion (2.3) directly and requires that demand and capacity are discrete—or in the continuous case that these quantities can be suitably discretized. The inner optimization in (2.3) is simplified by using the optimal protection levels y_{j-1}^* from the previous stage. Thus, substituting (2.5) into (2.3) we obtain the recursion

$$V_j(x) = E\left[p_j \min\{D_j, (x - y_{j-1}^*)^+\} + V_{j-1}(x - \min\{D_j, (x - y_{j-1}^*)^+\})\right], \tag{2.10}$$

where y_j^* is determined using (2.4), and we define $y_0^* = 0$. This procedure is repeated starting from $j = 1$ and working backward to $j = n$.

For discrete-demand distributions, computing the expectation in (2.10) for each state x requires evaluating at most $O(C)$ terms since $\min\{D_j, (x - y_{j-1}^*)^+\} \leq C$. Since there are C states (capacity levels), the complexity at each stage is $O(C^2)$. The critical values y_j^* can then be identified from (2.4) in $\log(C)$ time by binary search as $\Delta V_j(x)$ is nonincreasing. Indeed, since we know $y_j^* \geq y_{j-1}^*$, the binary search can be further constrained to values in the interval $[y_{j-1}^*, C]$. Therefore, computing y_j^* does not add to the complexity at stage j. Since these steps must be repeated for each of the $n-1$ stages (stage n need not be computed as mentioned above), the total complexity of the recursion is $O(nC^2)$.

2.2.3.2 Monte Carlo Integration

The second approach to computing optimal protection levels is based on using (2.9) together with Monte Carlo integration. It is most naturally suited to the case of continuous demand and capacity, though the discrete case can be computed (at least heuristically) with this method as well.

The idea is to simulate a large number K of demand vectors, $\mathbf{d}^k = (d_1^k, \ldots, d_n^k)$, $k = 1, \ldots, K$, from the forecast distributions for the n classes. We then progressively sort through these values to find thresholds $\mathbf{y}$ that approximately satisfy (2.9).

In what follows, it is convenient to note that

$$P(B_j(\mathbf{y}, \mathbf{D})) = P(D_1 + \cdots + D_j > y_j | B_{j-1}(\mathbf{y}, \mathbf{D}))P(B_{j-1}(\mathbf{y}, \mathbf{D}))$$

Thus, (2.9) implies that the optimal $\mathbf{y}^*$ must satisfy

$$\begin{aligned} P(D_1 + \cdots + D_j > y_j^* | B_{j-1}(\mathbf{y}^*, \mathbf{D})) &= \frac{1}{P(B_{j-1}(\mathbf{y}^*, \mathbf{D}))} \frac{p_{j+1}}{p_1} \\ &= \frac{p_{j+1}}{p_j}, \quad j = 1, \ldots, n-1, \end{aligned}$$

since $P(B_{j-1}(\mathbf{y}^*, \mathbf{D})) = p_j/p_1$. The following algorithm computes the optimal $\mathbf{y}^*$ approximately using the empirical conditional probabilities estimated from the sample of simulated demand data:

```
STEP 0: Generate and store $K$ random demand vectors $\mathbf{d}^k = (d_1^k, \ldots, d_n^k)$.
    For $k = 1, \ldots, K$ and $j = 1, \ldots, n-1$, compute the partial sums

        $S_j^k = d_1^k + d_2^k + \cdots + d_j^k$

    and form the vector $\mathbf{S}^k = (S_1^k, \ldots, S_n^k)$.
    Initialize a list $\mathcal{K} = \{1, \ldots, K\}$ and counter $j = 1$.

STEP 1: Sort the vectors $\mathbf{S}^k, k \in \mathcal{K}$ by their jth component values, $S_j^k$.
    Let $[l]$ denote the lth element of $\mathcal{K}$ in this sorted list so that

        $S_j^{[1]} \le S_j^{[2]} \le \cdots \le S_j^{[|\mathcal{K}|]}$.

STEP 2: Set $l^* = \lfloor \frac{p_{j+1}}{p_j} |\mathcal{K}| \rfloor$. Set $y_j = \frac{1}{2}(S_j^{[l^*]} + S_j^{[l^*+1]})$.

STEP 3: Set $\mathcal{K} \leftarrow \{k \in \mathcal{K} : S_j^k > y_j\}$, and $j \leftarrow j + 1$.
    IF $j = n - 1$ STOP
    ELSE GOTO STEP 1.
```

The complexity of this method is $O(nK \log K)$, which is nearly linear in the number of simulated demand vectors K. Thus, it is relatively efficient even with large samples. It is also quite simple to program and can be used with any general distribution. The following example illustrates the method:

Example 2.2 Consider a three-class example, where the prices are $p_1 = 100$, $p_2 = 70$, and $p_3 = 42$. The demand for each class is normally distributed. Class 1 has a mean of 20 and standard deviation of 9; class 2 has a mean of 45 and standard deviation of 12. (The statistics for class 3 do not affect the calculation.)

Figure 2.3 shows a plot of the partial sums $S_1 = D_1$ and $S_2 = D_1 + D_2$ for 50 simulated data points of this problem. Since the first ratio $p_2/p_1 = 70/100 = 0.7$, the Monte Carlo algorithm starts by finding a value y_1 such that 70% of these points (35 points) have S_1 values above y_1. The result is shown in Figure 2.3 by the vertical line at $y_1 = 16.3$.

In the next iteration of the algorithm, the 35 points to the right of this vertical line are sorted again by their S_2 value and y_2 is chosen so that a fraction $p_3/p_2 = 42/70 = 0.6$ of the points (21 points) lie above y_2. This occurs at $y_2 = 68.3$.

The algorithm then terminates with the estimates $y_1^* \approx 16.3$ and $y_2^* \approx 68.3$.

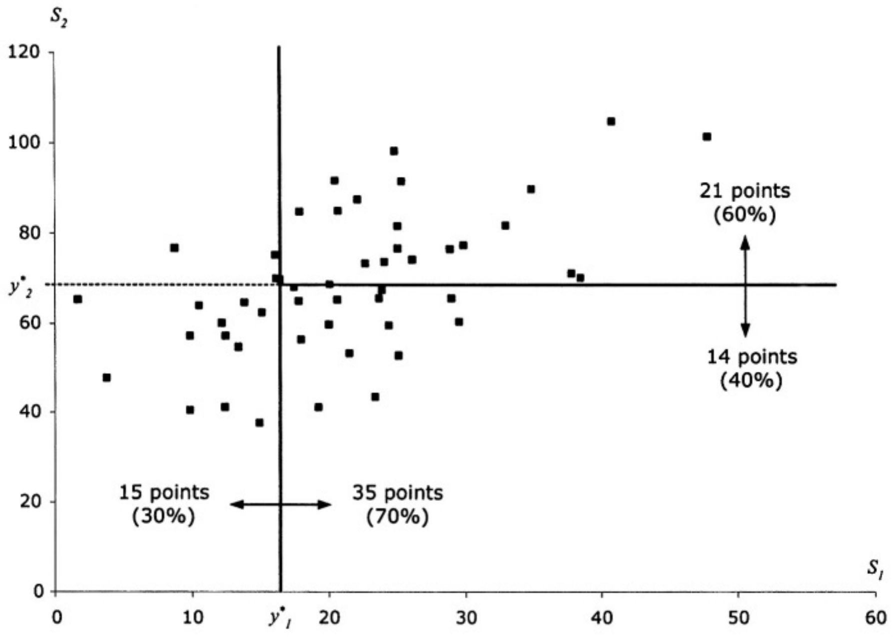


Figure 2.3. A plot of the partial sums S_1, S_2 and resulting Monte Carlo estimates of optimal protection levels for fifty simulated data points.

2.2.4 Heuristics

As we have seen, computing optimal controls for the static single-resource model is not particularly difficult. Despite this fact, exact optimization models are not widely used in practice. Indeed, most single-resource airline RM systems use one of several heuristics to compute booking limits and protection levels.

There are two main reasons for this state of affairs. The first is simply a case of practice being one step ahead of the underlying theory. As mentioned, in the airline industry the practice of using capacity controls to manage multiple classes quickly gained popularity following deregulation in the mid 1970s. But this predates the theory of optimal controls by more than a decade. The only known optimal controls in the 1970s were Littlewood's results for the two-class problem. As a result, heuristics were developed for the general n-class problem. During the decade following deregulation, RM software embedded these heuristics, and people grew accustomed to thinking in terms of them. The inertia generated from this early use of the heuristics is one reason for their continued popularity today.

Heuristics are also widely used because they are simpler to code, quicker to run, and generate revenues that in many cases are close to optimal. Indeed, many practitioners in the airline industry simply believe that even the modest effort of computing optimal controls is not worth the benefit they provide in improved revenue performance. Proponents of heuristics argue that the potential improvement from getting better revenue data and improving demand forecasts swamps the gains from using optimal controls—reflecting the philosophy that it is better to be "approximately right" than it is to be "precisely wrong."

While these points are well taken, such criticisms are somewhat misdirected. For starters, using optimal controls does not mean one has to give up on improvements in other areas, such as forecasting. These activities are not mutually exclusive, though a understaffed development group might very well consider refining optimization modules a low-priority task. Yet given the very modest cost of coding and computing optimal controls, the strong objections to the use of optimal controls are often not entirely rational.

Regardless of one's view on the use of heuristics, it is important to understand them. They remain widely used in practice and can also help develop useful intuition.

We next look at the two most popular heuristics: EMSR-a and EMSR-b, both of which are attributed to Belobaba [38–40]. Both heuristics are based on the n-class, static, single-resource model defined above in Section 2.2. They differ only in how they approximate the problem. Static model assumptions apply: classes are indexed so that $p_1 > p_2 > \cdots > p_n$, $F_j(x)$ denotes the c.d.f. of class j demand, and low-revenue demand arrives before high-revenue demand in stages that are indexed by j as well. Moreover, for ease of exposition we assume that capacity and demand are continuous and that the distribution functions $F_j(x)$, $j = 1, \ldots, n$, are continuous as well, though these assumptions are easily relaxed.

2.2.4.1 EMSR-a

EMSR-a *(expected marginal seat revenue–version a)* is the most widely publicized heuristic for the single-resource problem. Despite this fact, it is less popular in practice than its close cousin, EMSR-b, which surprisingly is not well documented in the literature. Generally, EMSR-b provides better revenue performance, and it is certainly more intuitive, though EMSR-a is important to know just the same.

EMSR-a is based on the idea of adding the protection levels produced by applying Littlewood's rule to successive pairs of classes. Consider stage $j+1$ in which demand of class $j+1$ arrives with price p_{j+1}. We are interested in computing how much capacity to reserve for the remaining classes, $j, j-1, \ldots, 1$; that is, the protection level, y_j, for classes j and higher. To do so, let's consider a single class k among the remaining classes $j, j-1, \ldots, 1$ and compare k and $j+1$ *in isolation.* Considering only these two classes, we would use Littlewood's rule (2.2) and reserve capacity y_k^{j+1} for class k, where

$$P(D_k > y_k^{j+1}) = \frac{p_{j+1}}{p_k}. \tag{2.11}$$

Repeating for each future class $k = j, j-1, \ldots, 1$, we could likewise compute how much capacity to reserve for each such class k in isolation. The idea of EMSR-a, then, is simply to add up these individual protection levels to approximate the total protection level y_j for classes j and higher. That is, set the protection level y_j as

$$y_j = \sum_{k=1}^{j} y_k^{j+1}, \tag{2.12}$$

where y_k^{j+1} is given by (2.11). One then repeats this same calculation for each stage j.

EMSR-a is certainly simple and has an intuitive appeal. For a short while it was even believed to be optimal, but this notion was quickly dispelled once the published work on optimal controls appeared.

Indeed, it is not hard to see intuitively that EMSR-a can be excessively conservative and produce protection levels that are larger than optimal in certain cases. This is because adding the individual protection levels y_k^{j+1} ignores the statistical averaging effect *(pooling effect)* produced by aggregating demand across classes. For example, for the sake of illustration, suppose that at stage $j+1$ all future demand has the same revenue, i.e., $p_j = p_{j-1} = \cdots = p_1 = p$. Then EMSR-a will set protection levels so that

$$P(D_k > y_k^{j+1}) = \frac{p_{j+1}}{p}, \quad k = 1, \ldots, j.$$

However, it is clear that all these future classes should be aggregated since they have identical revenues, in which case we can apply Littlewood's rule (2.2) to obtain the optimal protection level, y_j^*, using

$$P(\sum_{k=1}^{j} D_k > y_j^*) = \frac{p_{j+1}}{p}.$$

Since for any random variables and any numbers y_k^{j+1}

$$P(\sum_{k=1}^{j} D_k > \sum_{k=1}^{j} y_k^{j+1}) \leq \sum_{k=1}^{j} P(D_k > y_k^{j+1}),$$

the optimal protection level y_j^* is less than the EMSR-a protection level, $\sum_{k=1}^{j} y_k^{j+1}$.[6] This behavior suggests that EMSR-a may perform badly when there are large numbers of classes whose revenues are close together.

2.2.4.2 EMSR-b

EMSR-b (*expected marginal seat revenue–version b*) is an alternative single-resource heuristic that avoids the lack-of-pooling defect in EMSR-a mentioned above. EMSR-b is again based on an approximation that reduces the problem at each stage to two classes, but in contrast to EMSR-a, the approximation is based on aggregating *demand* rather than aggregating *protection levels.* Specifically, the demand from future classes is aggregated and treated as one class with a revenue equal to the weighted-average revenue.

Consider stage $j+1$ in which we want to determine protection level y_j. Define the aggregated future demand for classes $j, j-1, \ldots, 1$ by

$$S_j = \sum_{k=1}^{j} D_k,$$

and let the weighted-average revenue from classes $1, \ldots, j$, denoted $\bar{p}_j$, be defined by

$$\bar{p}_j = \frac{\sum_{k=1}^{j} p_k E[D_k]}{\sum_{k=1}^{j} E[D_k]}. \tag{2.13}$$

Then the EMSR-b protection level for class j and higher, y_j, is chosen by Littlewood's rule (2.2) so that

$$P(S_j > y_j) = \frac{p_{j+1}}{\bar{p}_j}. \tag{2.14}$$

It is common when using EMSR-b to assume demand for each class j is independent and normally distributed with mean μ_j and variance σ_j^2, in

[6]To see this, consider a sample realization of demand, and note that if $\sum_{k=1}^{j} D_k > \sum_{k=1}^{j} y_k^{j+1}$, then $D_k > y_k^{j+1}$ for at least one k but the converse need not be true. So the probability of the event $\sum_{k=1}^{j} D_k > \sum_{k=1}^{j} y_k^{j+1}$ cannot exceed the sum (over $k = 1, \ldots, j$) of the probabilities of the events $D_k > y_k^{j+1}$.

which case

$$y_j = \mu + z_\alpha \sigma,$$

where $\mu = \sum_{k=1}^{j} \mu_k$ is the mean and $\sigma^2 = \sum_{k=1}^{j} \sigma_k^2$ is the variance of the aggregated demand to come at stage $j+1$ and $z_\alpha = \Phi^{-1}(1 - p_{j+1}/\bar{p}_j)$ (recall $\Phi^{-1}(x)$ is the inverse of the standard normal c.d.f.). Again, one repeats this calculation for each j. (See Section 11.A for approximation formulas for the normal and inverse normal distributions.)

EMSR-b clearly captures the *pooling*—or *statistical averaging*—effect that is lacking in EMSR-a. This is an advantage of EMSR-b over EMSR-a. However, using the weighted-average revenue is a somewhat crude approximation that can distort the resulting protection levels. In practice EMSR-b is more popular and seems to generally perform better than EMSR-a, though studies comparing the two have at times shown mixed results. Belobaba [41] reports studies in which EMSR-b is consistently within 0.5 percent of the optimal revenue, whereas EMSR-a can deviate by nearly 1.5 percent from the optimal revenue in certain cases, though with mixed order of arrival and frequent reoptimization, he reports that both methods perform well. However, another recent study by Polt [425] using Lufthansa airline data showed more mixed performance, with neither method dominating the other.

2.2.4.3 Numerical Examples

A few simple numerical examples give some sense of the protection levels and revenues produced by these two approximations. The example we consider is based on a slightly modified instance of the data reported by Wollmer [576]:

Example 2.3 There are four classes, and demand is assumed to be normally distributed. Table 2.1 shows the demand data and Table 2.2 the protection levels produced by EMSR-a, EMSR-b, and the optimal policy.

Table 2.1. **Static single-resource model and protection levels.**

		Demand Statistics		*Protection Levels (y_j)*		
j	p_j	μ_j	σ_j	*OPT*	*EMSR-a*	*EMSR-b*
1	1050	17.3	5.8	16.7	16.7	16.7
2	567	45.1	15.0	42.5	38.7	50.9
3	534	39.6	13.2	72.3	55.6	83.1
4	520	34.0	11.3			

Table 2.2. Revenue performance for Example 2.3.

		OPT	*EMSR-a*		*EMSR-b*	
C	*DF*	*Rev.*	*Rev.*	*% Sub. Opt.*	*Rev.*	*% Sub. Opt.*
80	1.70	49,666	49,515	0.30%	49,463	0.41%
90	1.51	54,846	54,721	0.23%	54,560	0.52%
100	1.36	60,063	59,985	0.13%	59,786	0.46%
110	1.24	65,112	65,078	0.05%	64,881	0.35%
120	1.13	69,916	69,904	0.02%	69,764	0.22%
130	1.05	73,975	73,973	0.00%	73,898	0.10%
140	0.97	77,177	77,174	0.00%	77,143	0.04%
150	0.91	79,544	79,544	0.00%	79,534	0.01%

Note from Table 2.1 that (in this example) there is a considerable discrepancy between the protection levels computed under the heuristics, both compared with each other and with the optimal protection levels.

The revenue performance of the methods from a simulation study are shown in Table 2.2. Capacity is varied from 80 to 150 to create demand factors (ratio of total mean demand to capacity) in the range 1.7 to 0.9. The percentage suboptimality is also reported (one minus the ratio of heuristic revenues to optimal revenues). Note for this example that EMSR-a is slightly better than EMSR-b, though both perform quite well; the suboptimality gap of EMSR-b reaches a high of 0.52%, while the maximum suboptimality of EMSR-a is only 0.30%.

Example 2.4 The demand statistics are the same as in Example 2.3, but the revenue values are more evenly spaced. The data and resulting protection levels are shown in Tables 2.3 and 2.4.

Table 2.3. Static single-resource model data for Example 2.4.

		Demand Statistics		*Protection Levels (y_j)*		
j	p_j	μ_j	σ_j	*OPT*	*EMSR-a*	*EMSR-b*
1	1050	17.3	5.8	9.7	9.8	9.8
2	950	45.1	15.0	54.0	50.4	53.2
3	699	39.6	13.2	98.2	91.6	96.8
4	520	34.0	11.3			

The revenue performance of the heuristics in Example 2.4 is shown in Table 2.4. In this case, there is less discrepancy among the protection levels computed under the heuristics and the optimal policy. The performance of both heuristics is also very good, especially under EMSR-b, which is for all practical purposes optimal. Performance such as this helps explain why these heuristics are popular in practice.

Table 2.4. Revenue performance for Example 2.4.

		OPT	*EMSR-a*		*EMSR-b*	
C	*DF*	*Rev.*	*Rev.*	*% Sub. Opt.*	*Rev.*	*% Sub. Opt.*
80	1.70	67,512	67,462	0.07%	67,516	-0.01%
90	1.51	74,003	73,950	0.07%	74,000	0.00%
100	1.36	79,429	79,164	0.33%	79,426	0.00%
110	1.24	84,884	84,554	0.39%	84,862	0.03%
120	1.13	89,879	89,668	0.23%	89,875	0.00%
130	1.05	95,054	94,899	0.16%	95,045	0.01%
140	0.97	99,072	99,004	0.07%	99,068	0.00%
150	0.91	102,346	102,339	0.01%	102,346	0.00%

2.3 Adaptive Methods

We next examine an adaptive algorithm for determining optimal protection levels for the static, single-resource model of Section 2.2.2. The optimal protection levels in this case are determined by the conditions (2.8). Typically, application of the optimality conditions (2.8) requires three steps. First, historical demand data are studied to determine suitable models for the demand distributions. Second, forecasting techniques are applied to estimate the parameters of these distributions. Third, the forecasts are passed to an optimization routine that solves for protection levels $\mathbf{y}^*$. The resulting controls are then used to make individual accept or deny decisions as reservations come in. In practice, bookings from similar resources are fed back into the forecasting system, and the process is repeated cyclically over time.

In this section, we look at a method for *directly* updating booking policy parameters for the next resource usage based on observations of the performance of the parameters on previous instances, without recourse to the complex cycles of forecasting and optimization. We show how to construct a simple adjustment scheme of this sort that is based on stochastic approximation methods (the Robbins-Monro algorithm [443]) and that provably converges to an optimal policy with repeated application. The convergence, however, is guaranteed only for the case of stationary, independent demand and may be quite slow, requiring a large number of adjustments to reach a near-optimal set of protection levels.

2.3.1 Adaptive Algorithm

Our starting point in developing an adaptive algorithm is condition (2.9), which states that for $\mathbf{y}^*$ to be an optimal set of protection levels, it must satisfy

$$P(B_j(\mathbf{y}^*, \mathbf{D})) = \frac{p_{j+1}}{p_1}, \quad j = 1, 2, \ldots, n-1.$$

For example, if the revenue ratio $\frac{p_{j+1}}{p_1}$ is 0.6, then this condition says that the fill event $B_j(\mathbf{y}, \mathbf{D})$ defined by (2.8) should occur on 60% of the service instances, on average, if $\mathbf{y} = \mathbf{y}^*$. Note that it is easy to determine the frequency of the fill events $B_j(\mathbf{y}, \mathbf{D})$ from historical records, since it is necessary only to observe if demand reached the protection levels, not the degree to which it exceeded them.[7]

To develop the algorithm, for $j = 1, \ldots, n-1$ define

$$H_j(\mathbf{y}, \mathbf{D}) = \frac{p_{j+1}}{p_1} - \mathbf{1}(B_j(\mathbf{y}, \mathbf{D})),$$

where $\mathbf{1}(E)$ denotes the indicator function of event E (a function that is 1 if event E occurs and is zero otherwise). The quantity $H_j(\mathbf{y}, \mathbf{D})$ will be negative if the j^{th} fill event occurs and positive otherwise. If protection levels are being adjusted, an occurrence of the j^{th} fill event (all of classes 1 through j reached their protection levels) suggests that the protection level y_j should be adjusted upward. Thus $-H_j(\mathbf{y}, \mathbf{D})$ can be viewed as an *adjustment direction* for protection level y_j. The corresponding *adjustment vector* is $\mathbf{H}(\mathbf{y}, \mathbf{D}) = (H_1(\mathbf{y}, \mathbf{D}), \ldots, H_{n-1}(\mathbf{y}, \mathbf{D}))$. Define

$$h_j(\mathbf{y}) = \frac{p_{j+1}}{p_1} - P(B_j(\mathbf{y}, \mathbf{D})) = E[H_j(\mathbf{y}, \mathbf{D})], \quad j = 1, \ldots, n-1,$$

and let $\mathbf{h}(\mathbf{y}) = (h_1(\mathbf{y}), \ldots, h_{n-1}(\mathbf{y}))$. Thus, $-\mathbf{h}(\mathbf{y})$ can be properly viewed as the *expected adjustment vector* for protection levels given current levels $\mathbf{y}$. The optimality condition (2.9) stipulates that we should seek a $\mathbf{y}^*$ such that the expected adjustment for all protection levels is zero; or, $\mathbf{h}(\mathbf{y}^*) = \mathbf{0}$.

The Robbins-Monro [443] algorithm (generalized for vector quantities) constructs a sequence of parameter estimates, $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots\}$, from a sequence of independent instances, $\{\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \ldots\}$, using the recursion

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} - \gamma_k \mathbf{H}(\mathbf{y}^{(k)}, \mathbf{D}^{(k)}), \tag{2.15}$$

[7]There are two important exceptions to this statement. First, if y_j happens to exceed the maximum number of seats available for sale (usually the physical capacity plus an overbooking *pad*), then the event $D_1 + \cdots + D_j > y_j$ is not observable (unless the rejected sales are recorded). Second, if protection levels are revised during the booking period prior to the usage of the resource, it can easily happen that a new protection level exceeds the *remaining* capacity (a problem similar to the first one) that earlier, high protection levels constrained demand during part of the booking period in one or more discount classes. In this case, total demand is not observed (a variant of censorship of the demand data).

where γ_k is a sequence of nonnegative step sizes satisfying

$$\sum_k \gamma_k = +\infty \quad \text{and} \quad \sum_k \gamma_k^2 < +\infty. \tag{2.16}$$

The simplest example of a suitable step size sequence is defined by $\gamma_k = \frac{1}{k}$; however, this simple averaging sequence takes small steps early in the procedure, which can delay convergence. In the development to follow, we use a sequence of the form $a/(k+b)$, where a and b are constants chosen to give larger early steps.

The directions $\mathbf{H}(\mathbf{y}^{(k)}, \mathbf{D}^{(k)})$ can be determined after the completion of each instance k (each departure in the airline case). If the j^{th} fill event occurs, $H_j = p_{j+1} - 1 < 0$, and the protection level $y_j^{(k)}$ is increased by $\gamma_k(1 - p_{j+1})$; if not, then $H_j = p_{j+1} > 0$, and y_j^k is reduced by $\gamma_k p_{j+1}$. Thus protection levels are stepped up when high demand is observed and stepped down when low demand is observed, with the step size becoming smaller as the algorithm progresses.

Some relatively mild regularity conditions ensure that the procedure (2.15) does converge (a.s.) to a value $\mathbf{y}^*$ satisfying $h(\mathbf{y}^*) = 0$. (See van Ryzin and McGill [526] for exact conditions as well as bounds on the rate of convergence.)

2.3.2 A Numerical Comparison with EMSR and Censored Forecasting

We next look at a brief numerical example of the performance of the adaptive algorithm compared with a procedure that combines censored forecasting with EMSR-b protection levels. These comparisons are based on simulated data in an idealized stationary setting, but do provide some insight into the performance of each method.

The combined forecasting/EMSR-b (F/EMSR) scheme constructs a demand forecast from censored data based on the Kaplan-Meier estimator of the survivor function $S(x) = P(D > x)$. (See Section 9.4.3.) Protection levels are set using EMSR-b.

The test problem has four classes (three protection levels). Demand is modeled using a normal distribution. The distribution data, along with optimal and EMSR-b protection levels, are shown in Table 2.5. The protection level y^* is the optimal level when demand is normally distributed, while y-EMSR is the protection level computed by the EMSR-b heuristic. To illustrate convergence, starting protection levels are set far from optimal, corresponding to cases of very high and very low starting values (Table 2.6). There are two demand scenarios. The high-demand scenario has a starting inventory of 124 seats, corresponding to a 125% demand factor (ratio of expected total demand to capacity) and approx-

Table 2.5. Fares, demand statistics, and protection levels for adaptive-method numerical examples.

Class	*Fare ($)*	*Mean*	*Std. Dev.*	*y-EMSR*	y^*
1	1,050	17.3	5.8	16.7	16.7
2	567	45.1	15.0	51.5	44.6
3	527	73.6	17.4	131.4	134.0
4	350	19.8	6.6	n.a.	n.a.

Table 2.6. Starting values of protection levels for adaptive-method numerical examples.

	y_1	y_2	y_3
Low	0	15	65
High	35	110	210

imately a 95% load factor (ratio of average number of seats sold to capacity) under optimal protection levels. The low-demand scenario starts with 164 seats, resulting in a demand factor of 95% and a load factor of approximately 90% under optimal protection levels. For the stochastic approximation procedure, the step size sequence is $\gamma_k = 200/(10+k)$.

Figure 2.4 shows three graphs of the data for the case of low demand and high starting values. The top graph of Figure 2.4 shows the average cumulative revenue as a percentage of the optimal revenue for the two methods as a function of the number of iterations (flights). The error bars show the 95% confidence intervals about these averages. The middle graph shows the average protection levels over time for the stochastic approximation (SA) procedure. The horizontal dotted lines are the optimal protection levels. The lowest line corresponds to y_1^*, the middle line to y_2^*, and the top line to y_3^*. The solid lines are the corresponding average protection levels produced by the stochastic approximation (SA) method. The error bars on the solid lines give the 25-percentile and 75-percentile values for each protection level at each iteration, which provides some sense of the variability in protection levels across sample paths. The bottom graph shows the identical plot of protection levels for the F/EMSR method.

Figure 2.4 shows that the F/EMSR procedure converges more quickly than the SA procedure. In this case, the faster convergence of the F/EMSR has a significant impact on the cumulative revenue performance: F/EMSR generates about 2 to 3% higher revenue on average in the early iterations. With low demand and low starting values, the two methods perform comparably.

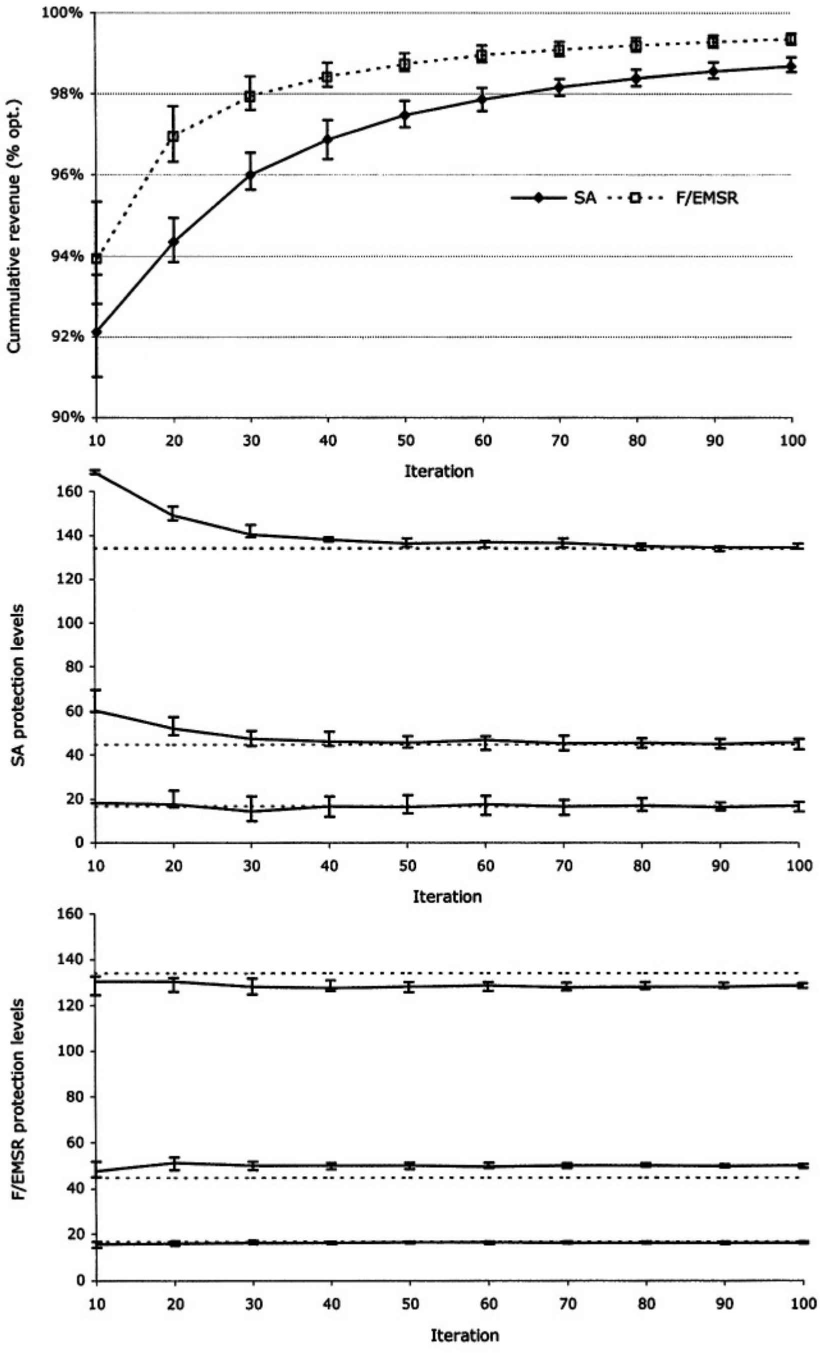


Figure 2.4. Adaptive-method example: low demand, high start, normal distribution.

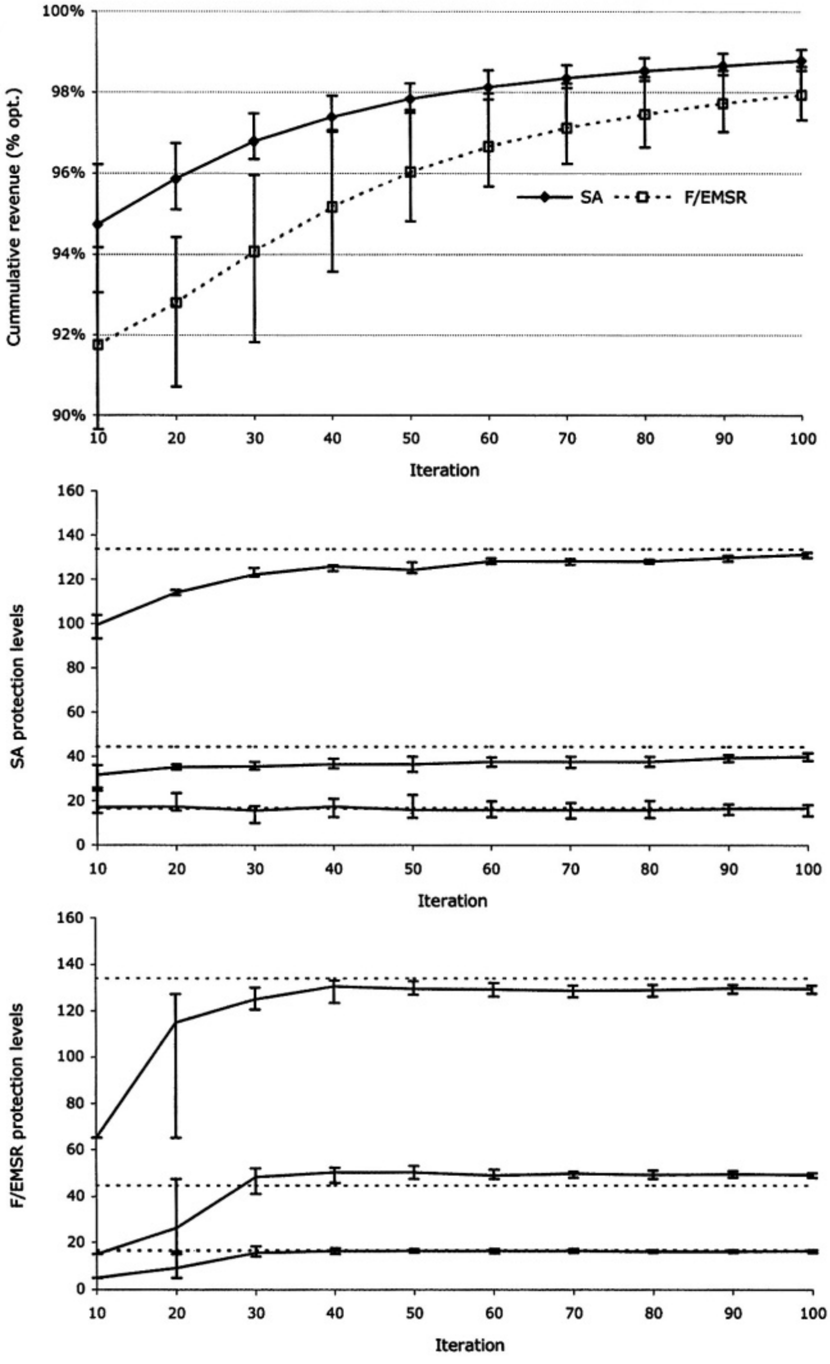


Figure 2.5. Adaptive-method example: high demand, low start, normal distribution.

The results are quite different in the high demand factor case. Figure 2.5 shows the simulation results for low starting protection values and high load factor. Note, as indicated by the error bars in the bottom graphs, that the F/EMSR procedure is very volatile and somewhat slow to converge in the early iterations. The revenue effect of this behavior is quite significant, with F/EMSR generating cumulative revenues roughly 8% lower than optimal and 2 to 3% lower than SA in the early iterations. However, the performance and protection levels of F/EMSR improve after about 30 iterations. In contrast, the SA procedure is considerably more stable, and it converges faster in the early iterations, which accounts for its superior revenue performance. F/EMSR performs badly in this case because the forecasting procedure suffers from the frequent censoring caused by a combination of low protection levels and high demand. For high starting protection levels, the F/EMSR performs better, since there is less initial censoring of data.

This behavior suggests that adaptive methods may be useful as a means of automatically adjusting protection levels in cases where very little demand information is available (such as with new products) and forecasting is difficult due to a high degree of censoring. In such cases, adaptive methods provide a robust way to adjust protection levels and may also help speed up the forecasting method itself by nudging protection levels in the right direction, thereby reducing the amount of censoring.

2.4 Group Arrivals

Group arrivals can pose additional complications. A group request is a single request for multiple units of capacity (such as a family of four traveling together). We briefly describe this case but omit any detailed formulations because the basic ideas follow readily from what we have seen thus far (and the more complicated ideas are beyond the scope of this text).

If groups can be *partially accepted*—that is, given a request for $m \geq 1$ units, we can sell any quantity u in the range $0 \leq u \leq m$ (and more important, the customer is willing to buy any amount in this range, something that is not unusual among tour operators)—then there is little impact on the single-resource models discussed above. Indeed, the static model (2.3) can be thought of as a group model where in each period one "large group" of size D_j arrives because we can sell u units, where $0 \leq u \leq \min\{D_j, x\}$ and x is the total available capacity. Thus, with groups that can be partially accepted, we need only to keep track of the aggregate demand for each class and the formulations are essentially the same as those of the static case in Section 2.2.2.

The real complication in group arrivals occurs when groups must be accepted on an *all-or-none* basis—that is, given a request for $m > 1$ units we can sell only all m units or none at all. This seemingly modest change has a profound impact on the structure of optimal allocation policies. First, we must specify the distribution of group sizes to model how much demand we have from groups of various sizes. But this in itself does not pose too much of a theoretical difficulty. The bigger problem is that the value function may not be concave (the marginal value of capacity may in fact increase), so using booking limits, protection levels, or bid prices may not be optimal.

An example illustrates what can go wrong. Consider a static model with only two stages. Suppose that in the last stage (stage $j = 1$), only groups of size two arrive. In the first stage (stage $j = 2$), groups of varying sizes can arrive. Now suppose that we have $x = 2$ units of capacity remaining. Note the marginal value of the last unit of capacity in stage 1 is zero; that is, $\Delta V_1(1) = V_1(1) - V_1(0) = 0$, because we only get demand for groups of size 2 in the last stage, and therefore having only one unit of capacity is of no value. On the other hand, provided we have some positive probability of demand for a group of size 2 in stage 1, then the second unit of capacity will have a positive marginal value; that is, $\Delta V_1(2) = V_1(2) - V_1(1) > 0$. Hence, the marginal value of capacity $\Delta V_j(x)$ is no longer decreasing in x. As a result, in first stage it can be optimal to reject a request for a single unit of some class when there are two units of capacity remaining but optimal to accept the same request when there is only one unit of capacity remaining. So the notion that there is a booking limit above which we will not sell to a class is no longer valid.

Essentially, the requirement to completely accept or reject groups creates a combinatorial *(bin-packing)* phenomenon in allocating capacity. The resulting nonmonotone value function means that optimal policies are considerably more complex than in the case where groups can be partially accepted. Intuitively, one might expect that if a sufficiently large fraction of demand is from small groups (size one or two) and the capacities are reasonably large, then these combinatorial effects could be ignored and the nongroup models may be a good approximation. This is the implicit assumption in most single-resource RM systems used in practice.

2.5 Dynamic Models

Dynamic models relax the assumption that the demand for classes arrives in a strict low-to-high revenue order. Instead, they allow for an arbitrary order of arrival, with the possibility of interspersed arrivals of several classes. While at first this seems like a strict generalization of the static case, the dynamic models require the assumption of Markovian (such as Poisson) arrivals to make them tractable. This puts restrictions on modeling different levels of variability in demand. Indeed, this limitation on the distribution of demand is the main drawback of dynamic models in practice. In addition, dynamic models require an estimate of the pattern of arrivals over time (called the *booking curve*), which may be difficult to estimate in certain applications. Thus, the choice of dynamic versus static models essentially comes down to a choice of which set of approximations is more acceptable and what data is available in any given application.

Other assumptions of the static model are retained. Demand is assumed independent between classes and over time and also independent of the capacity controls. The firm is again assumed to be risk-neutral. The justifications (or criticisms) for these assumptions are the same as in the static-model case.

2.5.1 Formulation and Structural Properties

In the simplest dynamic model, we have n classes as before with associated prices $p_1 \geq p_2 \geq \cdots \geq p_n$. There are T total periods and t indexes the periods, with the time index running forward ($t = 1$ is the first period, and $t = T$ is the last period; this is in contrast to the static dynamic program, where the stages run from n to 1 in the dynamic programming recursion). Since there is no longer a one-to-one correspondence between periods and classes, we use separate indices—t for periods and j for classes.

In each period we assume, by a sufficiently fine discretization of time, that at most one arrival occurs.[8] The probability of an arrival of class j in period t is denoted $\lambda_j(t)$. The assumption of at most one arrival per period implies that we must have

$$\sum_{j=1}^{n} \lambda_j(t) \leq 1.$$

In general, the periods need not be of the same duration. For example, early in the booking process when demand is low we might use a period of several days whereas during periods of peak booking activity we might use a period of less than an hour. Note also the arrival probabilities may vary with t, so the mix of classes that arrive may vary over time. In

[8] The assumption of one arrival per period can be generalized as shown by Lautenbacher and Stidham [330], but both theoretically and computationally it is a convenient assumption.

particular, we do not require lower classes to arrive earlier than higher classes.

2.5.1.1 Dynamic Program

As before, let x denote the remaining capacity and $V_t(x)$ denote the value function in period t. Let $R(t)$ be a random variable, with $R(t) = p_j$ if a demand for class j arrives in period t and $R(t) = 0$ otherwise. Note that $P(R(t) = p_j) = \lambda_j(t)$. Let $u = 1$ if we accept the arrival (if there has been one) and $u = 0$ otherwise. (We suppress the period subscript t of the control as it should be clear from the context.) We want to maximize the sum of current revenue and the revenue to go, or

$$R(t)u + V_{t+1}(x - u).$$

The Bellman equation is therefore

$$\begin{aligned} V_t(x) &= E\left[\max_{u \in \{0,1\}} \{R(t)u + V_{t+1}(x-u)\}\right] \\ &= V_{t+1}(x) + E\left[\max_{u \in \{0,1\}} \{(R(t) - \Delta V_{t+1}(x))\, u\}\right], \qquad (2.17) \end{aligned}$$

where $\Delta V_{t+1}(x) = V_{t+1}(x) - V_{t+1}(x-1)$ is the expected marginal value of capacity in period $t+1$. The boundary conditions are [9]

$$V_{T+1}(x) = 0, \quad x = 0, 1, \ldots, C,$$

and

$$V_t(0) = 0, \quad t = 1, \ldots, T.$$

2.5.2 Optimal Policy

An immediate consequence of (2.17) is that if a class j request arrives, so that $R(t) = p_j$, then it is optimal to accept the request if and only if

$$p_j \geq \Delta V_{t+1}(x).$$

Thus, the optimal control can be implemented using a bid-price control where the bid price is equal to the marginal value,

$$\pi_t(x) = \Delta V_t(x). \qquad (2.18)$$

[9]The second boundary condition can be eliminated if we use the control constraint $u \in \{0, \min\{1, x\}\}$ instead of $u \in \{0, 1\}$. However, it is simpler conceptually and notationally to use the $x = 0$ boundary conditions instead.

Revenues that exceed this threshold are accepted; those that do not are rejected.

As in the static case, an important property of the value function is that it has decreasing marginal value $\Delta V_t(x) = V_t(x) - V_t(x-1)$. (See Appendix 2.A for proof.)

PROPOSITION 2.2 *The increments $\Delta V_t(x)$ of the value function $V_t(x)$ defined by (2.17) satisfy $\forall x, t$:*
(i) $\Delta V_t(x+1) \leq \Delta V_t(x)$,
(ii) $\Delta V_{t+1}(x) \leq \Delta V_t(x)$.

This theorem is natural and intuitive since one would expect the value of additional capacity at any point in time to have a decreasing marginal benefit and the marginal value at any given remaining capacity x to decrease with time (because as time elapses, there are fewer opportunities to sell the capacity).

As a consequence, the optimization on the right-hand side of (2.17) can also be implemented as a nested-allocation policy, albeit one that has time-varying protection levels (or booking limits). Specifically, we can define time-dependent optimal protection levels

$$y_j^*(t) = \max\{x : p_{j+1} < \Delta V_{t+1}(x)\}, \quad j = 1, 2, \ldots, n-1 \qquad (2.19)$$

that have the usual interpretation that $y_j^*(t)$ is the capacity we protect for classes $j, j-1, \ldots, 1$. Then the protection levels are nested, $y_1^*(t) \leq y_2^*(t) \leq \cdots \leq y_{n-1}^*(t)$, and it is optimal to accept class j if and only if the remaining capacity exceeds $y_{j-1}^*(t)$. The situation is illustrated in Figure 2.6.

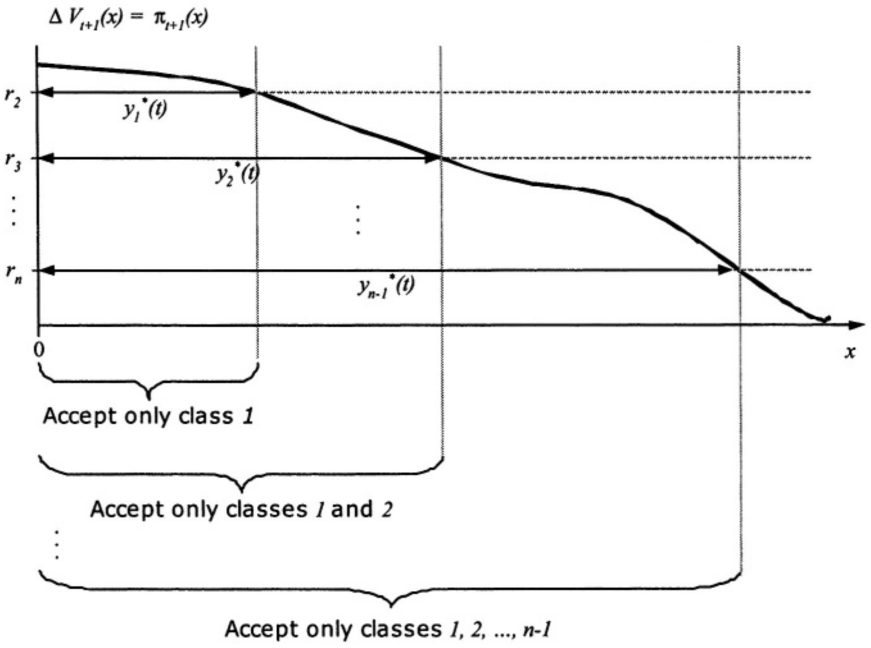
Time-dependent nested booking limits can also be defined as before by

$$b_j^*(t) \equiv C - y_{j-1}^*(t), \quad j = 2, \ldots, n, \qquad (2.20)$$

That the booking limits and protection levels depend on time in this case essentially stems from the fact that the demand to come varies with time in the dynamic model. The change in demand to come as time evolves effects the opportunity cost and therefore the resulting booking limit and protection levels.

As a practical matter, because the value function is not likely to change much over short periods of time, fixing the protection levels or booking limits computed by a dynamic model and updating them periodically (as is done in most RM systems in practice) is usually close to optimal. Still, the time-varying nature of the protection levels remains a key distinction between static and dynamic models.

We summarize these results in the following theorem:



$\Delta V_{t+1}(x) = \pi_{t+1}(x)$
r_2
r_3
r_n
$y_1^*(t)$
$y_2^*(t)$
$y_{n-1}^*(t)$
0
x
Accept only class 1
Accept only classes 1 and 2
Accept only classes 1, 2, ..., n-1

Figure 2.6. **Optimal protection level $y_j^*(t)$ in the dynamic model.**

THEOREM 2.2 For *the dynamic model defined by (2.17), the optimal control can be achieved using either:*
(i) Time-dependent nested protection levels defined by (2.19),
(ii) Time-dependent nested booking limits defined by (2.20), or
(iii) Bid-price tables defined by (2.18).

2.5.2.1 Computation

Computationally, the dynamic model is solved by substituting the optimal policy into (2.17). This yields the recursion

$$\begin{aligned} V_t(x) &= V_{t+1}(x) + E\left[(R(t) - \Delta V_{t+1}(x))^+\right] \\ &= \sum_{j=1}^{n} \lambda_j(t)(p_j - \Delta V_{t+1}(x))^+, \quad t = 1, \ldots, T. \end{aligned}$$

Starting with the boundary condition $V_{T+1}(x) = 0$, $\forall x$, we proceed with the recursion backward in time t. Each stage t requires $O(nC)$ operations, so the overall complexity is $O(nCT)$. Usually, the value of T is $O(C)$ because in most practical problems the total expected demand is the same magnitude as the capacity and the periods are typically chosen so that there is $O(1)$ arrival per period. So the complexity in terms of

n and C is approximately $O(nC^2)$, which is the same as that of the dynamic program for the static model.

2.6 Customer-Choice Behavior

A key assumption in the models that we have described thus far is that demand for each of the classes is completely independent of the capacity controls being applied by the seller. That is, it is assumed that the likelihood of receiving a request for any given class does not depend on which other classes are available at the time of the request. Needless to say, this is a somewhat unrealistic assumption. For example, in the airline case the likelihood of selling a full-fare ticket may very well depend on whether a discount fare is available at the same time and the likelihood that a customer buys at all may depend on the lowest available fare. When customers buy a higher fare when a discount is closed it is called *buy-up* (from the firm's point of view, this is also called *sell-up*); when they choose another flight when a discount is closed it is called *diversion.*

Clearly, such customer behavior could have important RM consequences and ought to be considered when making control decisions. We next look at some heuristic and exact methods for incorporating customer-choice behavior in single-resource problems.

2.6.1 Buy-Up Factors

One approach to modeling customer-choice behavior that works with the two-class model is to include buy-up probabilities—also called *buy-up factors*—in the formulation.

The approach works as follows. Consider the simple two-class static model, and recall that Littlewood's rule (2.2) (slightly restated) is to accept demand from class 2 if and only if

$$p_2 \geq p_1 P(D_1 > x), \tag{2.21}$$

where x is the remaining capacity—that is, if the revenue from accepting class 2 exceeds the marginal value of the unit of capacity required to satisfy the request. Now suppose that there is a probability q that a customer for class 2 will buy class 1 if class 2 is closed. The net benefit of accepting the request is still the same, but now rather than losing the request when we reject it, there is some chance the customer will buy up to class 1. If so, we earn a net benefit of $p_1 - p_1 P(D_1 > x)$ (the class 1 revenue minus the expected marginal cost). Thus, it is optimal to accept class 2 now if $p_2 - p_1 P(D_1 \geq x) \geq q p_1 (1 - P(D_1 > x))$ or equivalently if

$$p_2 \geq (1 - q) p_1 P(D_1 > x) + q p_1. \tag{2.22}$$

Note that the right-hand side of the modified rule (2.22) is strictly larger than the right-hand side in Littlewood's rule (2.21), which means that the modified rule (2.22) is more likely to reject class 2 demand. This is intuitive because with the possibility of customers upgrading to class 1, we should be more eager to close class 2.

The difficulty with this approach is that it does not extend to more than two classes—at least not in an exact way—because the probability that a customer buys class i given that class j is closed depends not only on i and j but also on which other classes are also available. In other words, with more than two classes the customer faces a *multinomial* choice rather than a *binary* choice.

However, one can at least heuristically extend the buy-up factor idea to EMSR-a or EMSR-b because these heuristics approximate the multi-class problem using the two-class model.

For example, EMSR-b can be extended to allow for a buy-up factor by modifying the equation for determining the protection level y_j, (2.14), as follows:

$$p_{j+1} = (1 - q_{j+1})\bar{p}_j P(S_j > y_j) + q_{j+1}\hat{p}_{j+1}, \qquad (2.23)$$

where q_{j+1} is the probability that a customer of class $j+1$ buys up to one of the classes $j, j-1, \dots, 1$; $\bar{p}_j$ is the weighted-average revenue from these classes as defined by (2.13); and $\hat{p}_{j+1} > p_{j+1}$ is an estimate of the average revenue received given that a class $j+1$ customer buys up to one of the classes $j, j-1, \dots, 1$ (for example, $\hat{p}_{j+1} = p_j$ if customers are assumed to buy up to the next-highest price class). Again, the net result of this change is to increase the protection level y_j and close down class $j+1$ earlier than one would do under the traditional EMSR-b rule.[10]

While this modification to EMSR-b provides a simple heuristic way to incorporate choice behavior, it is a somewhat ad hoc adjustment to an already heuristic approach to the problem. Beyond the limitations of the model and its assumptions, there are some serious difficulties involved in estimating the buy-up factors. Indeed, in current applications of the model, they are often simply made-up, reasonable-sounding numbers. Moreover, the assumptions of the model can clash with unconstraining and recapture procedures that are subsequently applied, resulting in double counting of demand. Despite these limitations, buy-up factors have proved useful as a rough-cut approach for incorporating choice behavior in practice.

[10]That it increases the protection level about the usual EMSR-b value can be seen by noting that $p_{j+1} = \bar{R}_j P(S_j > y_j)$ in the usual EMSR-b case and $\hat{p}_{j+1} > p_{j+1}$; thus, y_j has to increase to satisfy the equality (2.23).

2.6.2 Discrete-Choice Models

We next look at a single-resource problem in which customer-choice behavior is explicitly modeled using a general discrete-choice model. In contrast to the heuristic approach of buy-up factors, this model provides a more theoretically sound approach to incorporating choice behavior. It also provides insights into how choice behavior affects the optimal availability controls. The theory is first developed for the general choice model case and then applied to some special demand models.

2.6.2.1 Model Definition

As in the traditional dynamic model of Section 2.5, time is discrete and indexed by t, with the indices running forward in time ($t = T$ is the period of resource usage). In each period there is at most one arrival. The probability of arrival is denoted by λ, which we assume, for ease of exposition, is the same for all time-periods t. There are n classes, and we let $\mathcal{N} = \{1, \ldots, n\}$ denote the entire set of classes. We let choice index 0 denote the no-purchase choice; that is, the event that the customer does not purchase any of the classes offered. Each class $j \in \mathcal{N}$ has an associated price p_j, and without loss of generality we index classes so that $p_1 \geq p_2 \geq \cdots \geq p_n \geq 0$. We let $p_0 = 0$ denote the revenue of the no-purchase choice.

Customer purchase behavior is modeled as follows. In each period t, the seller chooses a subset $S_t \subseteq \mathcal{N}$ of classes to offer. When the set of classes S_t is offered in period t, the probability that a customer chooses class $j \in S_t$ is denoted $P_j(S_t)$. $P_0(S_t)$ denotes the no-purchase probability.

The probability that a sale of class j is made in period t is therefore $\lambda P_j(S_t)$, and the probability that no sale is made is $\lambda P_0(S_t) + (1 - \lambda)$. Note that this last expression reflects the fact that having no sales in a period could be due either to no arrival at all or an arrival that does not purchase. This leads to an incomplete-data problem when estimating the model, as discussed in Section 9.4.1.2.

The only condition we impose on the choice probabilities $P_j(S)$ is that they define a proper probability function. That is, for every set $S \subseteq \mathcal{N}$, the probabilities satisfy

$$\begin{aligned} P_j(S) &\geq 0, \quad \forall j \in S \\ \sum_{j \in S} P_j(S) + P_0(S) &= 1. \end{aligned}$$

This includes most choice models of practical interest (see Ben-Akiva and Lerman [48]) and even some rather pathological cases.[11] The following running example will be used to illustrate the model and analysis:

Example 2.5 An airline offers three fare products—*Y*, *M*, and *K*. These products differ in terms of revenues and conditions as shown in Table 2.7. The airline has

Table 2.7. **Fare-product revenues and restrictions for Example 2.5.**

Fare Product (Class)	*SA Stay*	*21-Day Adv.*	*Revenue*
Y	No	No	\$800
M	No	Yes	\$500
K	Yes	Yes	\$450

five segments of customers—two business segments and three leisure segments. The segments differ in terms of the restrictions that they qualify for and the fares they are willing to pay. The data describing each segment are given in Table 2.8. The second

Table 2.8. **Segments and their characteristics for Example 2.5.**

		Qualifies for Restrictions?		*Willing to Buy?*	
Segment	*Prob.*	*SA Stay*	*21-Day Adv.*	*Y Class*	*M Class*
Bus. 1	0.1	No	No	Yes	Yes
Bus. 2	0.2	No	Yes	Yes	Yes
Leis. 1	0.2	No	Yes	No	Yes
Leis. 2	0.2	Yes	Yes	No	Yes
Leis. 3	0.3	Yes	Yes	No	No

column of Table 2.8 gives the probability that an arriving customer is from each given segment.

Given this data for Example 2.5, the first four columns of Table 2.9 give the choice probabilities that would result.[12]

[11]For example, some psychologists have shown that customers can be overwhelmed by more choices, and they may become more reluctant to purchase as more options are offered (see Iyengar and Lepper (265)). Such cases would be covered by a suitable choice of $P_j(S)$ that results in the total probability of purchase, $\sum_{j\in S} P_j(S)$, being decreasing in *S*.

[12]To see how the probabilities in Table 2.9 are derived, consider the set $S = \{Y, K\}$. If $S = \{Y, K\}$ is offered, segments Business 1 and Business 2 buy the *Y* fare because they cannot qualify for both the SA stay and 21-day advance-purchase restrictions on *K*, so $P_Y = 0.1 + 0.2 = 0.3$. Similarly, Leisure 1 cannot qualify for the SA stay restriction of *K* and is not willing to purchase *Y*, so these customers do not purchase at all. Segments Leisure 2 and 3, however, qualify for both restrictions on *K* and purchase *K*. Hence, $P_K = 0.2 + 0.3 = 0.5$. Class *M* is not offered, so $P_M = 0$. The other rows of Table 2.9 are filled out similarly.

Table 2.9. Choice probabilities $P_j(S)$, probability of purchase $Q(S)$, and expected revenue $R(S)$ for Example 2.5.

S	$P_Y(S)$	$P_M(S)$	$P_K(S)$	$Q(S)$	$R(S)$	*Efficient?*[a]
$\{Y\}$	0.3	0	0	0.3	240	Yes
$\{M\}$	0	0.4	0	0.4	200	No
$\{K\}$	0	0	0.5	0.5	225	No
$\{Y, M\}$	0.1	0.6	0	0.7	380	No
$\{Y, K\}$	0.3	0	0.5	0.8	465	Yes
$\{M, K\}$	0	0.4	0.5	0.9	425	No
$\{Y, M, K\}$	0.1	0.4	0.5	1	505	Yes

[a] Efficient sets are defined in Section 2.6.2.3.

This particular method of generating choice probabilities is only for illustration. Other choice models could be used and in general any proper set of probabilities could be used to populate Table 2.9.

2.6.2.2 Formulation

As before, let C denote the total capacity, T the number of time-periods, t the current period, and x the number of remaining inventory units. Define the value function $V_t(x)$ as the maximum expected revenue obtainable from periods $t, t+1, \dots, T$ given that there are x inventory units remaining at time t. Then the Bellman equation for $V_t(x)$ is

$$\begin{aligned} V_t(x) &= \\ & \max_{S \subseteq \mathcal{N}} \left\{ \sum_{j \in S} \lambda P_j(S)(p_j + V_{t+1}(x-1)) + (\lambda P_0(S) + 1 - \lambda) V_{t+1}(x) \right\} \\ &= \max_{S \subseteq \mathcal{N}} \left\{ \sum_{j \in S} \lambda P_j(S)(p_j - \Delta V_{t+1}(x)) \right\} + V_{t+1}(x), \qquad (2.24) \end{aligned}$$

where $\Delta V_{t+1}(x) = V_{t+1}(x) - V_{t+1}(x-1)$ denotes the marginal cost of capacity in the next period, and we have used the fact that for all S,

$$\sum_{j \in S} P_j(S) + P_0(S) = 1.$$

The boundary conditions are

$$V_{T+1}(x) = 0, \quad x = 0, 1, \dots, C \qquad (2.25a)$$

$$V_t(0) = 0, \quad t = 1, \dots, T. \qquad (2.25b)$$

Note one key difference in this formulation compared to our analysis of the traditional independent-class models of Section 2.2.2 and Section 2.5—we assume the seller *precommits* to the open set of classes S

in each period, while in the traditional models, we assume the seller observes the class of the request and then makes an accept or deny decision based on the class. The reason for the difference is that in the traditional models the class of an arriving request is completely independent of the controls, so it doesn't matter whether we precommit to the set of open classes or not. However, in the choice-based model, the class that an arriving customer chooses depends (through the choice model $P_j(S)$) on which classes S we report as being open. Hence, the formulation (2.24) reflects this fact (we are taking max $E[\cdot]$ in 2.24 instead of E[max($\cdot$)]); we must choose S prior to seeing the realization of the choice decision.

2.6.2.3 Structure of the Optimal Policy

The problem (2.24) at first seems to have very little structure, but a sequence of simplifications provides a good characterization of the optimal policy. The first simplification is to write (2.24) in more compact form as

$$V_t(x) = \max_{S \subseteq \mathcal{N}} \{\lambda(R(S) - Q(S)\Delta V_{t+1}(x))\} + V_{t+1}(x), \tag{2.26}$$

where

$$Q(S) = \sum_{j \in S} P_j(S) = 1 - P_0(S)$$

denotes the total probability of purchase, and

$$R(S) = \sum_{j \in S} P_j(S) p_j$$

denotes the total expected revenue from offering set S. Table 2.9 gives the values $Q(S)$ and $R(S)$ for our Example 2.5. For theoretical purposes, we also consider allowing the seller to randomize over the sets S that are offered at the beginning of each time-period, but this relaxation is not strictly needed since there is always at least one set S that achieves the maximum in (2.26).

The second simplification is to note that not all $2^n - 1$ subsets need to be considered when maximizing the right-hand side of (2.26). Indeed, the search can be reduced to only those sets that are efficient as defined below:

DEFINITION 2.1 *A set T is said to be* ***inefficient*** *if there exist probabilities $\alpha(S), \forall S \subseteq \mathcal{N}$ with $\sum_{S \subseteq \mathcal{N}} \alpha(S) = 1$ such that*

$$Q(T) \geq \sum_{S \subseteq \mathcal{N}} \alpha(S) Q(S) \quad \text{and} \quad R(T) < \sum_{S \subseteq \mathcal{N}} \alpha(S) R(S),$$

A set is said to be ***efficient*** *if no such probabilities $\alpha(S)$ exist.*

In words, a set T is inefficient if we can use a randomization of other sets S to produce an expected revenue that is strictly greater than $R(T)$ with no increase in the probability of purchase $Q(T)$.

The significance of inefficient sets is that they can be eliminated from consideration:

PROPOSITION 2.3 *An inefficient set is never an optimal solution to (2.24).*

The proof is omitted, but the fact that such sets should be eliminated from consideration is quite intuitive from (2.26); an inefficient set T provides strictly less revenue $R(T)$ than do other sets and incurs at least as high a probability of consuming capacity $Q(T)$ (and hence incurs at least as high an opportunity cost $Q(S)\Delta V_{t+1}(x)$ in (2.26)).

For Example 2.5, Table 2.9 shows which sets are efficient—namely, the sets $\{Y\}$, $\{Y,K\}$, and $\{Y,K,M\}$. That these sets are efficient follows from inspection of Figure 2.7, which shows a scatter plot of the values $Q(S)$ and $R(S)$ for all subsets S. Note from this figure and Definition 2.1 that an efficient set is a point that is on the "efficient frontier" of the set of points $\{Q(S),\ R(S)\}, S \subseteq \mathcal{N}$. Here, "efficiency" is with respect to the tradeoff between expected revenue $R(S)$ and probability of sale $Q(S)$.

The third simplification is to note that the efficient sets can be easily ordered. Indeed, let m denote the number of efficient sets. These sets can be indexed $S_1, \ldots, S_m$ such that both the revenues and probabilities of purchase are monotone increasing in the index. That is, if the collection of m efficient sets is indexed such that $Q(S_1) \leq Q(S_2) \leq \cdots \leq Q(S_m)$, then $R(S_1) \leq R(S_2) \leq \cdots \leq R(S_m)$ as well. The proof of this fact is again omitted, but it is easy to see intuitively from Figure 2.7. Note from Table 2.9 that there are $m = 3$ efficient sets $\{Y\}$, $\{Y, K\}$, and $\{Y, K, M\}$. These can be ordered $S_1 = \{Y\}$, $S_2 = \{Y, K\}$, and $S_3 = \{Y, K, M\}$ with associated probabilities of purchase $Q_1 = 0.3$, $Q_2 = 0.8$, and $Q_3 = 1$ and prices $p_1 = \$240$, $p_2 = \$465$, and $p_3 = \$505$ as claimed.

Henceforth, we assume the efficient sets are denoted $S_1, \ldots, S_m$ and are indexed in increasing revenue and probability order. Also, to simplify notation we let $R_k = R(S_k)$ and $Q_k = Q(S_k)$ and note R_k and Q_k are both increasing in k. So the Bellman equation can be further simplified to

$$V_t(x) = \max_{k=1,\ldots,m} \{\lambda(R_k - Q_k \Delta V_{t+1}(x))\} + V_{t+1}(x). \tag{2.27}$$

The final simplification is to show that when expressed in terms of the sequence $S_1, \ldots, S_m$ of efficient sets, the optimal policy has a simple form as stated in the following theorem:

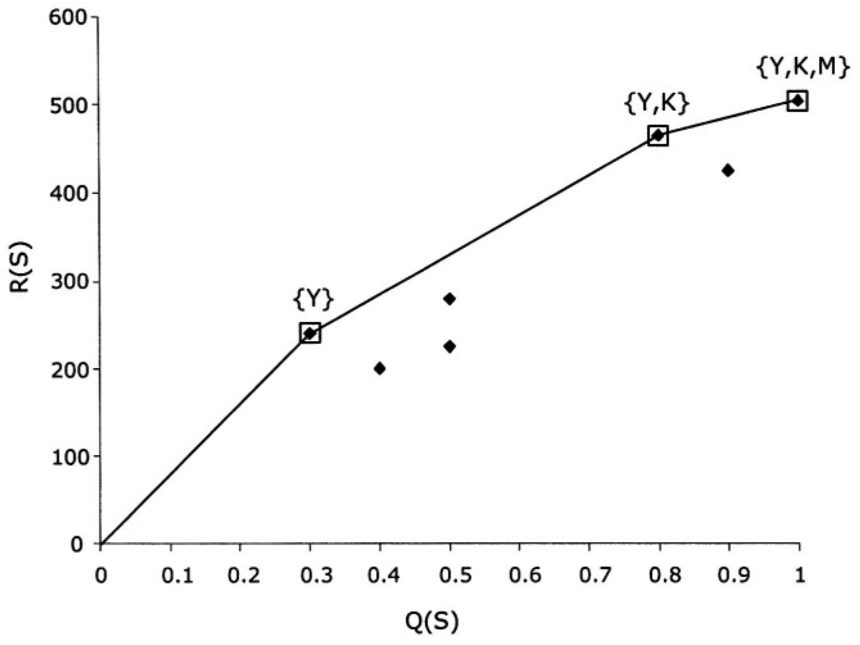


Figure 2.7. Scatter plot of $Q(S)$ and $R(S)$ for Example 2.5 (efficient points are enclosed in squares and labeled).

THEOREM 2.3 *An optimal policy for (2.24) is to select a set k^* from among the m efficient, ordered sets $\{S_k : k = 1, \ldots, m\}$ that maximizes (2.27). Moreover, for a fixed t, the largest optimal index k^* is increasing in the remaining capacity x, and for any fixed x, k^* is increasing in time t.*

The proof of this theorem is involved but derives from the fact that the marginal value $\Delta V_{t+1}(x)$ is decreasing in x (see Appendix 2.A for a proof) and the fact that the optimal index k^* is decreasing in this marginal value.

This characterization is significant for several reasons. First, it shows that the optimal sets can be reduced to only those that are efficient, which in many cases significantly reduces the number of sets we need to consider. Moreover, it shows that this limited number of sets can be sequenced in a natural way and that the more capacity we have (or the less time remaining), the higher the set we should use in this sequence.

For example, applying Theorem 2.3 to Example 2.5, we see that the efficient sets $S_1 = \{Y\}$, $S_2 = \{Y, K\}$, and $S_3 = \{Y, K, M\}$ would be used as follows. With very large amounts of capacity remaining, S_3 is optimal: all three fare classes are opened. As capacity is consumed, at some point we switch to only offering S_2: class *M* is closed, and only *Y* and *K* are offered. As capacity is reduced further, at some point we close class *K* and offer only class *Y* (set S_1 is used).

Note what's odd here; it can be optimal to offer the highest fare *Y* and the lowest fare *K*, but not the middle fare *M*. This is because opening *M* causes some buy-down from *Y* to *M*, whereas *K* is sufficiently restricted to prevent buy-down. Only when capacity is plentiful is *M* opened.

2.6.2.4 Identifying Efficient Sets

Finding the efficient sets is, in general, computationally complex. The naive approach is to enumerate all $2^n - 1$ subsets of $\mathcal{N}$ and for each set *T* solve a linear program (in variables $\alpha(S), S \subseteq \mathcal{N}$) to test for efficiency using the conditions in Definition 2.1.

However, a more efficient alternative is to use the following *largest marginal revenue* procedure. First, let $S_0 = \emptyset$. Then successive sets can be found by the following recursion. Let S_i be the i^{th} efficient set. Then the $(i+1)^{\text{st}}$ efficient set, S_{i+1}, is found by checking among the sets *S* with $Q(S) \geq Q(S_i)$ and $R(S) \geq R(S_i)$ for the one that maximizes the marginal revenue ratio

$$\frac{R(S) - R(S_i)}{Q(S) - Q(S_i)}.$$

(Note that this is simply the increase in expected revenue per unit increase in expected demand.) The procedure starts with $i = 0$ and stops as soon as no sets *S* with $Q(S) \geq Q(S_i)$ and $R(S) \geq R(S_i)$ exist; it returns the complete sequence $S_1, \ldots, S_m$. Since there are $O(2^n)$ subsets to check at each step, the recursion has complexity $O(m2^n)$, where m is the number of efficient sets (which in the worst case could be $O(2^n)$ itself).

For small numbers of classes, this largest marginal revenue procedure is practical, especially since it can be performed off line. But it is still exponential in the number of classes n. For large numbers of classes, heuristic or analytic approaches can be used to reduce the complexity of identifying efficient sets. For example, one could enumerate a limited collection of subsets *S* rather than all $2^n - 1$ subsets and apply the largest marginal revenue procedure to determine which subsets in the collection are efficient relative to other sets in the collection. In some special cases, as shown below, one can identify which subsets are efficient analytically, thus eliminating the need to enumerate all possible subsets.

2.6.2.5 Optimality of Nested-Allocation Policies

The optimization results above also have important implications for the optimality of nested-allocation policies. Indeed, Definition 2.1 and Theorem 2.3 can be used to provide a complete characterization of cases in which nested-allocation polices are optimal. They also can be used to provide conditions under which the optimal nesting is by revenue order.

We begin with a precise definition of a nested-allocation policy in the context of the choice model:

DEFINITION 2.2 *A control policy is called a* ***nested policy*** *if there is an increasing family of subsets $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_m$ and an index $k_t(x)$ that is increasing in x, such that set $S_{k_t(x)}$ is chosen at time t when the remaining capacity is x.*

Though this is a somewhat abstract definition of a nested policy, it is in fact a natural generalization of nested allocations from the traditional single-resource models of Section 2.2.2 and 2.5 and implies an ordering of the classes based on when they first appear in the increasing sequence of sets S_k. That is, class i is considered "higher" than class j in the nesting order if class i appears earlier in the sequence. Returning to Example 2.5, we see that the efficient sets are indeed nested according to this definition because $S_1 = \{Y\}$, $S_2 = \{Y, K\}$, and $S_3 = \{Y, K, M\}$ are increasing. Class *Y* would be considered the highest in the nested order, followed by class *K* and then class *M*.

If the optimal policy is nested in this sense, then we can define optimal protection levels $y_k^*(t), k = 1, \ldots, m$, such that classes lower in the nesting order than those in S_k are closed if the remaining capacity is less than $y_k^*(t)$, just as in the traditional single-resource case. The optimal protection levels for $k = 1, 2, \ldots, m-1$ are defined by

$$y_k^*(t) \quad = \quad \max\{x : R_k - Q_k \Delta V_{t+1}(x) > R_{k+1} - Q_{k+1} \Delta V_{t+1}(x)\}.$$

Nested booking limits can also be defined in the usual way, $b_k(t) = C - y_{k-1}(t)$.

We again return to Example 2.5 to illustrate this concept. Table 2.10 shows the objective function value $R_k - Q_k \Delta V_{t+1}(x)$ for each of the three efficient sets $k = 1, 2, 3$, for a particular marginal value function $\Delta V_{t+1}(x)$, which we assume is given in this example. Capacities are in the range $x = 1, 2, \ldots, 20$. The last column of Table 2.10 gives the index, $k_t^*(x)$, of the efficient set that is optimal for each capacity x.

Note that for capacities 1,2, and 3, the set $S_1 = \{Y\}$ is the optimal set, so class *Y* is the only open fare. Once we reach 4 units of remaining capacity, set $S_2 = \{Y, K\}$ becomes optimal and we open class *K* in addition to class *Y*. When the remaining capacity reaches 13, set $S_3 =$

$\{Y, K, M\}$ becomes optimal, and we open M in addition to Y and K. As a result, the optimal protection level for set S_1, is $y_1^* = 3$, and the protection level for set S_2 is $y_2^* = 12$. S_3 has a protection level equal to capacity.

Table 2.10. Illustration of nested policy for Example 2.5.

		$R_k - Q_k \Delta V_{t+1}(x)$			
x	$\Delta V_{t+1}(x)$	$k=1$	$k=2$	$k=3$	$k_t^*(x)$
1	780.00	6.00	-159.00	-275.00	1
2	624.00	52.80	-34.20	-119.00	1
3	520.00	84.00	49.00	-15.00	1
4	445.71	106.29	108.43	59.29	2
5	390.00	123.00	153.00	115.00	2
6	346.67	136.00	187.67	158.33	2
7	312.00	146.40	215.40	193.00	2
8	283.64	154.91	238.09	221.36	2
9	260.00	162.00	257.00	245.00	2
10	240.00	168.00	273.00	265.00	2
11	222.86	173.14	286.71	282.14	2
12	208.00	177.60	298.60	297.00	2
13	195.00	181.50	309.00	310.00	3
14	183.53	184.94	318.18	321.47	3
15	173.33	188.00	326.33	331.67	3
16	164.21	190.74	333.63	340.79	3
17	156.00	193.20	340.20	349.00	3
18	148.57	195.43	346.14	356.43	3
19	141.82	197.45	351.55	363.18	3
20	135.65	199.30	356.48	369.35	3

2.6.2.6 Nesting by Revenue Order

Revenues provide a natural nesting ordering, and, as described earlier in this chapter, this is traditionally how most quantity-based RM systems have been conceived and implemented. From a practical standpoint, therefore, it is important to understand when a particular choice model leads to nesting by revenue order. Yet Example 2.5 makes clear that nesting by revenue order need not be the optimal policy in general.

Talluri and van Ryzin [500] provide conditions that guarantee a given choice model will always have this property. The results show, for example, why the optimal control for the traditional independent-demand model is nested by revenue order. Talluri and van Ryzin [500] also show that if the choice probabilities follow the multinomial-logit (MNL) choice model (See Section 7.2.2.3.), then the optimal policy is always nested by revenue order. A similar result holds if customers' choice behavior is such that they always buy the lowest-price open class. However, in general nesting by revenue order need not be optimal.

2.6.2.7 Comparisons of Optimality Conditions

The optimality conditions in the nested-by-revenue-order case also provide some intuition into how choice-based controls differ from traditional controls. Let $C_k = \{1, \ldots, k\}$ denote the set of the k highest classes (in revenue order). Then one can show it is optimal to open class $k+1$ if and only if

$$P_{k+1}(C_{k+1})\,(r_{k+1} - \Delta V_{t-1}(x)) \geq \sum_{j=1}^{k} \Delta P_j(C_k)(r_j - \Delta V_{t-1}(x)), \quad (2.28)$$

where

$$\Delta P_j(C_k) = P_j(C_k) - P_j(C_{k+1})$$

is the change (usually an increase for most choice models) in purchase probability for class j as the result of *not* offering class $k+1$. This expression is intuitive: The left-hand side is the probability of selling class $k+1$ times the "net gain" from selling it; that is, the revenue we get from class $k+1$ minus the opportunity cost, $\Delta V_{t-1}(x)$, of using a unit of capacity. The right-hand side is the net gain (loss) among the other classes caused by eliminating (adding) class $k+1$ (the sum over all the other class j in C_k of the change in purchase probability times the net gain from selling j). Therefore, the condition (2.28) simply says that if the expected gain on class $k+1$ exceeds the incremental loss on the other classes caused by adding $k+1$, then it pays to open $k+1$; otherwise, $k+1$ should be closed.

The expression (2.28) should be compared with the optimality condition for independent-demand model; namely, it is optimal to open class $k+1$ if and only if

$$r_{k+1} - \Delta V_{t-1}(x) \geq 0.$$

(Indeed, note that (2.28) reduces to the above expression for the independent-demand model since for this model $\Delta P_j(C_k) = 0$ for all j, k.) Note that the right-hand side above is zero while the right-hand side of (2.28) is nonzero (typically positive). This happens because in the independent-demand model, if we close class $k+1$, we lose all demand for that class. Therefore, it is optimal to accept class $k+1$ whenever r_{k+1} exceeds the opportunity cost $\Delta V_{t-1}(x)$. However, in the choice-based model, if we close class $k+1$, customers choose from among the other classes that are offered (e.g., from C_k). Hence, the threshold on the right-hand side of (2.28) is nonzero. This difference reflects the fact that customers may buy up to a higher class, so there is some nonzero benefit to rejecting a request for $k+1$.

2.6.2.8 A Numerical Comparison to EMSR-b with Buy-Up

We next look at the results of a small simulation study comparing choice-based methods to a traditional EMSR-b method with buy-up as described in Section 2.6.1. While the EMSR-b model is developed under the static-model assumptions, it is frequently used as a heuristic in the dynamic case by simply aggregating the total demand to come for each class. Also, it is one of the few models available that incorporates some type of choice behavior. For these reasons, it serves as a useful benchmark for comparison.

This simulations are based on our running three-class example, Example 2.5. We compare the optimal control given by the choice dynamic program with the traditional EMSR-b (buy-up) recommendations. (Recall that for Example 2.5 the optimal dynamic programming policy uses the fare classes in the order *Y, K, M,* whereas the EMSR-b uses them in the fare order *Y, M, K.*)

The capacity C = 20, and there are three population sizes: 15,20, and 25. The fares, restrictions, and customer segments are as given in Table 2.8. For a population size of 20 this results in an unconstrained mean and variance as shown in Table 2.11. These statistics are used to create inputs for EMSR-b. The buy-up factors are computed as shown

Table 2.11. **The different segment choices in Example 2.5 if all classes are open and resulting demand for a population size of 20 (N.E. stands for not eligible).**

Segments	*%*	*Sat-Night*	*21-Day*	*Can Buy Y?*	*Can Buy M?*	*Eligible for K?*	*Choice*	*Mean*	*Var.*
B1	0.1	No	No	Yes	N.E	N.E.	Y	2	1.8
B2	0.2	No	Yes	Yes	Yes	N.E.	M	4	3.2
L1	0.2	No	Yes	No	Yes	N.E.	M	4	3.2
L2	0.2	Yes	Yes	No	Yes	Yes	K	4	3.2
L3	0.3	Yes	Yes	No	No	Yes	K	6	4.2

in Table 2.12, which lists the unconstrained choices and demands when all classes are open. (N.E. signifies that the segment is not eligible.) This estimate roughly mimics the traditional practice of unconstraining and forecasting demand in each class. Table 2.12 then sums the unconstrained demands for each fare class. The buy-up factors are estimated as follows. The buy-up factor for K is given by the percentage K customers who would buy up to M if we go from offering $\{Y, M, K\}$ to $\{Y, M\}$. Similarly the buy-up factor for M is the fraction who buy up

Table 2.12. Inputs to the EMSR-b model.

Class	*Mean*	*S.D.*	*Buy-Up Factors*
Y	2	1.34	
M	8	2.52	0.33
K	10	2.72	0.40

to Y when going from offering $\{Y, M\}$ to Y. Table 2.12 shows these demands and buy-up factors for a population size of 20, and Table 2.13 shows the computed EMSR-b protection levels for this population size. The results for the three load factors are summarized in Table 2.14.

Table 2.13. Protections for the EMSR-b model without and with buy-up factors.

Class	*EMSR-b*	*EMSR-b with Buy-Up*
Y	1.57	2.20
Y + M	7.55	8.71

Note that the choice dynamic program shows significant improvements

Table 2.14. Simulation results comparison between choice dynamic program and EMSR-b with buy-up.

Population Size	*EMSR-b Revenue*	*Choice DP Revenue*	*% gain from Choice DP*	*99% Conf. Int. Error on % Gain*
15	7,466	7,466		
20	9,825	10,129	3.10	± 0.0050
25	10,142	11,301	11.48	± 0.0079

on this example, achieving an 11.5% improvement in revenue in the high-demand case. Again, part of this improvement can be attributed to the fact that the choice dynamic program uses a different sequence of classes (only the efficient sets $\{Y\}$, $\{Y, K\}$, $\{Y, M, K\}$).

2.7 Notes and Sources

The notion of theft versus standard nesting is not well documented and is part of the folklore of RM practice. Our understanding, however, greatly benefited from discussions with our colleagues Peter Belobaba, Sanne de Boer, and Craig Hopperstad.

The earliest paper on the static models of Section 2.2 is Littlewood [347]. Another early applied paper is Bhatia and Parekh [64]. But there are close connections to earlier work on the stock-rationing problem in the inventory literature by Kaplan [288] and Topkis [515];

see also Gerchak [209, 210] and Ha [246]. Indeed, Topkis's [515] results can be used to show the optimality of nested-allocation policies.

Optimal policies for the $n > 2$ case were obtained in close succession (using slightly different methods and assumptions) in papers by Brumelle and McGill [91], Curry [139], Robinson [445], and Wollmer [576]. See also McGill's thesis [375]. Robinson [445] also analyzed the case where the order of arrival is not the same as the revenue order. Brumelle et al. [90] analyzed a two-class static model with dependent demand.

The Monte Carlo method for computing optimal overbooking limits presented in Section 2.2.3.2 is due to Robinson [445].

The dynamic model of Section 2.5 was first analyzed by Lee and Hersh [336]. However, the proofs of Propositions 2.1 and 2.2 in Appendix 2.A are due to Lautenbacher and Stidham [330], who provided a unified analysis of both the static and dynamic single-resource models. Walczak and Brumelle [543] relate this problem to a dynamic pricing problem using a Markov model of demand that allows for partial information on the revenue values or customer types. See Liang [342] for an analysis of a continuous-time version of the dynamic model.

The EMSR-a and EMSR-b heuristics are both due to Belobaba. The most detailed coverage of EMSR-a is contained in Belobaba's 1987 thesis [39], but see also the published articles from it [38] and [40]. EMSR-b was introduced in [41]; see also Belobaba and Weatherford [37].

The problem of group or batch request in Section 2.4 was addressed by Lee and Hersh [336] for the partially accepted case. For the more complex case where groups must be completely accepted, see Walczak and Brumelle [544], Kleywegt and Papastavrou [307], and Van Slyke and Young [530].

The adaptive algorithm in Section 2 3 is due to van Ryzin and McGill [526]. The buy-up heuristics in Section 2.6.1 are due to Belobaba [39, 38, 40]. See also Belobaba and Weatherford [37], Weatherford [555], and the simulation study of Bohutinsky [81]. See Titze [514] for a discussion of passenger behavior in the simple two-class model. The material in Section 2.6.2 on choice-based models is from Talluri and van Ryzin [500]; see also Algers and Besser [7] and Andersson [18] for an application of discrete-choice models at SAS. For a good reference on discrete-choice modeling, see Ben-Akiva and Lerman [48]. De Boer [155] is another recent work that addresses customer choice in a single-resource problem.

APPENDIX 2.A: Monotonicity Proofs

The proofs of monotonicity are based on a lemma of Stidham [487] originally developed to analyze queueing-control problems. The lemma was adapted to provide a convexity proof for a general single-resource problem, which includes both the static and dynamic models, in work by Lautenbacher and Stidham [330]. The proof here follows theirs.

We begin with a definition:

DEFINITION 2-2.A.3 *A function defined on the set of nonnegative integers, $g : \mathcal{Z}_+ \to \Re$ is concave if it has nonincreasing differences. That is, $g(x+1) - g(x)$ is nonincreasing in $x \geq 0$.*

LEMMA 2-2.A.1 *Suppose $g : \mathcal{Z}_+ \to \Re$ is concave. Let $f : \mathcal{Z}_+ \to \Re$ be defined by*

$$f(x) = \max_{a=0,1,\ldots,m} \{ap + g(x-a)\}$$

for any given $p \geq 0$ and nonnegative integer $m \leq x$. Then $f(x)$ is concave in $x \geq 0$ as well.

Proof

First, note that by changing variables to $y = x - a$ we can write $f(x) = \hat{f}(x) + rx$, where

$$\hat{f}(x) = \max_{x-m \leq y \leq x} \{-yp + g(y)\}.$$

We first analyze $\hat{f}(x)$. Let $y^* = \arg\max_{y \geq 0}\{-yp + g(y)\}$. Since $g(y)$ is concave, $-yp + g(y)$ is also concave, and moreover is nondecreasing for values of $y \leq y^*$ and nonincreasing for values of $y > y^*$. Therefore, for a given m and p,

$$\hat{f}(x) = \begin{cases} -xp + g(x) & x \leq y^* \\ y^*p + g(y^*) & y^* \leq x \leq y^* + m \\ -(x-m)p + g(x-m) & x \geq y^* + m. \end{cases}$$

Therefore, in the range $x < y^*$ and using the fact that $g(x)$ is concave

$$\begin{aligned} \hat{f}(x+1) - \hat{f}(x) &= -p + g(x+1) - g(x) \\ &\leq -p + g(x) - g(x-1) \\ &= \hat{f}(x) - \hat{f}(x-1). \end{aligned}$$

For $y^* \leq x < y^* + m$, it follows that $\hat{f}(x+1) - \hat{f}(x) = 0$, so $\hat{f}(x)$ is trivially concave in this range.

Finally, for $x \geq y^* + m$, from the concavity of $g(x)$

$$\begin{aligned} \hat{f}(x+1) - \hat{f}(x) &= -p + g(x+1-m) - g(x-m) \\ &\leq -p + g(x-m) - g(x-1-m) \\ &= \hat{f}(x) - \hat{f}(x-1). \end{aligned}$$

Thus, $\hat{f}(x)$ is concave in $x \geq 0$ and since $f(x) = \hat{f}(x) + rx$, $f(x)$ is concave in $x \geq 0$ as well. *QED*

Proof of Proposition 2.1

We first prove part (i) of Proposition 2.1—namely, that the marginal value $\Delta V_j(x)$ is nonincreasing in x (that $V_j(x)$ is concave in x). The proof is by induction on the stages. Note that in the terminal stage (stage 0), $V_{n+1}(x) = 0$ for all x, which is trivially concave. Now assume that $V_{j-1}(x)$ is concave in x and consider $V_j(x)$. Note that

$$V_j(x) = E\left[\max_{0 \le u \le \min\{D_j, x\}} \{p_j u + V_{j-1}(x-u)\}\right].$$

The inner maximization is precisely of the form given by Lemma 2-2.A.1 with $m = \min\{D_j, x\}$. Hence, it follows that for any realization of D_j, the function

$$H(D_j, x) = \max_{0 \le u \le \min\{D_j, x\}} \{p_j u + V_{j-1}(x-u)\}$$

is concave in x. Since $V_j(x) = E_{D_j}[H(D_j, x)]$, it is a weighted average (nonnegative weights) of concave functions, and hence it follows that $V_j(x)$ is concave as well.

Part (ii) of the Proposition 2.1 says that the marginal value at a given capacity x at stage j is less than at stage $j+1$. This is shown as follows:

$$\begin{aligned}
\Delta V_j(x) = & \\
& E\left[p_j \min\{D_j, (x - y^*_{j-1})^+\} + V_{j-1}(x - \min\{D_j, (x - y^*_{j-1})^+\})\right] \\
& -E\left[p_j \min\{D_j, (x - 1 - y^*_{j-2})^+\} + V_{j-1}(x - 1 - \min\{D_j, (x - y^*_{j-2})^+\})\right] \\
\ge \; & E\left[p_j \min\{D_j, (x - y^*_{j-2})^+\} + V_{j-1}(x - \min\{D_j, (x - y^*_{j-2})^+\})\right] \\
& -E\left[p_j \min\{D_j, (x - 1 - y^*_{j-2})^+\} + V_{j-1}(x - 1 - \min\{D_j, (x - y^*_{j-2})^+\})\right] \\
\ge \; & E\left[\Delta V_{j-1}(x - \min\{D_j, (x - y^*_{j-1})^+\})\right] \\
\ge \; & \Delta V_{j-1}(x),
\end{aligned}$$

where the first inequality follows because y^*_{j-1} is the optimal protection level at stage $j-1$, the second inequality follows from the nonnegativity of $\min\{D_j, (x - y^*_{j-2})^+\} - \min\{D_j, (x - 1 - y^*_{j-2})^+\}$, and the last inequality follows from the fact that $\Delta V_{j-1}(x)$ is decreasing in x. *QED*

Proof of Proposition 2.2 Similarly, we can use Lemma 2-2.A.1 to show that the increments $\Delta V_t(x)$ of the value function $V_t(x)$ defined by (2.17) are nonincreasing as well: $V_t(x)$ is concave in x. The proof is by induction on t. First, note $V_{T+1}(x) = 0$ for all x, so $\Delta V_{T+1}(x)$ is trivially decreasing in x. Next, assume $V_{t+1}(x)$ is concave and consider period t. The Bellman equation (2.17) is

$$V_t(x) = E\left[\max_{u \in \{0,1\}} \{R(t)u + V_{t+1}(x-u)\}\right].$$

The inner maximization is again in the form of Lemma 2-2.A.1 with $m = 1$. Hence, the function

$$G(R(t), x) = \max_{u \in \{0,1\}} \{R(t)u + V_{t+1}(x-u)\}$$

is concave in x for any realization of $R(t)$. Since $V_t(x) = E_{R(t)}[G(R(t), x)]$, it follows that $V_t(x)$ is concave in x as well.

To show monotonicity in t, note that

$$\begin{aligned}
\Delta V_t(x) &= E\left[(R(t) - \Delta V_{t+1}(x))^+ + V_{t+1}(x)\right] \\
&\quad -E\left[(R(t) - \Delta V_{t+1}(x-1))^+ + V_{t+1}(x-1)\right] \\
&= \Delta V_{t+1}(x) + E\left[(R(t) - \Delta V_{t+1}(x))^+ - (R(t) - \Delta V_{t+1}(x-1))^+\right] \\
&\ge \Delta V_{t+1}(x),
\end{aligned}$$

where the last inequality follows from the fact that $\Delta V_{t+1}(x) \leq \Delta V_{t+1}(x-1)$ and hence for any realization $R(t)$,

$$(R(t) - \Delta V_{t+1}(x))^{+} \geq (R(t) - \Delta V_{t+1}(x-1))^{+}.$$

QED

Proof of Monotonicity of Marginal Values from the Choice-Based Model

We next show for completeness that the marginal values in the choice-based models are also decreasing in remaining capacity. Namely,

PROPOSITION 2-2.A.4 *For the choice-based single-resource problem defined by (2.24), the value function satisfies*

$$\Delta V_t(x) \leq \Delta V_t(x-1), \quad \forall t, x$$

and

$$\Delta V_t(x) \geq \Delta V_{t-1}(x), \quad \forall t, x$$

Proof

The proof is by induction on t. First, the statement is trivially true for $t = T+1$ by the boundary conditions (2.25a). Assume it is true for period $t+1$ and consider period t. Let $S_t^*(x)$ denote the optimal solution to (2.24) and note

$$\begin{aligned}
\Delta V_t(x) - \Delta V_t(x-1) &= (\Delta V_{t+1}(x) - \Delta V_{t+1}(x-1)) \\
&\quad + \sum_{j \in S_t^*(x)} \lambda P_j(S_t^*(x))(p_j - \Delta V_{t+1}(x)) \\
&\quad - \sum_{j \in S_t^*(x-1)} \lambda P_j(S_t^*(x-1))(p_j - \Delta V_{t+1}(x-1)) \\
&\quad - \sum_{j \in S_t^*(x-1)} \lambda P_j(S_t^*(x-1))(p_j - \Delta V_{t+1}(x-1)) \\
&\quad + \sum_{j \in S_t^*(x-2)} \lambda P_j(S_t^*(x-2))(p_j - \Delta V_{t+1}(x-2))
\end{aligned} \tag{2.A.1}$$

From the optimality of the set defined by $S_t^*(\cdot)$, the following inequalities hold:

$$\sum_{j \in S_t^*(x-1)} \lambda P_j(S_t^*(x-1))(p_j - \Delta V_{t+1}(x-1)) \geq \sum_{j \in S_t^*(x)} \lambda P_j(S_t^*(x))(p_j - \Delta V_{t+1}(x-1))$$

and

$$\sum_{j \in S_t^*(x-1)} \lambda P_j(S_t^*(x-1))(p_j - \Delta V_{t+1}(x-1)) \geq \sum_{j \in S_t^*(x-2)} \lambda P_j(S_t^*(x-2))(p_j - \Delta V_{t+1}(x-1))$$

Substituting into (2.A.1) we obtain

$$\begin{aligned}\Delta V_t(x) - \Delta V_t(x-1) \leq\ & \Delta V_{t+1}(x) - \Delta V_{t+1}(x-1) \\ & + \sum_{j\in S_t^*(x)} \lambda P_j(S_t^*(x))(p_j - \Delta V_{t+1}(x)) \\ & - \sum_{j\in S_t^*(x)} \lambda P_j(S_t^*(x))(p_j - \Delta V_{t+1}(x-1)) \\ & - \sum_{j\in S_t^*(x-2)} \lambda P_j(S_t^*(x-2))(p_j - \Delta V_{t+1}(x-1)) \\ & + \sum_{j\in S_t^*(x-2)} \lambda P_j(S_t^*(x-2))(p_j - \Delta V_{t+1}(x-2))\end{aligned}$$

Rearranging and canceling terms yields

$$\begin{aligned}\Delta V_t(x) - \Delta V_t(x-1) \leq\ & (1 - \sum_{j\in S_t^*(x)} \lambda P_j(S_t^*(x)))(\Delta V_{t+1}(x) - \Delta V_{t+1}(x-1)) \\ & + \sum_{j\in S_t^*(x-2)} \lambda P_j(S_t^*(x-2))(\Delta V_{t+1}(x-1) \\ & - \Delta V_{t+1}(x-2))\end{aligned}$$

By induction, $\Delta V_{t+1}(x) - \Delta V_{t+1}(x-1) \leq 0$ and $\Delta V_{t+1}(x-1) - \Delta V_{t+1}(x-2) \leq 0$. Therefore, $\Delta V_t(x) - \Delta V_t(x-1) \leq 0$.

To show the marginal values are monotone increasing in the remaining time, note that

$$\begin{aligned}\Delta V_t(x) &= V_t(x) - V_t(x-1) \\ &= \max_k \{\lambda(R_k - Q_k \Delta V_{t-1}(x))\} - \max_k \{\lambda(R_k - Q_k \Delta V_{t-1}(x-1))\} + \Delta V_{t-1}(x).\end{aligned}$$

From the monotonicity in x we have that $\Delta V_{t-1}(x) \leq \Delta V_{t-1}(x-1)$, and therefore for any value k,

$$\lambda(R_k - Q_k \Delta V_{t-1}(x)) - \lambda(R_k - Q_k \Delta V_{t-1}(x-1)) \geq 0.$$

Hence

$$\max_k \{\lambda(R_k - Q_k \Delta V_{t-1}(x))\} - \max_k \{\lambda(R_k - Q_k \Delta V_{t-1}(x-1))\} \geq 0$$

as well, and it follows that $\Delta V_t(x) \geq \Delta V_{t-1}(x)$. *QED*