Introduction to Markov Decision Processes

Martin L. Puterman and Timothy C. Y. Chan

October 1, 2024

Chapter 1

Examples and Applications

This material will be published by Cambridge University Press as Introduction to Markov Decision Processes by Martin L. Puterman and Timothy C. Y. Chan. This pre-publication version is free to view and download for personal use only. Not for re-distribution, re-sale, or use in derivative works. ©Martin L. Puterman and Timothy C. Y. Chan, 2024.

We welcome all feedback and suggestions at:
martin.puterman@sauder.ubc.ca and tcychan@mie.utoronto.ca

For the things we have to learn before we can do them, we learn by doing them.
Aristotle, Greek philosopher, 384-322 BC

Representing a sequential decision problem as a Markov decision process requires specifying five objects: decision epochs, states, actions, transition probabilities and rewards. This chapter shows how to do so by identifying these objects in applications spanning a broad range of disciplines. The objective of this chapter is to highlight both the wide applicability of the MDP framework and the diverse aspects of model formulation. Note that several of these examples will be revisited in subsequent chapters.

We encourage readers to identify decision epochs, states, actions, transition probabilities and rewards on their own before looking at our approach. The chapter concludes with guidance on how to formulate Markov decision process models in general. The problems at the end of the chapter provide additional opportunities to formulate Markov decision processes or revise the provided examples when assumptions differ.

1.1 Inventory management

"When you have an inventory-based business, most people think only about the first order," Mr. Green said. With long lead times from the factory in China, he was almost immediately trying to figure out how big his next order should be. Underestimating

would hurt not just his sales but the status of his Amazon listing; overestimating would drain cash upfront, and he would incur further charges from Amazon for storing excess inventory in its warehouses.

The Great Amazon Flip-a-Thon; John Herrman, New York Times, S1,S7, April 4, 2021

Inventory models represent some of the earliest and most widely studied applications of Markov decision processes. They concern determining appropriate inventory levels for a product in the face of uncertain customer demand. As the quote above indicates, having too little inventory results in losing sales and reputation, while having too much inventory leads to excess storage and capital charges. These costs are key components of inventory models; balancing this trade-off is the primary challenge.

The models considered here assume that an *inventory manager* periodically (hourly, daily, weekly or monthly) observes the inventory level of a product and, if deemed opportunistic, places a replenishment order with a *supplier*. This replenishment may arrive immediately, at the end of the current period (prior to the next review period) or several periods in the future. The delay between placing and receiving an order is referred to as a *lead time*.

Features of inventory management applications that impact an Markov decision process formulation include:

1. **Ordering costs** consist of a fixed component and a variable component. The fixed component K represents the administrative cost of placing an order[1] and the variable component $c(u)$ represents the cost of ordering u units. Thus when an order is placed, the cost is $K + c(u)$ and when no order is placed, the cost is 0.

2. **Holding costs**, denoted by $h(u)$, represent the cost to the inventory manager of storing u units of product for one period. This cost only applies when $u > 0$. It is convenient to assume that $h(0) = 0$.

3. **Lost sales** occur if there is insufficient inventory to fulfill demand in the current period.

4. **Backlogged demand** means that unmet demand may be fulfilled at some period in the future when inventory is available. *A different formulation applies depending on whether unmet demand is met or backlogged.*

5. **Penalty costs**, denoted by $p(u)$, represent the cost to the inventory manager of backlogging u units of demand for one period. This cost applies only when $u < 0$. It is convenient to assume $p(0) = 0$.

[1] In some applications, $K = 0$, but usually $K > 0$. It is the presence of this cost that makes it advantageous for the inventory manager to not order every period.

6. **Total customer demand** in each period is random. It may arrive in one batch at a specific point of time within a period or may arrive at random times within a period. The timing of demand will impact the formulation. The demand distribution may be known or unknown, and may be static or time-varying. Let the non-negative[2] random variable Z_n denote the total demand in period n. We will represent its distribution by $P(Z_n = z)$.

7. **Revenue** corresponds to them amount spent by customers purchasing units of a product. is represented by an increasing function $R(\cdot)$; its relationship to demand depends on some of the above features. If unfulfilled demand is backlogged, it is simplest to assume that payments are made "up front", that is when an order is placed[3] so that if the demand is z, the revenue is $R(z)$. In the lost sales case, sales are capped by inventory on hand so that the revenue is $R(z)$ if the demand is less than inventory level s and $R(s)$ if the demand exceeds the inventory s. In this latter case, low inventory levels may result in lost sales and reduced revenue.

8. **Product shelf-life** may be finite or "infinite[4]". Products with finite shelf lives are said to be *perishable*. Perishable items may last for one-period (newspapers, fresh bread, defrosted vaccine or a seat at a sporting event) or multiple-periods (blood products, consumer electronics or fashion goods). Non-perishable goods include tools, books and basic clothing. Our discrete-state formulation implicitly assumes product is only available in whole units.

9. **Scrap value**, denoted by $H(s)$, equals the value of the ending inventory when there are s units on hand at the end of a finite planning horizon. If $s \geq 0$, $H(s)$ may include any potential holding cost before the inventory is liquidated. If $s < 0$, then $H(s)$ represents a penalty associated with not being able to fulfill the $|s|$ items backlogged (e.g., the loss associated with buying these items at a premium from a back-up supplier, or a loss in goodwill due to lost sales).

1.1.1 Inventory management with backlogged demand

Formulating sequential inventory management as a Markov decision process requires precisely specifying the timing of events and assumptions. Figure 1.1 depicts the event sequence for the following model. At the start of period N, the decision maker observes the inventory level s and decides on how many units a of product to order from the supplier to replenish inventory. Demand arrives randomly throughout the period and is fulfilled at the end of the period, using inventory on hand together with units that arrive from the current period's order. If demand exceeds inventory, it is

[2] Most inventory models assume non-negative demand.

[3] Alternatively payments may be made when the order is fulfilled.

[4] No product lasts forever, this is a modelling convention to facilitate representing future inventory levels as functions of the present inventory, the quantity ordered and the demand only

backlogged for fulfillment from future inventory replenishments. An Markov decision process formulation follows.

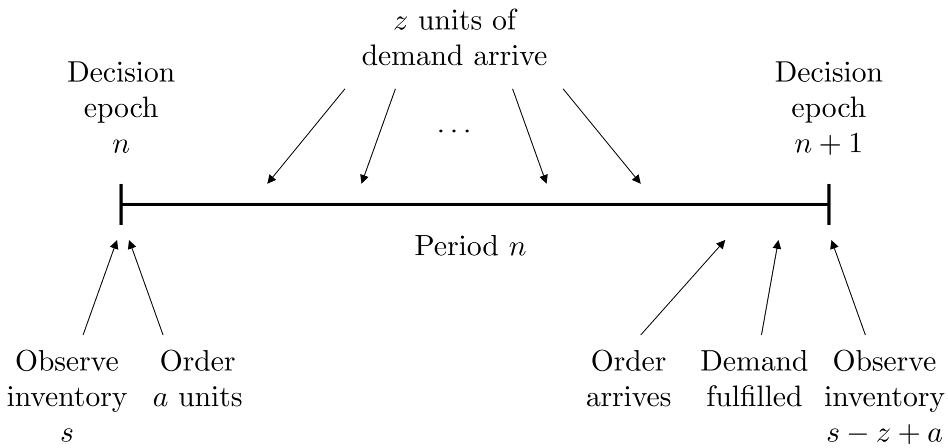

Figure 1.1: Timing of events in the periodic review inventory model.

Decision Epochs: Decision epochs correspond to the times at which the decision maker reviews the inventory. This problem can be modeled either with a finite or infinite planning horizon. Hence

$$T = \{1, 2, \ldots, N\}, \quad N \leq \infty.$$

States: States represent the number of units on hand at a decision epoch. A negative value indicates backlogged demand.

$$S = \{\ldots, -2, -1, 0, 1, 2, \ldots\}.$$

Note that the quantities backlogged and in inventory may be truncated at large values (e.g., the capacity of a warehouse) to ensure a finite state space. Doing so makes the formulation slightly more complex because of the resulting boundary conditions.

Actions: Actions represent the quantity ordered from the supplier for delivery prior to the next decision epoch. For each $s \in S$,

$$A_s = \{0, 1, 2, \ldots\}.$$

Similar to the state space, the action space can be truncated so that it is finite.

Rewards: The reward in a period equals revenue minus cost. As noted above, revenue depends on whether unmet demand is backlogged or lost. Since this formulation assumes backlogging, when the total demand in a period is z, the revenue equals $R(z)$. Note that the inventory level at the next decision epoch, j is given by $j = s + a - z$ so that knowing only the inventory levels s and j and the order quantity a one can infer that the demand is given $z = s + a - j$.

There are three cost components; ordering costs, which are incurred only if $a > 0$, holding costs if s is positive and penalty costs if s is negative. A simplifying assumption is that holding and penalty costs are assessed at the beginning of the period based on the *starting* inventory.

The reward can be written[5]

$$r_n(s,a,j) = R(s+a-j) - KI_{\mathbb{Z}_{>0}}(a) - c(a) - h(s)I_{\mathbb{Z}_{>0}}(s) - p(-s)I_{\mathbb{Z}_{>0}}(s)$$

when $n < N$. The argument of $p(\cdot)$ is $-s$, which equals the backlogged demand when s is negative.

If N is finite, $r_N(s) = H(s)$, corresponding to the scrap value of the remaining inventory at the end of the planning horizon.

Transition Probabilities: As noted above, when the state at decision epoch $n+1$ is j, this means that demand in period n was $s + a - j$. Moreover, the assumption of non-negative demand ensures that the state at the next decision epoch cannot exceed $s + a$. Hence, the transition probabilities are

$$p_n(j|s,a) = \begin{cases} P(Z_n = s+a-j), & j \le s+a,\ j \text{ integer} \\ 0, & j = s+a+1, s+a+2, \dots \end{cases}$$

Application challenges

Applying this model presents several challenges. In particular, one must determine demand distributions, ordering costs, holding costs and penalty costs. Demand distributions may be estimated from historical data; parameterizing the model in terms of a known distribution reduces the challenge in estimating model parameters. Moreover it is likely that demand has seasonal components at the day, week and month levels.

Per unit ordering costs should be easily obtainable but fixed ordering costs may be more challenging to determine. The fixed component encompasses administrative, shipping and handling costs that may be hard to untangle from the per unit cost. Holding costs are real and involve cost of capital and space charges.

Penalty costs are the most challenging to determine since they involve intangibles such as loss of goodwill due to unfulfilled or delayed orders. Instead, the decision maker

[5] In the definition of the reward $\mathbb{Z}_{>0}$ denotes the positive integers so that $I_{\mathbb{Z}_{>0}}(x)$ equals 1 when X is positive and 0 otherwise.

may specify a service level such as 95% of orders be processed from stock on hand and use a constrained Markov decision process model formulation.

Note that major disruptions to supply chain or consumer behavior arising from, for example, a global pandemic, can lead to significant challenges in estimating appropriate parameters. Procurement costs might be much higher due to increased demand for raw materials and lower manufacturing capacity. Demand for certain products could be significantly increased or decreased compared to historical levels.

1.1.2 The newsvendor problem

This section describes a simple yet widely studied and applied inventory model referred to as the *newsvendor problem*[6]. This model is fundamental in the operations research literature and applies to perishable goods with a shelf-life of one period.

As shown in Figure 1.2, events unfold as follows. At the start of the period, the newsvendor purchases a units of product (newspapers) from a supplier at a cost of c per unit. After receiving the units, a random demand of z units arrives. Items are sold throughout the period at a price of g with $g > c$. Each unit sold results in a profit (selling price minus cost) $G = g - c$. At the end of the period each unsold item is disposed of at a scrap value of h where $c > h$ resulting in a loss (cost minus salvage value) $L = c - h$ resulting in an ending of inventory zero.

The newsvendor seeks an order quantity a^* that maximizes the one-period expected reward. The choice of order quantity *trades off* ordering too many items and incurring a loss on unsold items with ordering too few items and foregoing potential additional profit. Another concern is that if a is set too low, the newsvendor will observe only *censored* demand complicating estimation of the demand distribution when it is unknown. Thus, initially the newsvendor may set a higher, trading off potential losses for gains in information about the demand distribution.

Although this model is usually analyzed from first principles, it can be formulated as a single-state single-period Markov decision process as follows:

Decision Epochs: There is only one decision epoch so that $T = \{1\}$. Note that this means that $N = 2$.

States: Since every period starts and ends with zero inventory, $S = \{0\}$.

Actions: Actions represent the quantity ordered from the supplier for immediate delivery so that

$$A_0 = \{0, 1, 2, \dots a_{\max}\}$$

where the set of actions is bounded by some "reasonable" value $a_{\max}$.

[6]This was first called a "newsboy" problem. It modelled the problem of a newsboy who purchased papers from a dealer, sold them at a street corner and returned the unsold papers to the dealer at the end of the day.

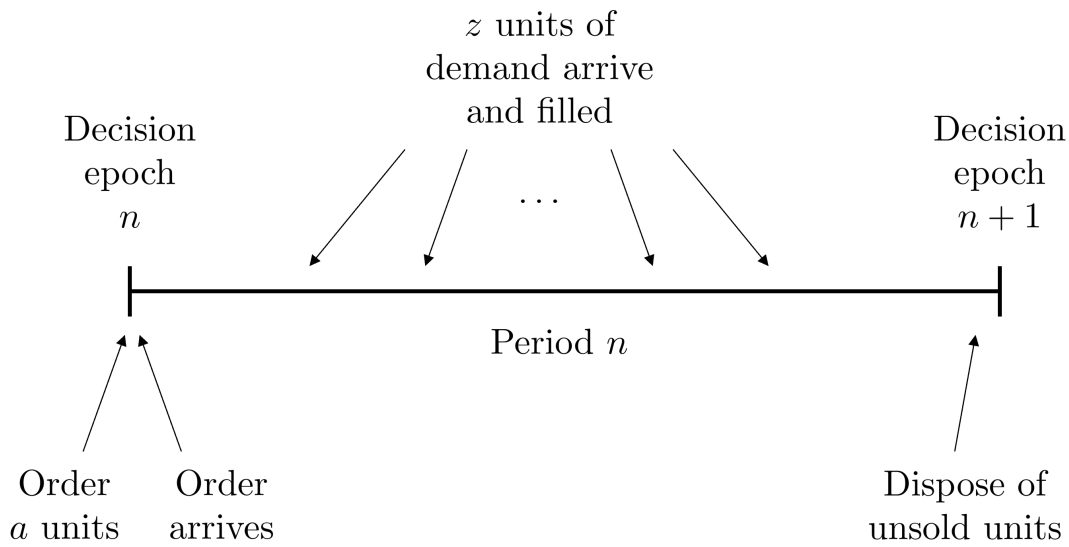

Figure 1.2: Timing of events in a newsvendor model. Demand is filled when it arrives.

Rewards: It is easiest to define the expected reward directly since the demand does not appear explicitly in the Markov decision process formulation. Consequently

$$r_1(0,a) = \sum_{z=0}^{a-1} \big(Gz - L(a-Z)\big) P[Z=z] + Ga \sum_{z=a}^{\infty} P[Z=z].$$

The first term corresponds to the situation when the demand z is less than the order quantity a. In this case, $a-z$ remain unsold resulting in a loss of L units per item. The second term corresponds to the situation when demand exceeds supply. Consequently $z-a$ units of demand are unmet and the reward is Ga.

Transition Probabilities: Since there is a single state $p_1(0|0,a) = 1$ for all $a \in A_0$.

A solution

The newsvendor's objective is to choose an order quantity a^* that maximizes $r_1(0,a)$ Thus the problem can be summarized as that of finding

$$a^* \in \arg\max_{a \in A_0} r_1(0,a).$$

One particularly attractive feature of this model is that if the demand distribution is known, the optimal order quantity can be shown to satisfy

$$a^* = F^{-1}\left(\frac{G}{G+L}\right), \tag{1.1}$$

where $F(z) = \sum_{u=0}^{z} P[Z=u]$. Since the product is assumed to be available only in discrete units, a^* may need to be rounded up or down.

Note that the ratio $(G/G+L)$, which is sometime refereed to as the *critical fractile*, represents to the fraction of demand that will be met. For example if $G = 2L$, two-thirds of the demand will be satisfied on average[7].

Application challenges

The newsvendor model has been widely applied in the retail fashion-goods industry but also applies to other perishable goods. One novel application (unpublished) was its use by one us to the determine the number of instructors to have available for drop-in ski lessons each day. In applications, the greatest challenge is to model the demand distribution, which may vary by day-of-week or month-of-year and change over time.

1.2 Revenue management: Using price to manage demand

This section describes a different approach to inventory management. It pertains primarily to perishable products that decline in value over multiple periods.

As a concrete example, consider the challenges faced by our friend, Frank Z., previous owner of a chain of women's fashion stores in Vancouver, Canada. One evening, on the way to a poker game, he told us[8] that:

Setting prices is like a game of chance; if I mark down prices too early in the season, I lose revenue, but if I wait too late, I can't sell my inventory and also must pay to store it.

A more general formulation of Frank's problem for a single product follows[9]. A retailer acquires M units of a product (for example, a size 10 woman's dress in a particular pattern and design) and prices it at a_K. The retailer hopes that fashion-conscious and well-off customers will purchase all of the inventory at this price but if not, some inventory will remain. To sell the remaining inventory, the retailer may choose to *markdown* the price in order to make it accessible to more price-sensitive customers.

More formally, prices are set at the beginning of each month, are chosen from a finite set of prices, and remain constant throughout the month. Assume that when the price is a the demand in period n is random and Poisson distributed with rate $\lambda_{n,a}$. It is reasonable to assume that $\lambda_{n,a}$ decreases in both time (n) and price (a). The rationale is that fashion products become less attractive as time goes on so that expected demand at a fixed price decreases over time. Moreover, fundamental economic principles suggest that in any month demand would increase when price was reduced[10]

[7] Assuming the newsvendor faces this situation on many occasions.

[8] Martin L. Puterman

[9] Frank stocks hundreds of products each season so that this problem must be solved hundreds of times each season; a daunting task without a formal model to do so.

[10] In economic modelling, this is referred to as a downward sloping demand curve.

Assume a monthly holding cost of $h(s)$ when the end of month inventory equals s units with $h(0) = 0$. Any goods left over at the end of month N are sold to an outlet store at a low price H per unit, representing the scrap value.

Decision Epochs: Prices are set at the beginning of each month, so

$$T = \{1, 2, \ldots, N\}.$$

States: States represent the number of items in stock at the start of each month in the planning horizon:

$$S = \{0, 1, \ldots, M\}.$$

Actions: Actions represent the possible set of prices to choose from each period. Assuming there are K candidate prices,

$$A_s = \{a_1, a_2, \ldots, a_K\}$$

for all $s \in S$. Assume that $H < a_1 \leq a_2 \leq \cdots \leq a_K$.

Rewards: If the inventory at the end of the month is j, that means $s - j \geq 0$ units were sold during that month. Thus, for $n < N$, $a_k \in A_s$, and $s \in S$,

$$r_n(s, a_k, j) = \begin{cases} a_k(s-j) - h(j), & j = 0, \ldots, s \\ 0, & j = s+1, \ldots \end{cases}$$

and $r_N(s) = Hs$.

Transition Probabilities: Since in this formulation no inventory is added during the planning horizon, only transitions to states with ending inventory $j \leq s$ have non-zero probability. If the demand equals or exceeds s, then the ending inventory is 0. This logic leads to the following transition probabilities:

$$p_n(j|s, a) = \begin{cases} e^{-\lambda_{n,a}} \lambda_{n,a}^{s-j}/(s-j)! & j = 1, \ldots, s \\ \\ \sum_{i=s}^{\infty} e^{-\lambda_{n,a}} \lambda_{n,a}^{i}/i! & j = 0 \\ \\ 0 & j = s+1, \ldots \end{cases}$$

Application challenges

Applying this model presents two challenges: determining the set of mark down prices and estimating the time-varying demand function parameters. In retail, the markdown prices can be set as a percentage of the original price such as "50% off" or "80% off". Estimating the demand function requires considerable amounts of data and may be product specific. Data from similar products may provide guidance when there is insufficient historical data for the product. The parameter $\lambda_{n,a}$ may itself be represented as a function of n and a and learned during the decision problem.

1.3 Discrete-time queuing models

A queuing system consists of arrivals, a queue and one or more servers. Jobs arrive, wait in a queue until if all servers are busy, are served by a free server and then depart the system when service is completed. Queuing systems have been well-studied in the operations research and engineering literature, and are applicable to a wide variety of service systems, including retail (jobs represent customers), healthcare (jobs represent patients), communication systems (jobs represent packets) and computer systems (jobs represent computing tasks).

From a decision perspective, the most widely studied models are:

1. **Service rate control:** The decision maker varies the service rate to control the queue length and throughput. The service rate may be controlled directly or through the addition and removal of servers.

2. **Admission control:** The decision maker chooses whether or not to admit an arriving job.

3. **Routing control:** In a network of queues, the decision maker chooses how to route jobs based on the workload at each queue.

The section formulates Markov decision process models for optimizing service rate and admission control. Exercise 8 provides a routing control example.

Some general comments regarding the formulation of discrete-time queuing systems follow:

1. Queuing systems are usually modeled as continuous-time Markov processes or semi-Markov processes. Here, we consider a discrete-time formulation. To do so assume observation of the system starts at time 0. Let h denote a "small" unit of time and let decision epoch n correspond to "time" nh. Then the set of decision epochs be denoted by $\{1, 2, \dots\}$ corresponding to times $\{h, 2h, \dots\}$. The time increment h is chosen to be sufficiently small so that it is very unlikely more than one event (an arrival or service completion) occurs during a period of length h.

2. Queuing systems are usually modeled with infinite planning horizons to reflect that they are on-going and decision epochs occur frequently. No terminal reward is specified.

3. Rewards and transition probabilities are assumed to be independent of the decision epoch.

4. The system state is the number of jobs in the queue *and* in service.

1.3.1 Service rate control

Consider a single-server infinite capacity queuing system. At each decision epoch, the decision maker chooses a service completion probability from the set $\{a_1, a_2, \ldots, a_K\}$. Let b denote the probability that a job arrives between two decision epochs, independent of the number of jobs in the system. Assume that $a_1 \leq a_2 \leq \cdots \leq a_K$ and $a_K + b \leq 1$. Figure 1.3 provides a schematic representation of this queuing system.

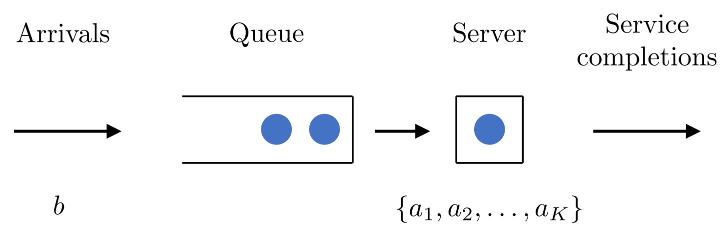

Figure 1.3: Schematic representation of a single server queuing system with adjustable service rate.

There are two costs to consider, a cost $m(a)$ per period for serving at rate a, and a delay cost of $f(s)$ per period when there are s jobs in the system at the start of the period. It makes sense to assume that both $m(a)$ and $f(s)$ are non-decreasing in their arguments.

A Markov decision process formulation follows. Note it assumes h is pre-specified and does not appear in the Markov decision process specification.

Decision Epochs:

$$T = \{1, 2, \ldots\},$$

corresponding to time points $h, 2h, \ldots$.

States: States represent the number of jobs in the system (queue plus server):

$$S = \{0, 1, 2, \ldots\}.$$

Actions: Actions represent the probability that a job currently being served is completed in the current period prior to the next decision epoch. For $s \in S$,

$$A_s = \{a_1, a_2, ..., a_K\}^{11}.$$

Rewards: Costs should be represented as negative rewards, so

$$r(s, a) = -m(a) - f(s).$$

Note that this reward function is independent of the subsequent state.

Transition Probabilities: For $s = 1, 2, \ldots$ and $k = 1, 2, \ldots, K$,

$$p(j|s, a_k) = \begin{cases} a_k & j = s - 1 \\ b & j = s + 1 \\ 1 - a_k - b & j = s. \end{cases} \tag{1.2}$$

For $s = 0$ and $k = 1, 2, \ldots, K$,

$$p(j|0, a_k) = \begin{cases} b & j = 1 \\ 1 - b & j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

The rationale for these probabilities follows. When $s = 0$, a transition only occurs when there is an arrival. When $s > 0$, the assumption that h is small comes into play. This means that three things can happen, the state increases to $s + 1$ when an arrival occurs, the state decreases to $s - 1$ when a service is completed, and the state remains the same when neither occur.

A truncated model

An alternate formulation suitable for direct computation requires *truncating* the state space and modifying some transition probabilities. This is because direct numerical computation is impossible with a non-finite discrete state space[12]. In such a truncated model, all other model elements remain the same.

Truncated states: For some $W > 0$,

$$S = \{0, 1, \ldots, W\}.$$

[11] Since the server is idle when there are no jobs in the system, it is optimal to use the slowest service rate so that one could set $A_0 = \{a_1\}$. Ideally the server should be turned off, but this formulation does not allow that. Moreover because h is small, so it might be impractical to turn off the server.

[12] However, such computation may not be needed if the structure of an optimal policy and value function can be succinctly encoded, e.g., a control limit policy or monotonic value function. See Section **??**

Truncated transition probabilities When $s = W$, the system becomes blocked so arrivals are not possible. This means that for $k = 1, \ldots, K$,

$$p(s|W, a_k) = \begin{cases} a_k & \text{for } s = W - 1 \\ 1 - a_k & \text{for } s = W. \end{cases}$$

Choosing W presents some challenges. A well controlled system will spend most of its time in low-occupancy states so an optimal policy might not be sensitive to the choice of W if it is chosen to be sufficiently large. A continuous approximation, where the mean and variance of the queue length can be computed in closed form, can provide some guidance. Alternatively W may be varied and its impact noted.

1.3.2 Admission control

In admission control models, the decision maker assumes the role of a "gate keeper" by deciding whether or not to admit an arriving job into a queuing system. This provides an example of a model with a vector-valued state space and non-actionable states, which occur when no job arrives in the preceding period.

Again assume discretized time intervals of length h where h is sufficiently small so that at most one job can arrive between decision epochs and does so with probability b. If not admitted, the job is lost. Let $f(j)$ be the holding cost when there are j jobs in the system at the start of a period (after the admission decision, but before an arrival or service completion in the same period). Let w be the probability of a service completion between in a time interval of length h. Admitting a job into the system, generates a payment of R. Assume $b + w \leq 1$, which is reasonable when the time discretization step h is small. Figure 1.4 provides a schematic representation of the system and Figure 1.5 depicts the one-period dynamics.

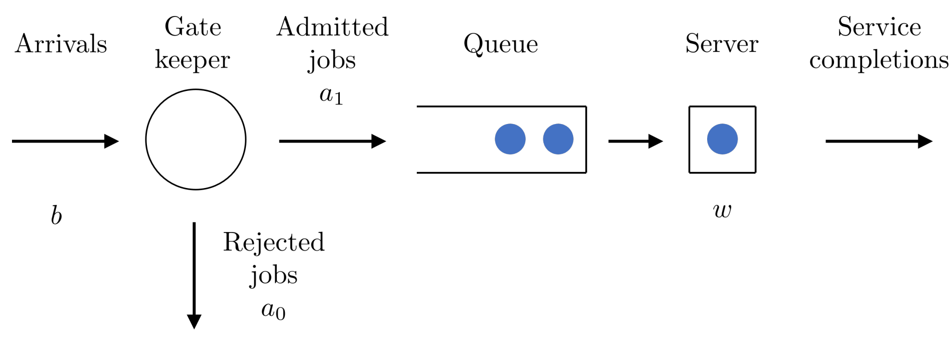

Figure 1.4: Schematic representation of a single server queuing system with admission control.

A Markov decision process formulation follows.

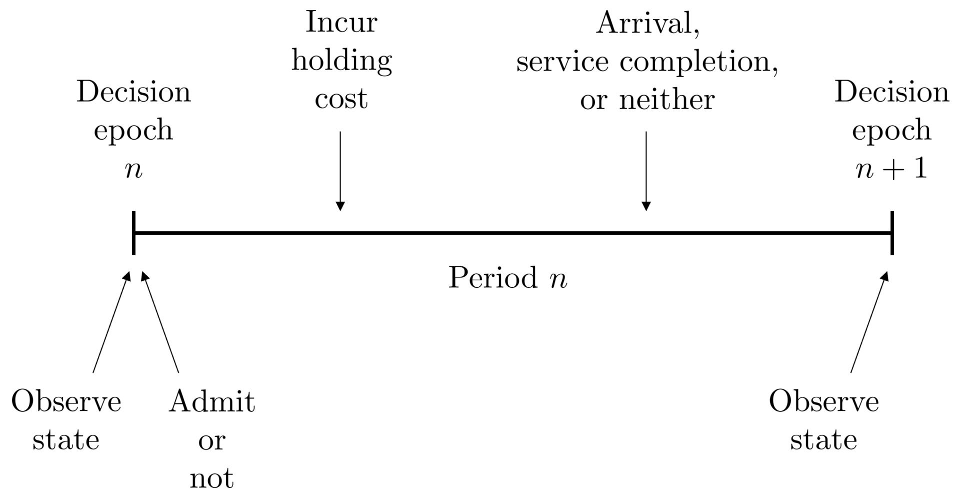

Figure 1.5: Timing of events in the queuing admission control model.

Decision Epochs:

$$T = \{1, 2, \dots\}$$

corresponding to time points $\{h, 2h, \dots\}$.

States: The state has two components. The first component, denoted j, represents the number of jobs in the system and takes values in $J = \{0, 1, \dots\}$, and the second component, denoted k, indicates whether there is a job waiting for admission ($k = 1$) or not ($k = 0$) so that $k \in \{0, 1\}$. Then the state space can be written as

$$S = J \times \{0, 1\}$$

with a typical state is represented by the two-dimensional vector (j, k).

Actions: Let a_0 correspond to the action "do not admit" and a_1 to the action "admit". Since admission is possible only if an arrival occurred since the previous decision epoch, there is no choice in states[13] where $k = 0$. Thus, for $j = 0, 1, \dots$,

$$A_{(j,k)} = \begin{cases} \{a_0, a_1\}, & k = 1 \\ \{a_0\}, & k = 0 \end{cases}$$

Recall that to formulate a Markov decision process model, actions need to be specified in all states even when the action set contains a single element.

[13]Such states are said to be *non-actionable*.

Rewards: For $j = 0, 1, \ldots,$

$$r((j,k),a) = \begin{cases} R - f(j+1), & k = 1,\ a = a_1 \\ -f(j), & k = 0,\ a = a_0. \end{cases}$$

Note that in this model, the rewards are independent of the subsequent state (j', k').

Transition Probabilities: If the action is to not admit ($a = a_0$), then whether there is a job currently waiting to be admitted or not ($k = 0$ or 1) is irrelevant since a waiting job will not be admitted. Thus, when $a = a_0$ for $j = 1, 2, \ldots$ and $k = 0$ or 1,

$$p((j',k')|(j,k),a_0) = \begin{cases} w & j' = j-1, k' = 0 \\ b & j' = j, k' = 1 \\ 1-b-w & j' = j, k' = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.4}$$

Since service completions are not possible if the system is empty, the transition probabilities when $j = 0$ and $a = a_0$ are given by

$$p((j',k')|(0,k),a_0) = \begin{cases} b & j' = 0, k' = 1, \\ 1-b & j' = 0, k' = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.5}$$

The "admit" action a_1 applies only when $k = 1$. So for $j = 0, 1, 2, \ldots$

$$p((j',k')|(j,k),a_1) = \begin{cases} w & j' = j,\ k' = 0 \\ b & j' = j+1,\ k' = 1 \\ 1-b-w & j' = j+1,\ k' = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{1.6}$$

In contrast to (1.4), (1.6) is valid when $j = 0$ since even when the system is empty at the decision epoch, a decision to admit will add a job to the system, which can be completed during the same period with probability w.

Figure 1.6 summarizes the possible state transitions for each action. Despite the simplifying assumption that at most one event can happen in a period, keeping track of all transitions requires a careful accounting of events and actions and their interactions. We revisit this example in Chapter **??**, which provides a simpler formulation of the model in terms of the *post-decision state* and *state-action value functions*. The post-decision state provides an alternative view of the decision timeline depicted in Figure 1.5 that allows the transition dynamics to be modeled more easily. As emphasized frequently in this chapter, drawing a correct timeline is an important part of formulating the model correctly.

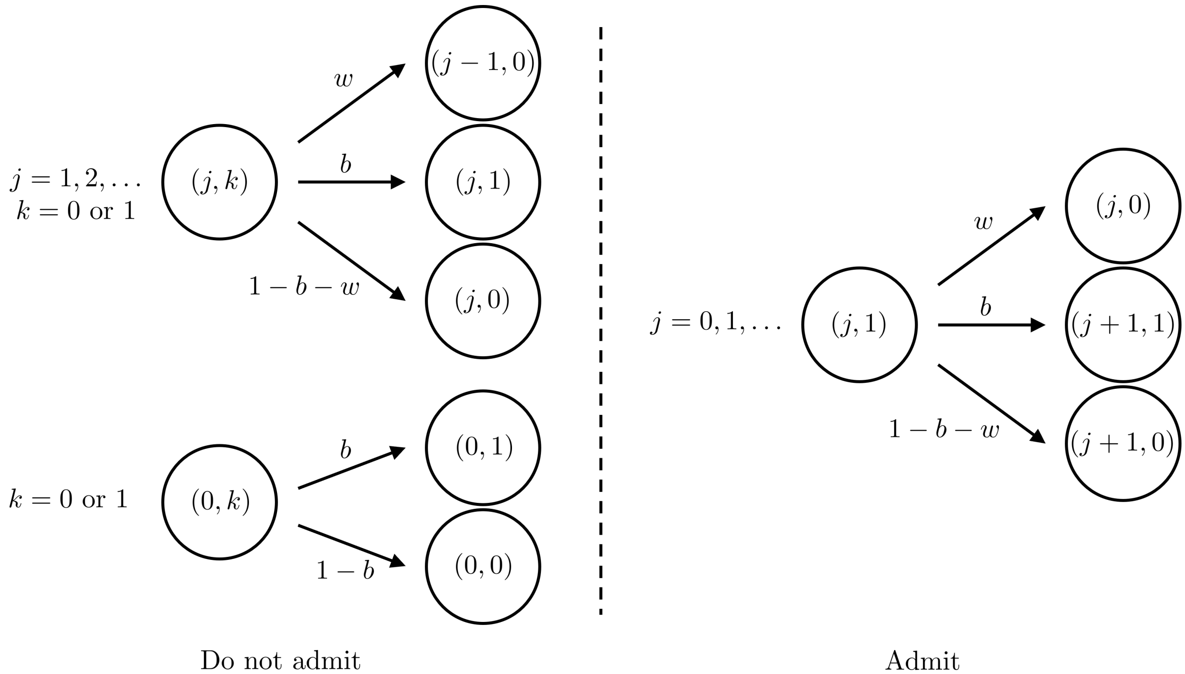

Figure 1.6: Possible state transitions for admission control problem. Note that not admitting a waiting job results in the same transitions as when no job arrives.

An alternative formulation

The following formulation is based on *post-decision states*. Shifting the timing of decision epochs leads to perhaps a simpler formulation. Suppose instead that decision epochs occur just prior to resolving the uncertain event "Arrival, service completion or neither" in Figure 1.5 so the timing now follows Figure 1.7. In this formulation at a decision epoch, the previous decision has been implemented so that the state need only represent the number of jobs in the system at this decision epoch. Actions need to be modified to be contingent on the resolution of the uncertain event as described below. Note that the holding cost is assessed after the action has been implemented.

Decision Epochs:

$$T = \{1, 2, \ldots\}$$

corresponding to time points $\{h, 2h, \ldots\}$ where $t = 1$ represents the time of the first decision.

States:

$$S = \{0, 1, \ldots\}$$

where $s \in S$ represents the number of jobs in the system at a decision epoch.

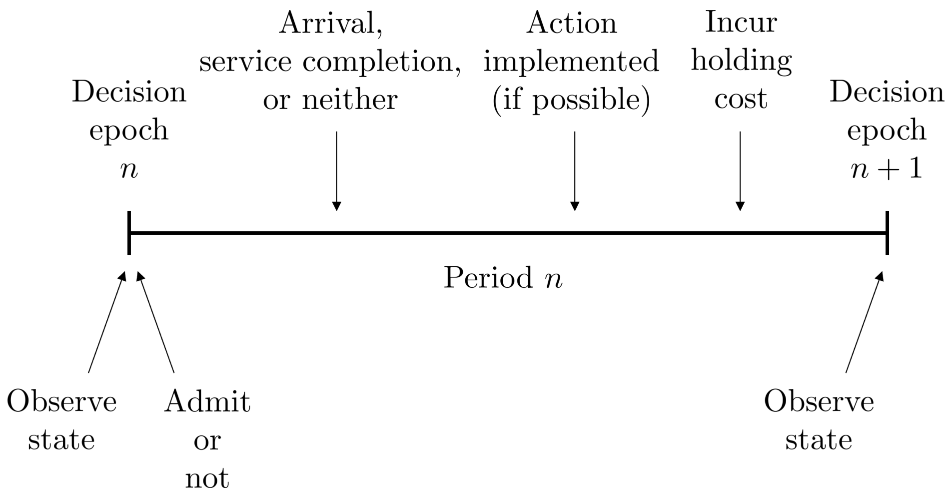

Figure 1.7: Timing of events in alternative formulation of the queuing admission control model. The expression "action implemented (if possible)" applies only if there is a prior arrival. Otherwise it cannot be implemented.

Actions: Let a_0 correspond to the action "do not admit" and a_1 to the action "admit". Since the decision maker does not know whether or not there will be an arrival, the result of this action must be contingent on the resolution of the event. Thus a_0 must be interpreted as "do not admit if there is an arrival" and a_1 as "admit if there is an arrival". By default if there is no arrival, the action a_0 applies since it will have the same effect as when there is an arrival. Thus for all $s \in S$,

$$A_s = \{a_0, a_1\}.$$

Rewards: In this formulation, rewards depend on the subsequent state and when the holding cost is assessed. Assuming the holding cost is incurred after the uncertain event is resolved and the decision implemented yields:

$$r(s, a_0, j) = \begin{cases} -f(j) & \text{for } j = s, s-1 \text{ and } s > 0 \\ -f(0) & \text{for } j = 0 \text{ and } s = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$r(s, a_1, j) = \begin{cases} R - f(j+1) & \text{for } j = s+1 \text{ and } s \geq 0 \\ -f(j) & \text{for } j = s, s-1 \text{ and } s > 0 \\ -f(0) & \text{for } j = 0 \text{ and } s = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.7}$$

Observe that the payment R is received only when choosing a_1 and an arrival occurs, which corresponds to the case when $j = s + 1$ in (1.7). In all other cases there are no admissions and only the holding cost is incurred.

Transition Probabilities: Under a_0 a service completion is the only possible event that can occur when $s > 0$. It results in a transition to $s-1$. When the system is empty, such a transition cannot occur so that:

$$p(j|s,a_0) = \begin{cases} 1-w & \text{for } j = s \text{ and } s > 0 \\ w & \text{for } j = s, s-1 \text{ and } s > 0 \\ 1 & \text{for } j = 0 \text{ and } s = 0 \\ 0 & \text{otherwise} \end{cases}$$

Under a_1, an arrival results in a transition to $s+1$. When there is no arrival the system either remains in the same state or moves to state $s-1$ if $s > 0$ if there is a service completion.

$$p(j|s,a_1) = \begin{cases} b & \text{for } j = s+1 \text{ and } s \geq 0 \\ w & \text{for } j = s-1 \text{ and } s > 0 \\ 1-b-w & \text{for } j = s \text{ and } s > 0 \\ 1-b & \text{for } j = 0 \text{ and } s = 0 \\ 0 & \text{otherwise} \end{cases} \tag{1.8}$$

We note that this formulation may be more natural when using state-action value functions and simulation or when formulating it directly as a continuous time model. We leave it to the reader to see which you prefer from the perspectives of ease of formulation and computation.

Application challenges

Applying these models requires estimates of arrival probabilities, service probabilities and costs. Queuing models are more commonly formulated in continuous time with inter-arrival and service times modeled using exponential random variables. Thus, if arrivals occur at rate λ, the probability of one arrival in an interval of length h is given by $\lambda h + o(h)$, the probability of no arrivals in an interval of length h is $1 - \lambda h + o(h)$, and the probability of greater than one arrival in an interval of length h is $o(h)$ where $o(h)$ is an expression that converges to zero as h decreases for zero. Thus, it is convenient to set the probability of an arrival in a short time interval of length h to be λh.

As in other models, determining costs and rewards is somewhat arbitrary, and may depend heavily on the application. It is important to investigate the impact of specific choices through sensitivity analyses.

1.4 Behavioral decision making: When should a lion hunt?

Markov decision processes provide a natural framework for modeling behavior when an organism faces a decision that trades off survival with reserving energy. Examples include choosing a location for food acquisition, deciding when to hunt for food, choosing a group size when hunting and deciding when to abandon its offspring. The primary objective in such research is the determine whether results from an optimization model agree with observed animal behavior.

As an example, consider the challenge facing a lion (*panthera leo*) when deciding to hunt for food. Suppose the lion seeks to maximize its probability of survival over a season of N days. A mature lion has an energy storage capacity of C units. Each day it does not hunt the lion depletes its energy reserves by d units. Hunting requires h units of energy with $h > d$. If its energy reserves fall below c_0 units, it will not survive to the next day.

At the start of each day, the lion decides whether or not to hunt and if so, what prey to seek. Typically, lions hunt impalas, gazelles, wildebeests, giraffes and zebras. About half of the time they hunt in groups. This development assumes that the lion hunts alone. (Exercise 20 asks you to formulate the group size decision problem.) Catch probability varies with species hunted. Assume that there are M species to choose from. Let w_m denote the probability that the lion catches an animal of species m; $m = 1, 2, \ldots, M$. A successful hunt for species m yields e_m units of energy. For simplicity, assume all relevant quantities have been discretized.

Decision Epochs: Decisions are made at the start of each day during the season, so

$$T = \{1, 2, \ldots, N\}.$$

States: The state represents the lion's energy reserves at each decision epoch. Thus

$$S = \{0, 1, \ldots, C\}.$$

Actions: Actions in state s may be denoted by

$$A_s = \begin{cases} \{a_0, a_1, \ldots, a_M\} & s \in \{c_0, c_0 + 1, \ldots, C\} \\ \{a_0\} & s \in \{0, 1, \ldots, c_0 - 1\}, \end{cases}$$

where action a_0 corresponds to "do not hunt" and action a_m corresponds to "hunt for species m".

Rewards: Given the lion's survival objective, the lion receives a reward of 1 if alive at the end of the season and a reward 0 if not. Therefore

$$r_N(s) = \begin{cases} 1 & s \in \{c_0, c_0 + 1, \ldots, C\} \\ 0 & s \in \{0, 1, \ldots, c_0 - 1\}. \end{cases}$$

No rewards are accrued throughout the planning horizon, so $r_n(s, a, j) = 0$ for $n = 1, 2, \ldots, N - 1$, $a \in A_s$, $s \in S$ and $j \in S$.

Transition Probabilities: When the lion has energy reserves of s units at the start of the day, hunts for species m and is successful, its energy reserves at the start of the next day is $\min\{s - h + e_m, C\}$. The logic underlying this observation is that starting in state s, the lion uses h units of energy hunting and if successful obtains prey that provides e_m units of energy. However, the total energy acquired cannot exceed it capacity C. As the result of an unsuccessful hunt, its energy reserves fall to $\max\{s - h, 0\}$. Therefore

$$p_n(j|s,a) = \begin{cases} 1 & j = \max\{s-d, 0\}, s = c_0, \ldots, C, a = a_0 \\ w_m & j = \min\{s-h+e_m, C\}, s = c_0, \ldots, C, a = a_m \\ 1 - w_m & j = \max\{s-h, 0\}, s = c_0, \ldots, C, a = a_m \\ 1 & j = s, s = 0, 1, \ldots, c_0 - 1, a = a_0 \\ 0 & \text{otherwise.} \end{cases}$$

Note that the transition probabilities take into account the fact that the lion's energy level cannot exceed the maximum capacity or fall below 0, and if its energy level falls below c_0, it cannot hunt.

Application challenges

This application highlights the fact that it can be challenging to determine parameter values for a Markov decision process, and to do so, one must often appeal to a wide range of sources. Moreover, in this particular example, it is essential to understand the underlying animal behavior and model it correctly, ideally with expert input. An added benefit of developing a formal Markov decision process model is that it identifies relevant parameters that can motivate related research.

The ecology literature suggests values for many of the key model parameters although not always exactly in the form needed. One can use $C = 30$ and $d = 6$ kilograms, based on averages of male and female lions. If a lion-specific parameter is not available from the literature, borrowing values from other species may provide some guidance. For example, wild dogs expend about 23% more energy per hour when hunting, with the average duration of a hunt approximately equal to 40 minutes. Noting that lions undertake on average three chases per day and assuming that they expend the same incremental amount of energy while hunting, on a day they decide to hunt, they will spend 3% more energy than on day they decide to rest so that $h = 1.03d$.

The literature suggests that gazelles yield a mean biomass of 12 kilograms with a catch probability of 0.15 on a single hunt. Observational data suggests that a lion may hunt up to three times a day if earlier hunts are unsuccessful; such dynamics can be incorporated into our model. Modeling the hunting of zebras presents additional challenges. Zebras yield an estimated 164 kilograms of edible biomass with a catch probability between 0.15 and 0.19 depending on the hunt location. Since they are large, their carcasses last for several days and are shared among several lions. Determining how much is available for the hunter requires further assumptions. Other sources provide estimates of the edible biomass for other types of prey such as impalas (29 kg), wildebeests (150 kg), and giraffes (468 kg), but not catch probabilities, which are harder to estimate. It is quite common for domain-specific literature to report data that allows us to estimate only some of the parameters of a Markov decision process, since such data is typically reported without a decision-making objective in mind. As a result, many assumptions must often be made, as described above. Especially when estimates are quite variable and many assumptions are needed, we recommend conducting sensitivity analyses of these parameters.

1.5 Clinical decision making: An application to liver transplantation

Markov decision processes have been widely applied to medical decision problems including organ transplantation, HIV treatment, cholesterol management and cancer diagnostics. As an illustration we describe a decision problem that arises when a patient requires a liver transplant.

We focus on the decision process for a patient with end-stage liver disease (ESLD)[14] who requires a liver transplant. Organs intermittently become available[15] and vary in quality. Depending upon the quality of the organ, the patient's medical team may either accept the organ and transplant it immediately, or reject it and wait for a higher quality organ.

To make this concrete, a relatively healthy patient may reject a lower-quality organ in the hopes of being offered a higher-quality organ in the future. On the other hand, a patient in poor health may accept the first liver available. A Markov decision process model can be used to formalize this decision problem and explore trade-offs.

To facilitate modeling, patient health status and organ quality are represented by discrete categories. Health states vary from 1 to H where 1 represents the healthiest state and H represents the least healthy state. The state Δ represents death by liver failure. Similarly, liver quality states are ordered from 1 (highest) to L (lowest) with state Φ representing the case where no liver is offered.

Rewards measure life expectancy in days. On a day when there is no transplant, the patient accrues a reward of 1 (an extra day of life) independent of the health state. The life expectancy post-transplant of a patient in health state h who accepts a liver of quality l is represented by $R(h, l)$. It is reasonable to assume (why?) that $R(h, l)$ is non-increasing (decreasing) in h and l.

Assume decisions are made at the start of each day. When a patient does not receive a transplant, the patient's health state either remains the same or deteriorates.

[14] *End-stage liver disease* or *cirrhosis* refers to a condition in which a patient's liver is severely damaged and no longer able to function adequately. Without a transplant, it is usually fatal.

[15] From a recently deceased individual. These are referred to as *cadaveric* organs in the medical literature.

Moreover whether or not a patient is offered a liver depends on a patient's health state. For each decision epoch, let

- $q(h'|h)$ denote the probability that a patient in health state h deteriorates to state h' with $h' = h, h+1, \ldots, H$,
- $\delta(h)$ denote the probability a patient in health state h dies from liver failure,
- $w(l|h)$ denote the probability that a patient in health state h is offered a liver of quality $l = 1, \ldots, L$ at the start of a decision epoch, and
- $\phi(h)$ denote the probability a patient in health state h is *not* offered a liver in a period.

Assume these distributions are stationary and independent. A Markov decision process formulation follows.

Decision Epochs: Assume decisions are made daily and that the horizon is infinite[16]. Thus,

$$T = \{1, 2, \ldots\}.$$

States: States represent the patient's health (if alive) and liver quality (if available). Let the absorbing state Γ represent the post-transplant state. For convenience, define $S_H = \{1, \ldots, H\}$, $S_L = \{1, \ldots, L\}$ and $S_L^+ = S_L \cup \{\Phi\}$. Then,

$$S = (S_H \times S_L^+) \cup \{\Delta, \Gamma\}.$$

Note there is no need to distinguish Δ and Γ since the decision process ends in either case.

Some explanation may be helpful. States of the form $(h, l) \in S_H \times S_L$, correspond to being in health state h and having an organ of quality l available. Φ replaces L when no organ is available and $s = \Gamma$ when no more decisions are possible.

Actions: Let a_t represent[17] the action to accept an organ (assuming one is available) and a_w represent the action to wait (do not accept an organ). Also let a_w represent the "do nothing" action, which applies in the death state, in the post-transplant state, and in any state when no organ is offered. Hence

[16] A practical upper bound on the number of decision epochs might be 26,000 (assuming no transplants for people over 90 or younger than 20). Given this upper bound, an infinite horizon model may be appropriate and simpler, especially since the process will reach an absorbing state eventually, either post-transplant or death most likely before reaching 90 years old.

[17] The subscript t corresponds to *terminate* the decision process by accepting an organ.

$$A_s = \begin{cases} \{a_t, a_w\} & s \in S_H \times S_L \\ \{a_w\} & s \in (S_H \times \{\Phi\}) \cup \{\Delta, \Gamma\}. \end{cases}$$

Note a_w applies when no organ is available, the patient has died or a transplant has occurred. Action a_t applies only when the patient is alive and an organ is available.

Rewards: The reward function may be represented for $s = (h, l)$ as:

$$r(s, a, j) = \begin{cases} R(h, l) & (h, l) \in S_H \times S_L,\ a = a_t,\ j = \Gamma \\ 1 & (h, l) \in S_H \times S_L^+,\ a = a_w,\ j = (h', l') \in \{h, \ldots, H\} \times S_L^+ \\ 0 & (h, l) \in S_H \times S_L^+, a = a_w,\ j = (h', l') \in \{0, \ldots, h-1\} \times S_L^+ \\ 0 & (h, l) \in S_H \times S_L^+, a = a_w,\ j = \Delta \\ 0 & s \in \{\Delta, \Gamma\},\ a = a_w,\ j = s. \end{cases}$$

Note the second and third expressions express the assumption that the health state cannot improve and that the patient does not accrue an additional day of life if death occurs on the current day.

Transition Probabilities: If the patient does not receive a transplant, then a transition from health state h to state h' with $h' \geq h$ may occur[18]. If the patient receives a transplant, then the transition is deterministic to the post-transplant state, Γ. In the post-transplant or death state, the system remains in that state, corresponding to a deterministic self-transition.

$$p(j|s, a) = \begin{cases} q(h'|h)w(l|h) & s = (h, l) \in S_H \times S_L^+,\ a = a_w,\ j = (h', l') \in \{h, \ldots, H\} \times S_L \\ q(h'|h)\phi(h) & s = (h, l) \in S_H \times S_L^+,\ a = a_w,\ j = (h', l') \in \{h, \ldots, H\} \times \Phi \\ \delta(h) & s = (h, l) \in S_H \times S_L^+,\ a = a_w,\ j = \Delta \\ 1 & s \in S_H \times S_L, a = a_t, j = \Gamma \\ 1 & s \in \{\Delta, \Gamma\}, a = a_w, j = s \\ 0 & \text{otherwise.} \end{cases}$$

Application challenges

Application of Markov decision process models to clinical decision making requires detailed medical domain knowledge including the nature and progression of the disease, and the treatment options and processes. For instance, applying this model to liver transplantation requires in-depth knowledge of the liver transplant system, the process

[18] Recall that smaller values of h correspond to better health.

used to allocate organs and the progression of end-stage liver disease. Data required includes discretized patient health and liver quality states, an estimate of post-transplant life expectancy, probabilities of health state deterioration, and arrival distributions of organs for transplantation by quality. Such data may be obtained from transplant centers and organizations that manage the transplantation system, such as the United Network for Organ Sharing (UNOS) in the United States. Patient health status can be measured using the Model for End Stage Liver Disease (MELD) score, which is a function of various laboratory values. The scores range from 6 to 40, with higher scores indicating poorer health and a higher mortality rate[19]. When data is sparse, MELD scores can be aggregated or smoothed. A similar approach can be taken when defining liver quality states, which may depend on donor age, race and sex. UNOS data may be used to estimate $R(h, l)$ (days of survival post-transplant) with a proportional hazards model[20] and organ arrival rates. Transitions between health states can be modeled using a natural history model[21].

1.6 Advance appointment scheduling

In many applications, decision makers must allocate scarce resources prior to the arrival of future random demand. The formulation below is from the perspective of a scheduler in a hospital diagnostic imaging department who faces the challenge of scheduling appointments for current medical imaging requests to meet patient-specific clinical wait time targets, without knowing how many and when future requests will arrive. Ideally, the scheduler would strive to minimize the fraction of patients scheduled beyond their target dates. The formulation[22] below introduces a cost to achieve this objective.

Suppose appointment requests arrive throughout the day and at the end of each day a radiologist assigns each request to one of K *urgency classes*. Urgency class k is associated with a target wait time of T_k days, $k = 1, 2, \ldots, K$, which is chosen based on clinical considerations. A patient in urgency class k should be scheduled prior to day T_k. The urgency classes are ordered, with 1 having the highest priority, so $T_1 < T_2 < \cdots < T_K$. If a patient in urgency class k is scheduled after T_k, a cost[23] C_k is incurred proportional to the number of days past T_k the appointment is scheduled. This cost can be thought of as a penalty related to worse clinical outcomes due to

[19]Internet sources suggest that the 3-month survival rate of 27% in patients with MELD score of 40 and 98% for patients with MELD score between 1 and 9.

[20]This is a statistical model that can be used to determine the impact of covariates on survival times when some patients in the data set are still alive.

[21]A commonly used epidemiological model that simulates disease progression accounting for different risk factors.

[22]Such an application presents many modeling challenges and requires formulating the problem carefully. The formulation described below was not immediately obvious to the investigators (see Section 1.11) and required considerable trial and error to achieve.

[23]Note these costs are artificial, wait time targets in medical settings are often specified in terms of the proportion of demand that is scheduled prior to its target date.

delayed diagnosis. Assuming $C_1 > C_2 > \cdots > C_K$ corresponds to setting higher delay costs for the most urgent cases. If a patient in class k is scheduled before the target T_k, no cost is incurred.

Let $p_k(w), k = 1, 2, \ldots, K$ denote the probability that w new class k appointments arrive each day. Let $\mathcal{W} = \{0, 1, \ldots, M\}$ denote the set of possible values of w. To ensure a finite state and action formulation, assume M is finite. Daily capacity is divided into B appointment slots. This means that at most B regular time appointments can be booked each day. Assume there are an unlimited number of overtime slots available and the system incurs a cost of h for each patient scheduled to overtime. Implicit is that $h > C_1$, which means that delaying a patient by a day beyond the target time is less costly than scheduling the patient to overtime. But for a sufficiently large delay, the cost of overtime will be less than the cost of delay.

Once all requests have an assigned urgency class, the scheduler assigns an appointment date to each request and informs the patient. Figure 1.8 provides a timeline for this process. The challenge is to schedule today's requests before realizing future requests in the face of limited capacity.

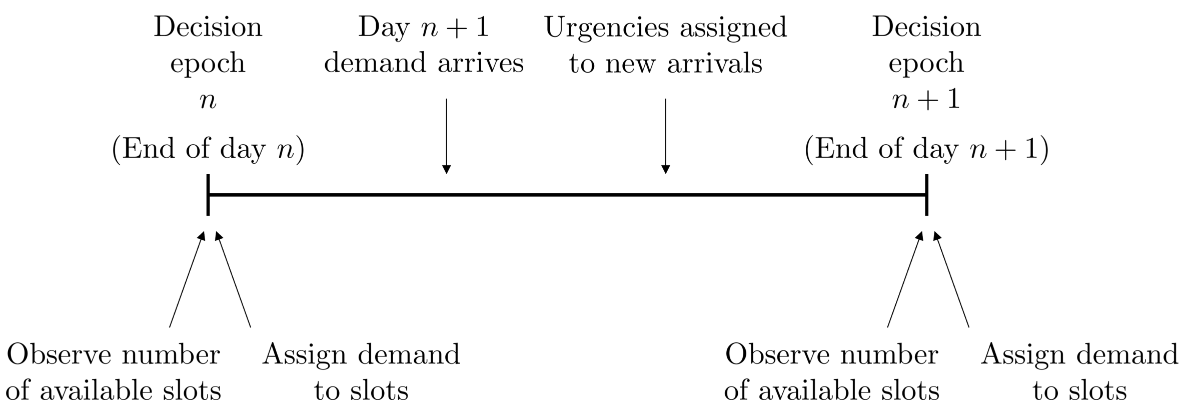

Figure 1.8: Timing of events in the advance appointment scheduling model.

There are numerous issues arising in this and other similar scheduling problems that can impact modeling:

1. Does the system have access to surge capacity or overtime?
2. How does appointment length vary between patients?
3. Can overbooking be used to account for patient no-shows and late cancellations?
4. Can appointment dates be changed after scheduling?
5. Are the targets flexible or must they be met?
6. Does demand have any seasonal patterns or correlation across urgency classes?

7. How far in advance can appointments be scheduled?

The formulation below assumes:

- access to overtime,
- fixed appointment lengths,
- no cancellations,
- no rescheduling,
- flexible target dates (but with a penalty for exceeding the target date),
- stationary arrivals and uncorrelated demand between urgency classes,
- a fixed *booking horizon* of N days.

Note that the booking horizon refers to how far into the future current appointment requests can be scheduled, and not the length of the planning horizon. As an alternative to overtime, appointments not scheduled during a particular day may be held over for scheduling in the future at some cost. Exercise 17 considers this variation.

Decision Epochs: Decision epochs correspond to the time in the day when the scheduling clerk assigns an appointment date to each appointment request waiting to be scheduled. This is naturally modeled as an infinite horizon problem, so

$$T = \{1, 2, , \ldots\}.$$

States: A typical state of the system is represented by $s = (b_1, \ldots, b_N, w_1, \ldots, w_K)$ where $b_i \in \mathcal{B} = \{0, 1, \ldots, B\}$ denotes the number of appointments that have already been booked on day i for $i = 1, \ldots, N$ and $w_k \in \mathcal{W}$ denotes the number of appointments of urgency class k waiting to be scheduled at a decision epoch. Note that if there are b_i appointments booked on day i, then there are $B - b_i$ remaining appointment slots on that day.

To simplify notation, introduce the vectors $\mathbf{b} := (b_1, \ldots, b_N)$ and $\mathbf{w} := (w_1, \ldots, w_K)$. Thus, a state is denoted by $(\mathbf{b}, \mathbf{w})$, and the state space is

$$S = \mathcal{B}^N \times \mathcal{W}^K.$$

Actions: Actions represent the number of waiting patients in urgency class k to schedule on each day within the booking window and possibly through overtime if B appointments if advantageous. Let x_{kn} denote the number of class k patients to book on day n and y_k denote the number of class k patients to book for overtime the next day. Note that it is not necessary to consider booking overtime slots at some point in the future since there is no limit on the number of patients booked through overtime, and booking overtime further in the future would simply increase costs.

Define the vectors $\mathbf{x} := (x_{11}, \ldots, x_{1N}, x_{21}, \ldots, x_{2N}, \ldots, x_{K1}, \ldots, x_{KN})$, $\mathbf{y} := (y_1, \ldots, y_K)$ and $\mathbf{0} := (0, 0, \ldots, 0)$ with lengths NK, K and $(N+1)K$, respectively. The action set $A_{(\mathbf{b},\mathbf{w})}$ is

$$A_{(\mathbf{b},\mathbf{w})} = \Bigg\{ (\mathbf{x}, \mathbf{y}) \geq \mathbf{0} \;\Bigg|\; \sum_{n=1}^{N} x_{kn} + y_k = w_k \text{ for } k = 1, \ldots, K \text{ and } b_n + \sum_{k=1}^{K} x_{kn} \leq B \text{ for } n = 1, \ldots, N \Bigg\}. \quad (1.9)$$

The first condition in (1.9) ensures that all class k requests must be scheduled either to a specific day or to overtime. The second condition ensures that at most B patients may be scheduled to regular time each day.

Rewards: Assume that the penalty associated with scheduling an urgency class k request n days from the current day costs $C_k(n - T_k)^+$. This function specifically models the cost as linear in the number of days a class k appointment is scheduled beyond its target, while incurring zero costs for scheduling prior to the target. Exercise 18 considers the variation where instead of the target representing a fixed day, a target window with both and earliest and latest time is assumed[24].

The reward for choosing actions $(\mathbf{x}, \mathbf{y})$ in state $(\mathbf{b}, \mathbf{w})$ is

$$r((\mathbf{b}, \mathbf{w}), (\mathbf{x}, \mathbf{y})) = -\sum_{k=1}^{K} \sum_{n=1}^{N} C_k(n - T_k)^+ x_{kn} - h \sum_{k=1}^{K} y_k.$$

The reward (negative of cost) captures the costs associated with exceeding the target times as well as overtime costs for the current set of appointment requests. Observe that the reward does not depend on the next state since the subsequent $\mathbf{b}$ vector is a deterministic function of the current action.

Transition Probabilities: Transition probabilities depend on both action choice and the random arrival distribution. At the start of a period, the calendar rolls forward one day[25] so previous bookings that were n days from the previous decision epoch are now $n-1$ days from the current decision epoch. Added to these bookings are the newly arriving demand that is booked over the N-day horizon starting at the current decision epoch. Finally, since there were no bookings $N+1$ days from the previous decision

[24]This applies to chemotherapy schedules where there is a narrow window when a treatment can be given.

[25]This is often referred to as a *rolling horizon*.

epoch, there are 0 appointments booked N days from the current decision epoch. The demand that has arrived since the last decision epoch is the only stochastic element in this model. Once this random demand has been assigned to urgency classes, the probability transitions are:

$$p((\mathbf{b}', \mathbf{w}') \mid (\mathbf{b}, \mathbf{w}), (\mathbf{x}, \mathbf{y})) =$$
$$\begin{cases} \prod_{k=1}^{K} p_k(w_k') & \mathbf{b}' = (b_2 + \sum_{k=1}^{K} x_{k2}, b_3 + \sum_{k=1}^{K} x_{k3}, \ldots, b_N + \sum_{k=1}^{K} x_{kN}, 0) \\ 0 & \text{otherwise.} \end{cases} \tag{1.10}$$

Some comments on this formulation follow:

1. The concept of a booking horizon may be regarded as an artifact of the modeling process and imposed to maintain a finite state space. Since overtime is unlimited, a fixed booking horizon applies. Without overtime, an unbounded booking horizon may be needed.

2. Because the booking horizon remains constant between decision epochs, but moves forward each day, the model may be regarded as using a *rolling horizon* model.

3. Note that the transitions decompose into a deterministic part corresponding to the number of booked appointments each day and a random part corresponding to the random demand for each urgency class.

4. When the maximum daily demand for each urgency class is M, the model has $(C+1)^N(M+1)^K$ states. This makes direct computation infeasible for practical sizes of these parameters and motivates the need for approximation (see Chapter **??**).

5. After an action is implemented at decision epoch n, the $\mathbf{b}$ component of the state does not change until after decision epoch $n+1$. For reasons discussed in Section **??** and Chapter **??**, it may be more convenient to formulate the model in terms of post-decision states.

6. In reality, many possible reward structures may be applicable for this problem. For example, as an alternative to incurring costs for appointments scheduled beyond their target date, a decision maker could strive to maximize the fraction of patients scheduled within their target dates.

Application challenges

Application challenges include specifying a booking horizon, specifying how unscheduled demand is satisfied, determining urgency classes and targets, specifying costs for delays, and estimating demand.

In real problem settings, a booking horizon may be determined by the decision maker based on their typical clinical processes. It has also been shown that when unlimited appointment diversion, such as through overtime, an optimal policy is independent of the booking horizon provided it exceeds the largest wait time target. Urgency classes should be defined based on clinical guidelines by medical professionals. The above model formulation was based on a real application. In that application the classes were "urgent" (7 day target), "semi-urgent" (14 day target) and "non-urgent" (28 day target). Emergency cases were scheduled to a different resource. Relative delay costs can potentially be quantified by calculating the impact on clinical outcomes of delayed treatment due to the delayed imaging.

Future demand may be forecasted using historical data. In practice, these distributions may be non-stationary as volumes generally increase over time. As an alternative to estimating the demand for each urgency class separately, if the historical data suggests that the relative proportion of cases from each urgency class is stable, it may be best to estimate total demand and then split it among each class based on the fixed proportion.

1.7 Grid World navigation

"A mathematician is a machine for turning coffee into theorems."
Alfred Renyi, Mathematician, 1921-1970

A working mathematician frequently requires coffee and to avoid time away from theorem proving, may employ a robot to bring coffee when needed. In the example described here, the robot's task is to carry the mathematician's empty cup from the office to the coffee room, fill the cup with coffee and bring it back to the mathematician's office, all while avoiding barriers and pitfalls such as an open stairwell. Figure 1.9 depicts a possible grid the robot must navigate to retrieve and deliver coffee.

In general, *grid worlds* represent simplified environments commonly used as examples to illustrate reinforcement learning methods. They possess a grid-like structure (like a chess board) where each cell in the grid represents a discrete state that an agent can occupy. The agent learns to move through the grid, typically aiming to achieve a goal while navigating around obstacles, collecting rewards, and avoiding penalties.

The formulation in this section assumes that the robot *knows* the arrangement of the grid, which grid cell it occupies and the location of boundaries, target cells and penalties. This means the robot will never attempt to move outside the grid boundary. Alternative formulations may assume:

1. the robot does not know its location,

2. the robot does not know the configuration of the grid, its boundaries, or the location of the stairwell,

3. the robot does not know the uncertainty involved when choosing which direction to move, or

4. some combination of the above.

In all of these alternative situations, navigating through the grid becomes a learning problem similar to many studied in the reinforcement learning literature.

The problem description below assumes that the robot knows the grid configuration and which cell it occupies. In each cell except the stairwell, regardless of whether the coffee cup is empty or full, the robot can move in any of the four directions that does not take it outside the grid boundary. If the robot falls into the stairwell, or returns to the office with a full coffee cup, the episode terminates[26].

Assume movement on the grid is subject to uncertainty as follows. Let p_E and p_F denote the probability that the robot moves in its intended direction when the coffee cup is empty and full, respectively. If in some cell there are k possible locations where the robot can end up (including the current location), the probability the robot moves in each unintended direction or remains where it was is $(1 - p_E)/(k - 1)$ or $(1 - p_F)/(k - 1)$, depending on the status of the coffee cup. For example, suppose the robot has a full coffee cup and is in cell 6. Then if it intends to move down, it does so with probability p_F and it moves left, up or remains in the same location with probability $(1 - p_F)/3$. The robot is more likely to be error prone with a full coffee cup because of the energy required not to spill the coffee. Thus, assume $p_E > p_F$.

The goal is to return with a full coffee cup as quickly as possible. If the robot successfully delivers coffee to the mathematician it receives a reward of B. If it falls down the stairs it incurs a penalty of X because the mathematician has to interrupt work to rescue the robot. Assume $X \gg B \gg 1$.

Decision Epochs: Assume the process evolves in discrete time, where decision epochs correspond to the instant at which the robot decides in which direction to move. The first decision epoch after the robot enters the coffee room is used to fill up the cup, and the next one corresponds to the instant immediately after the cup has been filled and it decides where to move next. Thus

$$T = \{1, 2, \dots\}.$$

The set of decision epochs is unbounded because the robot continues attempting to deliver the coffee to the mathematician until the random time when it is either successful or falls down the stairs.

States: States represent the location of the robot in the numbered grid, plus an additional variable to indicate whether or not the coffee cup is empty (E) or full (F).

[26]In some numerical examples in subsequent chapters, the episodes terminate only when the robot successfully delivers coffee.

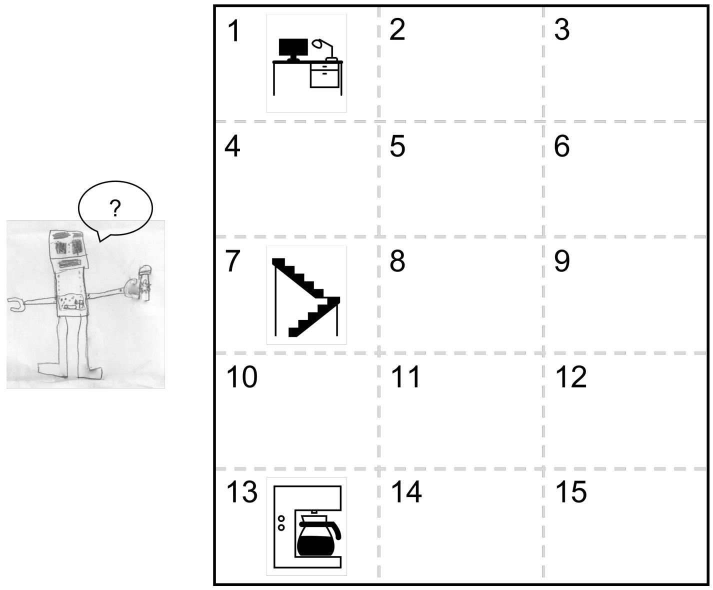

Figure 1.9: Schematic layout for Grid World navigation example.

The status of the coffee cup is required because it informs the success probability of an intended action. It also influences the direction in which the robot should proceed: a robot with an empty coffee cup seeks the coffee room, while a robot with a full coffee cup seeks the office. Therefore,

$$S = \{1, 2, \ldots, 15\} \times \{E, F\}.$$

Actions: Actions represent the direction the robot attempts to travel. Assume the robot can only move up (north), down (south), right (east) and left (west), denoted by U, D, R and L, respectively. Under the assumption that the robot knows the layout of the grid and its location, the grid boundary constrains its intended movement so the set of permissable actions must take this into account. For example,

$$A_{(5,\cdot)} = \{U, D, R, L\},\ A_{(7,\cdot)} = \{U, D, R\},\ \text{and}\ A_{(15,\cdot)} = \{U, L\}.$$

Action sets for some particular states follow :

$$A_{1,F} = A_{7,F} = A_{7,E} = \{a_0\},\ A_{13,E} = \{a_1\}.$$

The states $(1, F), (7, F)$ and $(7, E)$ are *absorbing* states. Once entered, the robot remains there forever; action a_0 corresponds to remaining in that state. Once such a state is entered, decision making stops. When the robot enters the coffee room with an empty cup, it takes one period to fill it. Denote the action of filling the cup by a_1.

Rewards: Each intended movement action and the act of filling the coffee cup costs c. Often $c = 1$. Transitions into the office with a full coffee cup result in a reward of $B - c$. For example,

$$r((2, F), a, (1, F)) = B - c, \quad a \in A_{(2,F)} \tag{1.11}$$

Transitions into the stairwell receive a reward of $-X - 1$ regardless of the status of the cup. For example,

$$r((4, k), a, (7, k)) = -X - c, \quad a \in A_{(4,k)} \text{ and } k \in \{E, F\}. \tag{1.12}$$

All other feasible state transitions between neighboring cells result in a reward of $-c$. For example,

$$r((5, k), a, (6, k)) = -c, \quad a \in A_{(5,k)} \text{ and } k \in \{E, F\}. \tag{1.13}$$

Finally,in this formulation, no rewards are received (or costs incurred) once the robot completes its task or falls down the stairs:

$$r((1, F), a_0, (1, F)) = 0 \text{ and } r((7, k), a_0, (7, k)) = 0,\ k \in \{E, F\}. \tag{1.14}$$

Note that in the absence of uncertainty, it is easy to see that the robot can complete its task in 13 steps, so the maximum possible reward is $B - 13c$.

Transition Probabilities: Some typical probabilities follow:

$$p((k, F)|(5, F), U) = \begin{cases} p_F & k = 2 \\ (1 - p_F)/4 & k \in \{4, 5, 6, 8\} \\ 0 & k \notin \{2, 4, 5, 6, 8\} \end{cases}$$

$$p((k, E)|(6, E), D) = \begin{cases} p_E & k = 9 \\ (1 - p_E)/3 & k \in \{5, 6, 9\} \\ 0 & k \notin \{3, 5, 6, 9\} \end{cases}$$

Transitions are deterministic when the system is in one of the absorbing states or when the robot enters the coffee room with an empty cup (since the only action is to fill the cup):

$$\begin{aligned} p((13, F)|(13, E), a_1) &= p((1, F)|1, F), a_0) \\ &= p((7, F)|(7, F), a_0) = p((7, E)|(7, E), a_0) = 1. \end{aligned}$$

A key feature of this model is that the robot must trade off between safe but slow and risky but fast policies. This type of trade-off is characteristic of the key tension in many applications modeled by Markov decision processes. Here, by trying to reach the coffee room and returning to the office by the shortest route, the robot risks a high probability of falling down the stairs. If robot motion with an empty cup does not involve any randomness, that is $p_E = 1$, the robot will travel from the office to the coffee room by the shortest path but most likely take a more cautious path when returning to the office with a full cup.

Application challenges

The above example is artificial but Markov decision processes have been widely applied to robotic control. Realistic challenges include:

1. converting sensor readings to state variables,
2. transforming actions to motor commands,
3. representing variability in intended movement,
4. providing the robot with a mapping of the area, and developing reward structures.

These challenges are amplified when the application involves robotic movement in three-dimensional space.

1.8 Optimal stopping

An elegant collection of applications are referred to as "optimal stopping problems". Examples include selling an asset, finding a parking spot, and online dating. Optimal stopping problems have attracted considerable research effort, which has primarily focused on showing that optimal policies have intuitively appealing structure.

In optimal stopping problems, the system evolves as a (possibly non-stationary) Markov chain on a set of states S' with transition probabilities $b_n(j|s)$ for $s \in S'$ and $j \in S'$ at epoch n. If the decision maker decides to "stop" in state s at decision epoch n, a reward of $g_n(s)$ is received. If the decision maker decides to "continue," the decision maker incurs a cost $f_n(s)$. In the finite horizon case, when the problem terminates after N decision epochs, the decision maker receives a reward $h(s)$ if the Markov chain is in state s at epoch N.

1.8.1 Model formulation

Decision Epochs: As noted above, this can be either a finite or infinite horizon model, so

$$T = \{1, \dots, N\}, \quad N \leq \infty.$$

States: The state space is the union of S' and a state Δ that denotes the stopped state:

$$S = S' \cup \{\Delta\}.$$

Actions: Let the action C (for *continue*) denote the decision to not stop and Q (for *quit*) represent the stopping decision. Then the action set is

$$A_s = \begin{cases} \{C, Q\} & s \in S' \\ \{C\} & s = \Delta. \end{cases}$$

We include the action to continue in the stopped state for completeness.

Rewards: The reward does not explicitly depend on the destination state j, so for $n < N$

$$r_n(s,a) = \begin{cases} -f_n(s) & s \in S', a = C \\ g_n(s) & s \in S', a = Q \\ 0 & s = \Delta, a = C, \end{cases}$$

with $r_N(s) = h(s)$ for $s \in S'$ if the horizon is finite. In the infinite horizon case, it is more appropriate that $f_n(s) = f(s)$ and $g_n(s) = g(s)$ for all $s \in S'$ and $n = 1, 2, \ldots$.

Transition Probabilities: For $n \leq N$

$$p_n(j|s,a) = \begin{cases} b_n(j|s) & s \in S', a = C, j \in S' \\ 1 & s \in S', a = Q, j = \Delta \text{ or } s = j = \Delta, a = C \\ 0 & \text{otherwise.} \end{cases}$$

Again, in the infinite horizon case, $b_n(j|s) = b(j|s)$ for $n = 1, 2, \ldots$.

1.8.2 Examples

Selling an asset

A homeowner who is moving to another city has N days to sell a house. Offers arrive throughout the day and by the end of the day, the homeowner has to decide whether to accept the best offer received that day, or wait until the next day for new offers. The set S' represents the set of possible values of the best daily offer for the house, assumed to be around its market value (i.e., bounded) and rounded to the nearest dollar (finite). By waiting until the next day, the homeowner incurs costs $f_n(s)$ related to continued home ownership such as maintenance, mortgage interest, property taxes, and advertising. By accepting the best offer on a given day, the homeowner receives a reward of s, minus the costs associated with selling the house, $L_n(s)$, which includes realtor fees and taxes. Hence $g_n(s) = s - L_n(s)$. At day N, the homeowner must accept the best offer that day, receiving a terminal reward of $h(s) = s - L_N(s)$. The best offer j at decision epoch $n + 1$ is determined by a probability distribution $b_n(j|s)$ that may be conditional on the best offer s in epoch n. This distribution may be non-stationary. For example, early in the planning horizon, rejections could signal that the homeowner expects higher offers in the future than the current best offer. Later in the planning horizon, if bidders know the homeowner must sell, the offers might decrease in value.

Finding a parking spot

A driver seeks a parking spot as close as possible to a restaurant. Assume the driver can move in one direction only and see only one spot ahead. If the spot is not occupied, the driver may choose to park in it or proceed forward. The probability that any spot is unoccupied is assumed to be equal to p, independent of all other spots.

The most natural formulation is as a stationary infinite horizon model with an infinite state space. Elements of S' consist of vectors, (j, k), where j indicates the location of the parking spot and k indicates whether the spot is free (1) or occupied (0). Let $\mathbb{Z} = \{\ldots, 2, 1, 0, -1, -2, \ldots\}$ represent the distance of each parking spot to the restaurant where 0 denotes the location of the restaurant, positive numbers denote locations before the restaurant and negative numbers denote locations past the restaurant. Thus, $S' = \mathbb{Z} \times \{0, 1\}$.

States of the form $(j, 1)$ are the only ones in which the action Q corresponding to stopping is available. No rewards are received if the individual does not park, that is $f\big((j, C)\big) = 0$. When the individual parks, the reward is the distance from the parking location to the restaurant, so that $g((j, 1)) = -|j|$.

Transition probabilities of the underlying Markov chain are given by

$$b\big((j', k')|(j, k)\big) = \begin{cases} p & j' = j - 1, k = 1 \\ 1 - p & j' = j - 1, k = 0. \end{cases} \tag{1.15}$$

This can be reduced to a finite state formulation by assuming:

- the driver will not start looking for a spot until reaching a distance M before the restaurant, and
- that if the driver has not parked before reaching the restaurant the driver will choose the next available spot.

As a result of the second assumption, the last decision is made in state $(1, 1)$ because the driver will definitely park in state $(0, 1)$ and continue in state $(0, 0)$. Hence the terminal reward is

$$h\big((0, k)\big) = \begin{cases} 0 & k = 1 \\ -\frac{1}{p} & k = 0. \end{cases}$$

To see this, note that from 0 onward, the distance to the next available parking spot follows a geometric distribution with parameter p so that

$$\big|h\big((0, 0)\big)\big| = 1p + 2(1 - p)p + 3(1 - p)^2 p + \ldots = \frac{1}{p}.$$

Online dating

This example provides a modern take on a classical problem that is often referred to as the *secretary problem*[27].

An individual is searching for dates on a dating app. The dating app shares a brief profile for each potential match, including a photo and description, one at a time. For each potential match, the individual can either "swipe left" to pass or "swipe right" to indicate interest. If the individual swipes left, that profile will not be shown again. If the individual swipes right, a match is made and the two will go on a date. The app offers a free trial until the first match is made or until N profiles have been viewed, whichever comes first. The individual is interested in maximizing the probability of finding the best match during the free trial.

Assume that the N potential matches have an unobservable ranking from 1 to N, with 1 representing the best match. Through the process of examining the profiles, the individual will be able to rank the candidates seen so far relative to each other in a manner that is consistent with the true ranking. If the best profile is seen, the individual will only know that it is the best seen so far, but will not know whether any future profiles will be better. The order of profiles is completely random, so every permutation of the ranks 1 to N is equally likely.

The following formulation may not be immediately obvious, but is the most succinct way to model the problem. Let $S' = \{0, 1\}$, where the state indicates whether the current profile is the best seen so far (1) or not (0). No rewards or costs are accrued if the individual swipes left ($f_n(s) = 0$). In this formulation, rewards correspond to the probability of choosing the best candidate and are only received upon stopping. If the individual reaches the last profile, the match is automatically made. The terminal reward is thus $h(1) = 1$ and $h(0) = 0$, since the last profile is either the best profile seen so far or not.

If the individual swipes right on the n-th profile, $n \leq N$, and it is not the best profile seen so far, then the probability that the n-th profile corresponds to the best match is $g_n(0) = 0$. But if it is the best profile seen so far, then $g_n(1) = n/N$. To see why this is true, write out the probability that $g_n(1)$ represents explicitly:

$$
\begin{aligned}
g_n(1) &:= P(\text{profile } n \text{ is rank 1} \mid \text{profile } n \text{ has the highest rank of first } n \text{ profiles}) \\
&= \frac{P(\text{profile } n \text{ is rank 1 "and" profile } n \text{ has the highest rank of first } n \text{ profiles})}{P(\text{profile } n \text{ has the highest rank of first } n \text{ profiles})} \\
&= \frac{P(\text{profile } n \text{ is rank 1})}{P(\text{profile } n \text{ has the highest rank of first } n \text{ profiles})} \\
&= \frac{1/N}{1/n} = \frac{n}{N}.
\end{aligned} \tag{1.16}
$$

The second equality follows from the definition of a conditional probability. The third

[27] A closely related problem was first formulated by Cayley [1875] and described in the historical summary in Chapter **??**.

equality is due to the fact that if profile n is the top ranked profile, it must be the top ranked profile within the first n profiles as well. Finally, the fourth equality is due to the fact that the order of the profiles is completely random. So the top ranked profile is equally likely to be in any of the N positions. Similarly, any of the first n profiles is equally likely to be the one with the highest rank.

To determine the transition probabilities, again appeal to the fact that the order of the profiles is completely random. Thus, regardless of the current state, the probability that profile $n+1$ will be the best among the first $n+1$ is $1/(n+1)$. Thus, the transition probabilities are

$$b_n(j|s) = \begin{cases} \frac{1}{n+1} & j = 1, s \in \{0, 1\} \\ \frac{n}{n+1} & j = 0, s \in \{0, 1\}. \end{cases} \tag{1.17}$$

An elegant solution to this problem is provided in Chapter **??**.

1.9 Sports strategy

Analytical methods have recently found widespread use in sport decision making, providing many opportunities for applying Markov decision processes. Some applications involve decisions made throughout a game while others are situational. Situational decisions such as whether to "go for it" on fourth down in North American football or whether to steal a base or sacrifice in baseball concern a decision in a particular state in a model of the whole game. They reduce to one-period problems (Section **??**) when the value function for all games states is estimated from historical data. Other examples such as those described below concern recurrent decisions throughout a game or some portion of it.

1.9.1 When to pull the goalie in ice hockey

In ice hockey, a team has the option of replacing its goalie with an offensive player at any time during a game. This is called "pulling the goalie". Doing so can be beneficial since there is a greater probability of scoring when an extra offensive player is in the game. However, pulling the goalie also results in a greater likelihood that the opponent scores, since the goal is undefended. Such a goal is referred to as an "empty net goal". Such a strategy is often employed late in a game in order to achieve a tie and send the game to overtime.

This decision problem is naturally modeled in continuous time but in keeping with the development of the book, a discrete time formulation follows. Assume that every h seconds a decision is made whether or not to pull the goalie, and that such a decision is not considered prior to M seconds remaining in the game. Usually, M is on the order of 180 seconds (3 minutes).

For concreteness, assume that Team A trails Team B by g goals. Let p_A (p_B) denote the probability Team A (Team B) scores one goal in an interval of length h when Team

A pulls its goalie and let w_A (w_B) denote the probability Team A (Team B) scores a goal in an interval of length h when Team A does not pull its goalie. Naturally, $p_A > w_A$ and $p_B > w_B$. It may also be the case that no team scores during an interval of length h, so $p_A + p_B < 1$ and $w_A + w_B < 1$. As is customary when discretizing continuous time models, assume that h is sufficiently small so the likelihood of scoring more than one goal in that interval is negligible.

Decision Epochs: Because decisions are made every h seconds up to M seconds,

$$T = \{1, 2, \ldots, N\},$$

where $N = M/h$, the first decision epoch corresponds to M seconds left in the game, the second corresponds to $M - h$ seconds left in the game, and so on.

States: The state represents the goal differential, defined as Team B's goals minus Team A's goals. Let G be the maximum goal differential at which the coach would consider pulling the goalie. Then

$$S = \{0, 1, \ldots, G + 1\}.$$

This formulation adds the absorbing state $G + 1$ to ensure a finite state space. If the goal differential reaches $G + 1$ the coach of Team A will not consider pulling the goalie anymore. If the score differential returns to G because Team A scored a goal without its goalie pulled, the decision problem starts anew. In practice, $G = 3$. A team would not pull its goalie if it is leading, so negative values are omitted from the state space. The state 0 corresponds to a tie score, which also is an absorbing state and the objective of pulling the goalie.

Actions: In all states the coach has the option to not pull the goalie (action a_0) or to pull the goalie (action a_1) when the goal differential is between 1 and G. Thus

$$A_s = \begin{cases} \{a_0\} & s = 0 \text{ or } G + 1 \\ \{a_0, a_1\} & s = 1, \ldots, G. \end{cases}$$

Rewards: Since the objective is to tie or win by the end of the planning horizon, rewards are only received at termination. So $r_n(s, a) = 0$ for $n < N$ and all s and a. The terminal reward is given by

$$r_N(s) = \begin{cases} 0 & s > 0 \\ 1 & s = 0. \end{cases}$$

Transition Probabilities: Under the assumption that one event can occur in a time interval of length h, the transitions probabilities satisfy

$$p_n(j|s,a) = \begin{cases} w_A & j = s-1, s = 1, \dots, G, a = a_0, \\ w_B & j = s+1, s = 1, \dots, G, a = a_0, \\ 1 - w_A - w_B & j = s, s = 1, \dots, G, a = a_0 \\ p_A & j = s-1, s = 1, \dots, G, a = a_1, \\ p_B & j = s+1, s = 1, \dots, G, a = a_1, \\ 1 - p_A - p_B & j = s, s = 1, \dots, G, a = a_1, \\ 1 & j = s = 0, G+1, a = a_0, \\ 0 & \text{otherwise.} \end{cases}$$

Application challenges

The key parameters in this model are the relative scoring probabilities. They may vary by team and also depend on the whether the opposing team has been assessed a penalty so that pulling the goalie results in a "two-man advantage" and an increased scoring probability.

The following data comes from the (North American) National Hockey League for the 2013-2020 seasons. When both teams are at full strength (no player is in the penalty box), teams score goals at the rate of 2.25 goals per 60 minutes. When a team pulls its goalie, its scoring rate increases to 6.39 goals per 60 minutes. However, the opposing team's scoring rate increases to 19.16 goals per 60 minutes. Assuming $h = 5$ seconds, these goal scoring rates correspond to $w_A = w_B = 0.003125$, $p_A = 0.008875$ when pulling the goalie and $p_B = 0.02661$ for an empty net goal. On average, teams pull their goalie around 4 minutes prior to end of the game with a three goal deficit, 2.3 minutes with a two goal deficit, and 1.4 minutes with a one goal deficit. The success rate for pulling the goalie with a one goal deficit is about 14%.

Penalties play a large role in ice hockey. Pulling the goalie when Team B is penalized[28] significantly affects the scoring probabilities, increasing p_A and decreasing p_B relative to the previous values. The same data shows that pulling the goalie when the opposing team has one penalized player increases Team A's scoring rate to approximately 12 goals per 60 minutes ($p_A = 0.0167$), and decreases Team B's scoring rate to approximately 11 goals per 60 minutes ($p_B = 0.0153$).

1.9.2 A tennis handicap system

This section proposes a handicapping system for tennis. Consider a tennis match between two players of unequal skill level. In order to have a fair (and enjoyable) match, the stronger player (Player B) offers the weaker player (Player A) a handicap. The

[28]The consequence of a penalty is that one or more fewer players are "on the ice".

handicap takes the form of a budget of "credits" that Player A can use to win a point without playing it. To our knowledge, such a system has been yet to be widely applied but conceptually presents an interesting strategic challenge: when should Player A use these credits?

Before describing how such a handicapping system may be employed, we briefly review relevant aspects of tennis scoring. A tennis match consists of games and sets. A player wins a game by scoring four points first, provided that the player "wins by at least two points." That is, if both players have scored three points, the winning score needs to be at least 5-3. A player wins a set by being to first to win 6 games, again with a "win by two game" rule in effect. If the set score reaches 6-6, then a tiebreaker is played, with the winner winning the set by a score of 7-6. Finally, a match is typically the best two out of three sets or best three out of five sets.

A key feature of this scoring system is that it is "hierarchical," the match score decomposes into sets and games, each with its own scoring system. Thus the score with Player A serving may be 1-1 in sets, 5-3 in games and 3-1 (commonly referred to as 40-15) in the current game. Faced with this situation, Player A could use a handicap point to win the game, and hence the set and the match.

Unlike sports such as soccer, in which every goal contributes equally to the final score, not every point is equally valuable in tennis. In fact, a player could win more than 50% of the total points in the match, but still lose the match, due to the hierarchical nature of the scoring system.

To make this precise, if Player A uses a credit at the start of a point, then Player A wins the point, the handicap budget is decremented by one, and the players proceed to play the next point. If Player A decides not to use a credit, then the point is played as usual. Let p_1 (p_0) be the probability[29] that Player A wins the point on serve (return) if it the point is played out. Assume that the budget applies to the entire match and that Player A can use a credit regardless of which player is serving.

Decision Epochs: The start of each point is a decision epoch. Given the scoring system described above, the horizon is infinite but with a random stopping time corresponding to the end of the match. Thus

$$T = \{1, 2, \ldots\}.$$

States: The state comprises the current match score, q, the budget of credits remaining, b, and an indicator for the serving player, k. Suppose the set of possible match scores is $Q = \{q_1, q_2, \ldots\}$, the starting number of credits is B, and $k = 1$ (0) indicates that Player A (Player B) is serving. The two absorbing states, W and L correspond to Player A winning and losing the match, respectively. Thus, the state space is

$$S = (Q \times \{0, 1, \ldots, B\} \times \{0, 1\}) \cup \{W\} \cup \{L\}. \tag{1.18}$$

[29] In most matches the probability of winning a point on serve is greater than when the opponent is serving.

Each state q represents a set score, a game score and a within-game score. Note that B is at most 24 times the number of sets[30] to win the match. In a best two-out-of-three sets match, $B \leq 48$ and in a best three-out-of-five sets match B is at most 72.

Actions: Actions are to use a credit (a_1) or not (a_0) at each decision epoch when there are credits remaining. Once no credits remain, the only action is a_0.

$$A_s = \begin{cases} \{a_0, a_1\}, & s = (q, b, k) \text{ with } b > 0 \\ \{a_0\}, & \text{otherwise.} \end{cases} \tag{1.19}$$

Rewards: Since Player A's objective is allocate handicap credits so as to maximize the probability of winning the match, the only non-zero reward is when there is a transition to state W, in which case the reward equals 1. Thus

$$r(s, a, s') = \begin{cases} 1, & \text{if } s \neq W, a \in A_s, s' = W \\ 0, & \text{otherwise.} \end{cases} \tag{1.20}$$

Similar to the online dating application, to maximize the probability of an event, the model formulation should be set up so that decision maker receives a reward of 1 only when that event occurs. This follows directly from the observation that the expected value of an indicator variable equals the probability of the indicated event.

Transition Probabilities: If a credit is used, then there is a deterministic transition to the state in which Player A has one extra point. Using such a credit could also affect the game and set score, and even result in winning the match. Otherwise, the transitions follow the point-winning distribution of Player A against Player B. For convenience, let q^+ (q^-) denote the score if Player A wins (loses) the point when the current score is q. Similarly, let k_q^+ (k_q^-) denote the server if Player A wins (loses) the point when the current score is q and the current server is indicated by k. The server changes only if by using the credit, Player A wins the current game. Thus, given that the current state is $s = (q, b, k)$,

$$p(j|s, a) = \begin{cases} 1, & \text{if } j = (q^+, b-1, k_q^+), a = a_1 \\ p_k, & \text{if } j = (q^+, b, k_q^+), a = a_0 \\ 1 - p_k, & \text{if } j = (q^-, b, k_q^-), a = a_0 \\ 0, & \text{otherwise.} \end{cases} \tag{1.21}$$

[30]Since it requires 4 points to win a game, 24 points is the minimum number of points needed to win a set.

Application challenges

The primary challenge in the application of this model revolves around the estimation of the point-win probabilities p_0 and p_1. Such probabilities depend on several factors include the strength of the opponent (perhaps considering the specific opponent and previous head-to-head successes), the court surface, the weather, and recent playing history. These probabilities are likely to be non-stationary as well due to factors like fatigue or injury.

1.10 The art of modeling

One learns to formulate Markov decision processes by studying how others have done so and trying out for themselves. Formulations that appear particularly crisp are likely the result of numerous iterations in formulating and re-formulating the problem. By being exposed to and working through many different examples, one starts to build an intuition for how certain problem types are formulated. Below, we present a systematic approach to model formulation.

How to formulate a Markov decision process model

- *Clearly define the problem.* Verbally describe what the decision maker wishes to achieve, what information is available on which to base decisions, how the system responds to these decisions, and what rewards or costs are incurred as a consequence of the decisions taken. A precise problem description facilitates identifying all model components. Often, however, one must return to, revise or redefine problem characteristics to ensure that the Markov decision process model properly represents the specified situation. The examples above present problem statements that are self-contained, with the information needed to fully formulate the problem.

- *Draw a timeline of events.* Carefully specify when the **state** information becomes available, when **actions** are chosen (i.e., the **decision epochs**), when **rewards** are received and when **transitions** occur. Changes to the timing of events can impact the specification of certain model components. Several examples in this chapter illustrate the sequence of events in a typical period. Selected problems at the end of the chapter ask you to reformulate models under modified assumptions about the timing of events.

- *Identify decision epochs.* Specify the precise time at which actions are chosen, using the timeline as a guide.

- *Determine the planning horizon.* Applications may have a horizon that is finite with fixed length, finite with variable length or infinite. Variable length models arise when the policy or realization of a probability take the system to an absorbing state such as in the the lion hunting model (Section 1.4), liver transplant model (Section 1.5), the Grid World model (Section 1.7) and the optimal stopping models (Section 1.8). Most infinite horizon models may be transformed into finite horizon problems through an appropriate reformulation or simply through truncation. An example is provided in the parking problem (Section 1.8.2).

- *Identify states.* The main challenge when formulating a Markov decision process is determining an appropriate state space. Doing so requires taking into account all other model components. Hence, this is the most important step. A well-defined state space can make the rest of the formulation appear obvious; a poorly defined state space may result in complicated (non-Markovian) dynamics, extra computational burden, or insufficient information to write down a complete model. The advance appointment scheduling (Section 1.6) and online dating application (Section 1.8.2) illustrate two challenging examples of state space formulation.

  States should encapsulate all of the information available to the decision maker (and no more than is necessary) to specify actions, rewards and transition probabilities. Do not include actions as part of the state space except when one needs to know the current action to decide on future actions[31]. Often, a time component is required for decision making, but the decision epoch itself may be sufficient to avoid including an extra variable in the state space. Some models may require the addition of zero-reward absorbing states to account for early or random termination times.

- *Specify actions.* Be sure to note whether different sets of actions may be available in each state. Since the Markov decision process formulation requires specifying an action for each state, in *non-actionable* states, that is where there is no meaningful action choice such as an absorbing state, the set of actions should be specified as a single element representing the "do nothing" action.

- *Determine rewards.* Rewards may be stationary or vary with decision epoch. In finite horizon models, be sure to specify a terminal reward function. Also, note whether rewards depend on the subsequent state. It may be possible to model a problem both ways, with rewards that do or do not depend on the next state. However, usually one of these two is more natural. Some of the above examples sometimes use a reward function that does not depend on subsequent state. When using criteria based on expected reward, rewards that depend on subsequent sates my be replaced by their expectations.

[31] As an example consider the queuing service rate control model when there is a fixed cost for changing an action. In this case one needs to know the current action to determine if a switching cost applies.

In cases when the decision maker seeks to maximize the probability of an outcome, such as surviving or winning, specify a reward of zero in all states not corresponding to that outcome and a reward of one when that outcome occurs. The reason for this is that the expected value of an indicator of an event equals the probability of the event occurring.

In some reinforcement learning formulations there is no explicit reward function available or its realization is delayed far into the future. In episodic models, the reward may correspond to winning a game or reaching a target state so that the reward is only learned when the episode is completed. In such cases it may prove useful to modify the reward function to indicate progress towards a goal.

While a Markov decision process is generally concerned with maximizing rewards, its formulation, and the reward function specifically, should be independent of the specific choice of optimality criterion such as expected total reward, expected discounted total reward, long-run average reward, or expected utility. Also, note that in many applications, a decision maker may seek to minimize costs, which can be regarded as negative rewards. Since the Markov decision process formulation here seeks to maximize rewards, costs are best regarded as negative rewards.

- *Specify transition probabilities.* These are often quite complicated and contain many special cases. It is important to appeal to the timeline and the order of events when writing down the transitions. Challenges include taking into account "edge cases" at state space boundaries and noting that some components of the state may evolve deterministically, while others may evolve stochastically. In the presence of absorbing states, be sure to note that under the "do nothing" action the system remains in that state with probability one. For completeness, be sure to note zero probability transitions corresponding to impossible combinations of states and actions, which can be captured under the catch-all heading "otherwise".

  Recall that a Markov decision process with a fixed policy results in a Markov reward process that evolves over a Markov chain. Drawing a Markov chain with directed arcs indicating transitions and denoting probabilities on arcs is a simple but effective method to help ensure that all transitions are accounted for (e.g., probabilities leaving a state for a given action sum to 1) and that they make sense (e.g., transitions occur between states as described in the problem statement). With complicated multi-dimensional state spaces, drawing such a picture is often a must. Figure 1.6 provides an example.

- *Estimate model parameters.* Most of the examples in this chapter are abstracted from real problem situations. To apply the models in concrete settings, one must estimate the model parameters such as transition probabilities and rewards.

  In some cases, such as inventory control (Section 1.1), revenue management (Section 1.2 and queuing control (Section 1.3) the transition probabilities may be derived from parametric distributions with the parameters estimated from historical data. When there are no parametric forms for transition probabilities, care must be taken in estimating probabilities because some may be non-zero but very small. In applications such as the lion hunting model (Section 1.4), clinical decision making (Section 1.5) and sports strategy (Section 1.9) one may appeal to the data and literature from those fields to obtain parameter estimates.

Another challenge when applying models in practice is specifying rewards. In applications where rewards refer to concrete monetary values, specifying rewards can be relatively straightforward. In examples such as lion hunting (Section 1.4), optimal parking and online dating (Section 1.8.2), pulling the goalie (Section 1.9.1), and tennis handicapping (Section 1.9.2) the reward is implicit in the chosen model objective. In other cases, rewards may be derived in consultation with the decision maker.

1.11 Bibliographic Remarks

Inventory models (Section 1.1) date back at least to Arrow et al. [1951] and Dvoretsky et al. [1952]. The book of Arrow et al. [1958] is an important early reference. Porteus [2002] provides an overview of the historical development in his book on inventory models. Much current research in inventory theory is subsumed under the heading "supply chain management". Section 8.9.2 of Puterman [1994] provides an inventory example of a constrained MDP with a service level constraint.

The newsvendor model seems to originate with Edgeworth [1888] where it is developed to determine optimal cash reserves to meet random withdrawals from a bank. Arrow et al. [1951] derived the critical fractile solution. It is now described in all operations management textbooks; Chen et al. [2016] provide a modern survey.

The model in Section 1.2 is in the spirit of Gallego and van Ryzin [1994] who provide one of the first examples of dynamic pricing. The book of Talluri and van Ryzin [2004] provides a comprehensive overview of revenue management. Dynamic pricing combined with overbooking has been applied extensively in the airline industry where it is referred to as yield management [Smith et al., 1992].

The queuing control models in Section 1.3 have mostly been studied in continuous time. Early references include Yadin and Naor [1967], Heyman [1968], Naor [1969] and Sobel [1969]. Our formulation of the discrete time service rate control model follows de Farias and van Roy [2003] where they use it to illustrate approximate dynamic programming methods.

The lion hunting example in Section 1.4 is adopted from Clark [1987]. That paper provides many of the parameter values described in the Application challenges section, including the energy storage capacity, daily energy depletion, biomass yield and catch probabilities of gazelles and zebra. The estimate of energy expenditure while hunting was based on Hubel et al. [2016]. Edible biomass of other prey listed were taken from

Smuts [1979]. Other applications in ecology include Mangel and Clark [1986], Kelly and Kennedy [1993] and Sirot and Bernstein [1996].

Numerous applications of using Markov decision processes in clinical decision making have appeared in the literature. The model described herein is based on Alagoz et al. [2007], who focus on liver transplantation. The discussion around application challenges is based on methods they employed to specify their model and estimate parameters. Other examples of clinical decision making using Markov decision processes include Shechter et al. [2008], who consider HIV therapy, and Kurt et al. [2011], who model statin treatments for diabetes patients.

The formulation of the advance scheduling model in Section 1.6 follows Patrick et al. [2008]. The result about an optimal policy being independent of the booking window if the window is longer than the largest wait time target and if the system has access to unlimited appointment diversion is given in that paper. Sauré et al. [2012] and Goçgun and Puterman [2014] analyze variants of this model. An example of using the impact of delayed treatment to quantify the cost of delayed imaging appointments is appears in Sauré et al. [2012] in the context of radiation therapy.

A Grid world model appears in Sutton and Barto [2018]. Such models have been widely used in the computer science community to illustrate Markov decision process and reinforcement learning concepts.

Optimal stopping problems originate with early work of Wald [1947], Wald and Wolfowitz [1948] and Arrow et al. [1949]. Karlin [1962] proposes and solves the asset selling problem. The optimal parking problem appears in Chow et al. [1971]. The online dating (i.e., secretary) problem was first proposed by Cayley [1875] in the context of evaluating a lottery.

Sports applications have appeared broadly. The path-breaking monograph of Howard [1960] introduces many of the key Markov decision process concepts, and contains an example of using Markov decision processes in baseball strategy. Carter and Machol [1971] and Chan et al. [2021] develop value functions in football. The pulling the goalie model in Section 1.9.1 originates with Morrison [1976]. A dynamic programming formulation was provided in Washburn [1991]. Hall [2020] provides the recent data quoted in the Applications Challenge section. The tennis handicapping model in Section 1.9.2 appears in Chan and Singal [2016]. The amazing book by Kemeny and Snell [1960] includes a Markov Chain model for a tennis game.

1.12 Exercises

1. Formulate a periodic inventory control problem in which orders arrive after demand has been fulfilled. Clearly show how the timing of events in Figure 1.1 changes.

2. Formulate a periodic inventory control problem in which sales are lost if demand exceeds supply in a period.

3. Formulate a periodic inventory control model in which there is limited storage capacity, limited backlogging and a bound on order size. Assume if the quantity backlogged exceeds its bound, an extra cost incurred.

4. Formulate the reward function of the inventory management example when the revenue of the inventory sold is included.

5. (A newsvendor model with one replenishment opportunity) Formulate the following variant of the newsvendor model as a Markov decision process. Assume at the start of the period that the newsvendor purchases units of a product from the supplier at a cost of c_1 per item but can purchase additional units items at a pre-specified later time during the period at a cost c_2 with $c_2 > c_1$. Such a problem arises when sales exceed initial expectation early in the period so that it may be opportune for the newsvendor to purchase additional item half-way through the period.

6. Formulate a service-rate control queuing model with a fixed cost C for changing the service rate. Note that as alluded to above in Section 1.10 it is not sufficient to just modify the reward function.

7. Formulate a combined admission and service rate queuing control problem as a Markov decision process.

8. (Call center routing) Consider a call center with monolingual (English-only) and bilingual (English and French) call takers. Assume there are two call takers of each type. English-speaking and French-speaking customers call in to the center and indicate their preferred language. English-speaking customers can be served by either type of call taker, but French-speaking customers can only be served by a bilingual call taker. Every minute an English-speaking customer calls in with probability p_E and a French-speaking customer calls in with probability p_F, where $p_E + p_F < 1$. We assume the probability of more than one caller per epoch is negligible. Customers incur a cost of C for every minute spent waiting before service. The duration of a call is geometric with parameter q, regardless of the language. Formulate this routing control problem as a Markov decision process.

9. (Control of a tandem queuing network) Consider a discrete-time queuing system composed of two single-server queues in tandem and two types of jobs. Type 1 jobs arrive at queue 1 and after completing service by server 1 also require service by server 2. Type 2 jobs arrive directly to queue 2 and require service at server 2 only. In each period, no job arrives or either a type 1 job arrives, a type 2 job arrives or one of each type arrives. Assume the probability of an arrival of a type i job is p_i independent of the whether or not the other type job arrives.

   Assume a finite buffer (waiting room) of size M_i in front of queue i. When the buffer is full jobs are blocked (lost) at penalty cost c_i. Completion of a type i job

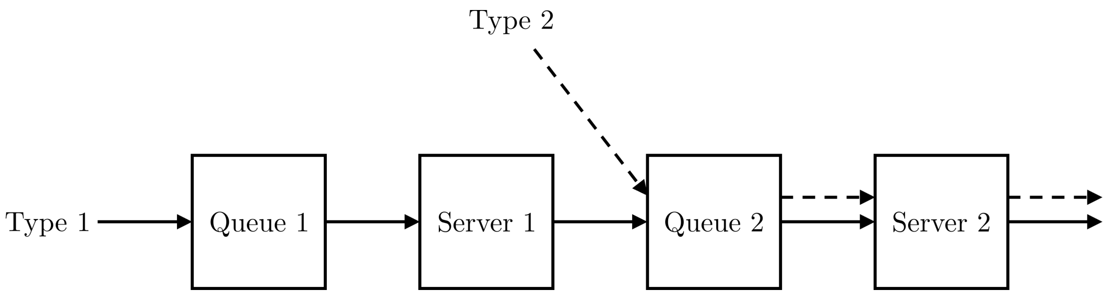

Figure 1.10: Tandem queuing network

yields revenue R_i with $R_1 > R_2$. In addition, assume a holding cost of $h(s_1, s_2)$ when there are s_i type i jobs in the system (either in the queue or in service).

Formulate the following revenue maximization problems as Markov decision process.

(a) (Job selection) In each period the controller of queue 2 can choose whether to serve a type 1 or type 2 job. Assume that the service is completed with probability q_2 independent of job type and whether or not a job at queue 1 has completed service with probability q_1. To simplify the formulation assume that if the service is not completed in the current period, the job reverts to the queue prior to the start of the subsequent period.

(b) (Service rate control) In each period the system can choose the service probability at queue 1 from the set $\{q_{1,1}, q_{1,2}, \ldots, q_{1,N_1}\}$ with cost $f_1(q)$ that is non-decreasing in q.

(c) Describe and formulate other possible control problems that can apply to this configuration.

10. Formulate a version of the Grid world navigation model in which there is a positive probability that the robot drops the coffee cup, which is larger if the cup is full. Assume the cup is breakable and if it is dropped, the robot needs to return to the office to retrieve another one.

    Clearly state any assumptions you are making in formulating this model.

11. Formulate a finite horizon version of the Grid World problem in which if the robot does not return with coffee after N decision epochs, the mathematician gets his or her own coffee and and incurs a penalty of C units. How should C be related to X and R?

12. (Equipment maintenance) Formulate the following maintenance problem. You own a piece of equipment that deteriorates over time. While it is operating, it contributes revenue of r dollars per month. When it has been operating for i months since its last maintenance, the probability it fails in the current month

is $p(i)$ and the probability it does not fail is $1 - p(i)$. If it fails at any time during a month the cost of repairing it is c_A and it is available for use at the start of the subsequent month. Assume that if it fails during a month, no revenue is generated during that month. On the other hand, the maintenance manager can schedule preventive maintenance in a month at cost c_B. Assume preventive maintenance is always scheduled at the beginning of the month, starts in the first week of the month and takes one month.

(a) Draw a timeline for the decision problem and clearly state any assumptions you are making.

(b) Formulate the maintenance manager's problem as a Markov decision process. Be clear to state any assumptions you make.

(c) In a real application, how do you think $p(i)$ will vary with i and what would be the relationship between c_A and c_B?

(d) Propose a "real life" application of this model.

13. [Stengos and Thomas, 1980] Consider the following generalization of the previous problem. You own two machines which sometime require maintenance that takes three weeks. Maintenance on one machine costs c_1 per week while maintenance on two machines costs c_2 per week. The probability either piece of equipment breaks down if it has operating for i weeks is $p(i)$. Assume that if the equipment breaks down during a week, maintenance begins at the start of the next week. However, if you decide to perform preventive maintenance, you do so at the start of a week. Moreover, assume that the two pieces of equipment break down independently.

    (a) Formulate this problem as an infinite horizon Markov decision process.

    (b) How are c_1 and c_2 related? Why?

14. Reformulate Exercise 13 assuming that the time it takes to complete maintenance on a piece on equipment is random. In particular, assume that maintenance is completed in any period with probability q, independent of other periods.

15. [Bertsimas and Shioda, 2003] A restaurant contains both two-seat and four-seat tables. Parties of two and four arrive randomly and request service. No reservations are taken. When a party of four arrives and a four-seat table is empty, they should be seated but should the manager ever seat a party of two at a four seat table? If so, when? Also, when should requests for service be denied, if ever?

    Formulate this problem as Markov decision process assuming the following, unrealistic as it may be. Decisions are made every 10 minutes and the restaurant operates 24 hours a day. There are two two-seat and two four-seat tables. Meals consist of two courses, durations of each are geometrically distributed independent of party size. Course 1's completion probability per epoch is 0.7, while course 2's is 0.8. In any period there is at most one arriving party. A party of two arrives with probability 0.2 and a party of four with probability 0.1. Assume that the waiting area holds at most 6 people. If it is full, arrivals are blocked and do not enter. Also any waiting party may leave in a 10 minute period with probability 0.05.

Revenue is as follows. A party of 2 contributes \$50 and a party of 4 contributes \$100. The cost of waiting (incurred by the restaurant) is \$6 per person per hour.

16. In the organ transplantation problem, consider an extension where outcomes depend on the quality of the organ that is offered. Modify the formulation to account for this possibility.

17. Formulate the advance scheduling problem when appointments not booked on first day available are added to the next days demand with cost C'.

18. Formulate the advance scheduling problem where instead of the target representing a fixed day, target windows are used for each urgency class. Let T_k^l and T_k^u be the lower and upper limits of the target window for urgency class k. If an appointment is scheduled within this window, no costs are incurred. If a class k appointment is scheduled before (after) T_k^l (T_k^u), then a cost of C_k^l (C_k^u) is incurred.

19. Modify the lion hunting problem to take into account that on any day, the lion may be captured by poachers with probability $1 - \lambda$. How is the related to discounting?

20. At the start of each day, a lion decides whether or not to hunt and if so, in what group size. The probability of catching prey varies with group size. This presents a trade-off, a larger group has a greater probability of a successful hunt, but then less food is available for each lion in the group. Assume a maximum group size of M and that all captured prey is split evenly among the group. Let λ_m, $m = 1, 2, \ldots, M$, denote the probability that a group of of size m is successful in its hunt. Assume the prey being hunted yields a total edible biomass of e units. Thus if the lion hunts in a group of size m and is successful, it receives e/m units of edible biomass.

21. Reformulate the original lion hunting behavior model so that the objective is to maximize the number of days of survival instead of the probability of survival. Clearly note what changes are necessary.

22. The ride-sharing driver's dilemma. At random times throughout the day, a ride-sharing driver receives offers of potential trips, including their expected revenue and time to complete the trip. The driver can either accept the trip or decline it and wait for the next offer. Formulate this problem as a discrete time Markov decision process clearly stating all assumptions being made.

23. Consider a variant of the online dating problem in which the decision maker's goal is to maximize the probability of choosing one of the two best candidates. How would you modify the formulation to take this into account.

24. Consider a variant of the online dating problem in which the decision maker's goal is to maximize the rank of the selected date. Modify the formulation accordingly.

25. Reformulate the tennis handicapping problem to account for second serves. That is, if a player misses a first serve, they have a second chance to get the ball in before losing the point. Suppose the probability that the first serve goes in is q_1 and the probability the second serve goes in is q_2. Conditioned on the serve going in, the probability of winning the point is $p_{1,1}$ and $p_{1,2}$ for the first and second serve, respectively. Since first serves tend to be more aggressive, $q_1 \leq q_2$ and $p_{1,1} \geq p_{1,2}$. Assume handicap credits can only be used when serving.

26. *"Scrabble, like life, is a trade-off between today and tomorrow-between spending and saving. It's what an economist would call a dynamic programming problem."* (Roeder [2022]) The game of Scrabble provides an opportunity for applying Markov decision processes. Develop a model for a decision of whether a player should replace some or all tiles during a turn. This is a rather complex model that will be challenging to formulate in its entirety. We do not believe it has been addressed in the Markov decision process literature.

Bibliography

O. Alagoz, L. M. Maillart, A. J. Schaefer, and M. S. Roberts. Determining the acceptance of cadaveric livers using an implicit model of the waiting list. *Operations Research*, 55:24–36, 2007.

K. J. Arrow, D. Blackwell, and M. A. Girshick. Bayes and minimax solutions of sequential decision problems. *Econometrica*, 17:213–244, 1949.

K. J. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica*, 19: 250–272, 1951.

K. J. Arrow, S. Karlin, and H. E. Scarf. *Studies in the Mathematical Theory of Inventory and Production*. Stanford University Press, Stanford, CA, 1958.

D. Bertsimas and R. Shioda. Restaurant revenue management. *Operations Research*, 51:472–486, 2003.

V. Carter and R. E. Machol. Technical note — operations research on football. *Operations Research*, 19:541–544, 1971.

A. Cayley. Mathematical questions with their solutions, no. 4528. *Educational Times*, 23:18, 1875.

T. C. Y. Chan and R. Singal. A Markov Decision Process-based handicap system for tennis. *Journal of Quantitative Analysis in Sports*, 12:179–189, 2016.

T. C. Y. Chan, C. Fernandes, and M. L. Puterman. Points gained in football: Using Markov process-based value functions to assess team performance. *Operations Research*, 69:877–894, 2021.

R.R. Chen, T.C.E. Cheng, et al. Novel advances in applications of the newsvendor model. *Decision Sciences*, 47(1):8–10, 2016.

Y. S. Chow, H. Robbins, and D. Siegmund. *Great Expectations: The theory of optimal stopping*. Houghton-Mifflin, New York, 1971.

C. W. Clark. The lazy adaptable lions: a Markovian model of group foraging. *Animal Behavior*, 35:361–368, 1987.

D. P. de Farias and B. van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51:850–865, 2003.

A. Dvoretsky, J. Kiefer, and J. Wolfowitz. The inventory problem: I. case of known distributions of demand. *Econometrica*, 20:187, 1952.

F.Y. Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 51(1):113–127, 1888.

G. Gallego and G. van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40:999–1020, 1994.

Y. Goçgun and M. L. Puterman. Dynamic scheduling with due dates and time windows: An application to chemotherapy patient appointment booking. *Health Care Management Science*, 17:60–76, 2014.

M. Hall. The state of goalie pulling in the NHL. `https://hockey-graphs.com/2020/05/18/the-state-of-goalie-pulling-in-the-nhl/`, May 2020. [accessed 14-September-2021].

D. P. Heyman. Optimal operating policies for M/G/1 queueing systems. *Operations Research*, 16:362–382, 1968.

R. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

T. Hubel, J. Myatt, N. Jordan, O. Dewhirst, J. McNutt, and A. Wilson. Energy cost and return in african wild dogs and cheetahs. *Nature Communications*, 7:11034, 2016.

S. Karlin. Stochastic models and optimal policies for selling an asset. In K. J. Arrow, S. Karlin, and H. Scarf, editors, *Studies in Applied Probability and Management Science*, pages pp. 148–158. Stanford University Press, Palo Alto, CA, 1962.

E. J. Kelly and P. L. Kennedy. A dynamic stochastic model of mate desertion. *Ecology*, 74:351–366, 1993.

J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand-Reinhold, New York, 1960.

M. Kurt, B. T. Denton, A. J. Schaefer, N. D. Shah, and S. A. Smith. The structure of optimal statin initiation policies for patients with type 2 diabetes. *IIE Transactions on Healthcare Systems Engineering*, 1:49–65, 2011.

M. Mangel and C. W. Clark. Towards a unified foraging theory. *Ecology*, 67:1127–1138, 1986.

D. G. Morrison. On the optimal time to pull the goalie: A poisson model applied to a common strategy used in ice hockey. *TIMS Studies in Management Science*, 4: 67–78, 1976.

P. Naor. On the regulation of queue size by levying tolls. *Econometrica*, 37:15–24, 1969.

J. Patrick, M. L. Puterman, and M. Queyranne. Dynamic multi-priority patient scheduling. *Operations Research*, 56:1507–152, 2008.

E. Porteus. *Foundations of Stochastic Inventory Theory.* Stanford Business Books, 2002.

M. L. Puterman. *Markov Decision Processes:Discrete Stochastic Dynamic Programming.* John Wiley & Sons., 1994.

O. Roeder. *Seven games: ahuman history.* W. W. Norton, New York, 2022.

A. Sauré, J. Patrick, and M. L. Puterman. Optimal multi-appointment scheduling. *European Journal of Operational Research*, 223:573–584, 2012.

S. M. Shechter, M. D. Bailey, A. J. Schaefer, and M. S. Roberts. The optimal time to initiate HIV therapy under ordered health states. *Operations Research*, 56:20–33, 2008.

E. Sirot and C. Bernstein. Time sharing between host searching and food searching in parasitoids: state-dependent optimal strategies. *Behavioral Ecology*, 7:189–194, 1996.

B. C. Smith, J. F. Leimkuhler, and R. M. Darrow. Yield management at american airlines. *Interfaces*, 22:8–31, 1992.

G.L. Smuts. Diet of lions and spotted hyenas assessed from stomach contents. *S. Afr. J. Wildl. Res.*, 9:19–25, 1979.

M. J. Sobel. Optimal average-cost policy for a queue with start-up and shut-down costs. *Operations Research*, 17:145–162, 1969.

D. Stengos and L. C. Thomas. The blast furnaces problem. *European Journal of Operational Research*, 4:330–336, 1980.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An introduction, second edition.* MIT Press, Cambridge, MA, 2018.

K. Talluri and G. van Ryzin. *The Theory and Practice of Revenue Management.* Springer US, 2004.

A. Wald. *Sequential Analysis.* John Wiley & Sons, New York, 1947.

A. Wald and J. Wolfowitz. Optimal character of the sequential probability ratio test. *Ann. of Math.. Stat.*, 19:326–339, 1948.

A. Washburn. Still more on pulling the goalie. *Interfaces*, 21:59–64, 1991.

M. Yadin and P. Naor. On queueing systems with variable service capacities. *Naval Research Logistics Quarterly*, 14:43–53, 1967.

Index

Examples
- Advanced appointment scheduling, 24
- Grid world, 29
- Inventory management, 1
  - Backlogged demand, 3
- Liver transplantation, 21
- Newsvendor, 6
- Online dating, 36
- Optimal parking, 35
- Optimal stopping, 33
- Pull the goalie, 37
- Queuing admission control, 13
- Queuing service rate control, 11
- Revenue management, 8
- Secretary problem, 36
- Selling an asset, 34
- Tennis, 39

Post-decision states, 16