INTELLIGENT ALARMS: ALLOCATING ATTENTION AMONG CONCURRENT PROCESSES

A DISSERTATION

SUBMITTED TO THE PROGRAM IN MEDICAL INFORMATION SCIENCES

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

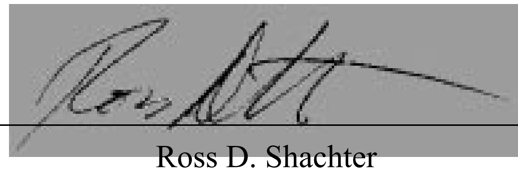
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Cecil Huang

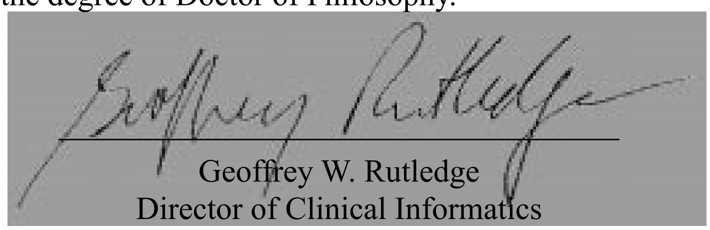
March 1999

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



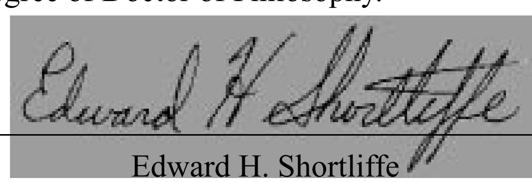
Ross D. Shachter
Department of Engineering-Economic Systems
and Operations Research
Stanford University
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



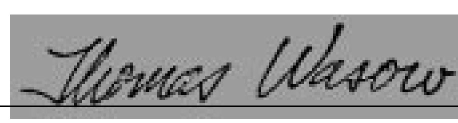
Geoffrey W. Rutledge
Director of Clinical Informatics
Healtheon Corporation

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.



Edward H. Shortliffe
Departments of Medicine and Computer Science
Stanford University

Approved for the University Committee on Graduate Studies



Dean of Graduate Studies

Abstract

I have developed and evaluated a computable, normative framework for intelligent alarms: automated agents that allocate scarce attention resources to concurrent processes in a globally optimal manner. I have been motivated to undertake this research by the inadequacy of current alarms—particularly of those in the intensive-care unit (ICU) setting. Although their underlying approaches allow current alarm systems to detect events of varying complexity, these systems lack an integrated understanding of the competition among processes for attention. Thus, the aims of my research are twofold: (1) to develop a domain-independent method for allocating scarce attention resources among concurrent processes, and (2) to evaluate this method's potential usefulness in a real-world ICU setting.

My approach rests on decision-theoretic foundations, and relies on Markov decision processes to model time-varying, stochastic systems that respond to externally applied actions. I developed a process model that enables partial amounts of attention to be applied over a specified time horizon. I then used separable-optimization techniques to build a framework that allocates attention to a collection of such processes. Given a collection of continuing processes and a specified time horizon, my framework computes two quantitative metrics for each process: (1) an attention allocation, which reflects how much attention the process is awarded, and (2) an activation price, which reflects the process's priority in receiving the allocated attention amount. My framework computes these metrics efficiently—in time proportional to the number of processes being maintained.

I have developed a prototype, SIMON, that computes these alarm signals for a simulated ICU. My validity experiments investigate whether sensible input results in sensible output. The results show that SIMON produces alarm signals that can be explained from the normative basis, and that are consistent with sound clinical judgment. To assess computability, I used SIMON to generate alarm signals for an ICU that contained 144 simulated patients; the entire computation took about 2 seconds on a machine with only moderate processing capabilities. I thus conclude that my alarm framework is valid and computable, and therefore is potentially useful in a real-world ICU setting.

Acknowledgments

I thank Ross Shachter, my principal adviser, for inspiring me over the past 5 years. His mastery of the principles and pragmatics of rational decision making permeated both his academic activities and his everyday decision making. Ross handled uncertain situations in a calm and unbiased manner, and he consistently provided cogent analyses and advice.

I am grateful to Geoff Rutledge for his indispensable contributions. Geoff provided a unique perspective that came from his extensive experiences as a physician and as a medical informatician. His involvement made my evaluation possible.

I am delighted to acknowledge Ted Shortliffe for shaping both my research interests and my professional development as a physician–scientist. He believed in my abilities, broadened my vision, and provided me the necessary resources to pursue my goals. Ted has also enriched me indirectly through his distinguished career, and through the Stanford Medical Informatics (SMI) training environment, which he spearheaded.

Paul Dagum and Adnan Darwiche each deserve special recognition. For 2 years, Paul took time from his normal activities as a surgical housestaff and a research computer scientist to meet with me regularly. He provided a unique camraderie and many a morale boost. Adnan worked closely with me on probabilistic-inference algorithms for 1.5 years, and on what would eventually become my first journal paper.

Several decision-theoretic researchers made key contributions. Adam Seiver helped me appreciate the many problems that impede the effective delivery of critical care. Eric

Horvitz's work on time-critical decisions helped springboard my technical innovations. Ron Howard's 1960 monograph became the basis for my work's innermost workings.

Kurt Huang has been a friend and an intelligent sounding board. This dissertation has also benefited from earlier interactions with Russ Altman, Tom Chavéz, Larry Fagan, David Gaba, David Heckerman, Rebecca Lee, Ramani Pichumani, Yuval Shahar, and Jay Strain.

Several SMI folk smoothed my ride through the program. Joe Norman, Smadar Shiffman, and Maria Tovar encouraged me during our sporadic encounters. John Egar, Glenn Rennels, and Mike Walker shared their own graduate experiences. Mark Musen and Darlene Vian kept my bearings straight while I made crucial choices about my research direction.

Like many of my SMI predecessors, I thank Lyn Dupré for editing this entire manuscript—and for teaching me how to write more clearly in the process.

I must also acknowledge the friendship of a few significant housemates. Raymond Shen was my companion during my most difficult moments, and he helped me to learn a few things about optimization. Enoch Huang inspired me with his ability to complete difficult tasks methodically and efficiently, and with his unsurpassed facility with the MacOS platform. Frank Wu has been my bridge to humanity and reality during the final phases.

Financially, I have been supported mostly by the Medical Scientist Training Program under grant 5T32GM07365 from the National Institute of General Medical Sciences. I also received support from the National Science Foundation grant IRI-93-11950, the Stanford University CAMIS project grant IP41LM05305 from the National Library of Medicine of the National Institutes of Health, and the Rockwell International Science Centers in Palo Alto and Thousand Oaks, California.

I am indebted to my parents and siblings for their unconditional love and support. I dedicate this dissertation to Mom and Dad, for their vicarious participation in my doctoral journey. Finally, I thank God for orchestrating this journey and for making it meaningful.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this dissertation, I develop a normative framework for intelligent alarms. I am motivated by the inadequacy of real-world alarms—particularly of those in the intensive-care unit (ICU) setting. *My hypothesis is that we can compute, in real time, normatively sound alarm signals that allocate the attention of busy agents among a set of concurrent processes that they are managing.*

1.1 The alarm task

People rely on alarms in many real-world scenarios. A watch alarm is set to chime at a preset time; a fire alarm is triggered when the density of ambient smoke particles exceeds a certain threshold; a pager vibrates when its owner's attention is being requested. Alarms help seismic scientists detect earthquakes; they help industrial engineers control manufacturing-plant processes; they help ICU clinicians manage the health of critically ill patients.

From these and other examples, we can make a few observations about alarms in general. An **alarm** is an automated agent[1] that monitors one or more time-varying quantities, and that issues alarm signals, or **alerts,** in response to these monitored inputs (Figure 1.1). The alarm employs a reasoning mechanism that transforms the time-varying inputs into alerts. Alerts convey meaningful summaries of the inputs, and are thus informative. Alerts also convey a sense of urgency: They draw attention to events that may be important to consider now, rather than later. Hence, alerts are also necessarily intrusive.

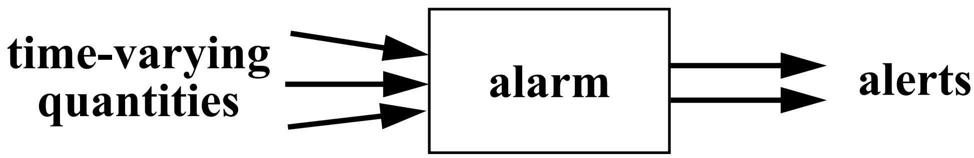


Figure 1.1. Generic alarm.
An alarm is a device that generates alarm signals, or alerts, in response to a set of time-varying input quantities.

1.1.1 Alarms as monitoring assistants

To have a conversation about how alarms *should* behave, we must examine the monitoring tasks that alarms are intended to support. Figure 1.2a depicts a generic monitoring task. An agent[2] is responsible for managing an ongoing process. She observes the process over time, performing actions on the process as the observations indicate.

Figure 1.2b depicts the same monitoring scenario, but with an alarm interposed between the process and the agent. The alarm observes the process over time and issues an alert whenever the process needs an action that requires the agent's presence. In this manner, the alarm untethers the agent from the process: She is free to perform other tasks, as long as she can respond to alerts in a timely, effective manner.

---

1. For convenience, I refer to alarms as devices in this dissertation. However, an alarm may be a *human* agent who faithfully carries out an alert protocol; for example, an inventory-control clerk reports shortages of various stocked items to the purchasing manager, who then acts on such information.

2. For the remainder of the dissertation, I use *agent* to refer to the decision maker who manages the concurrent processes.

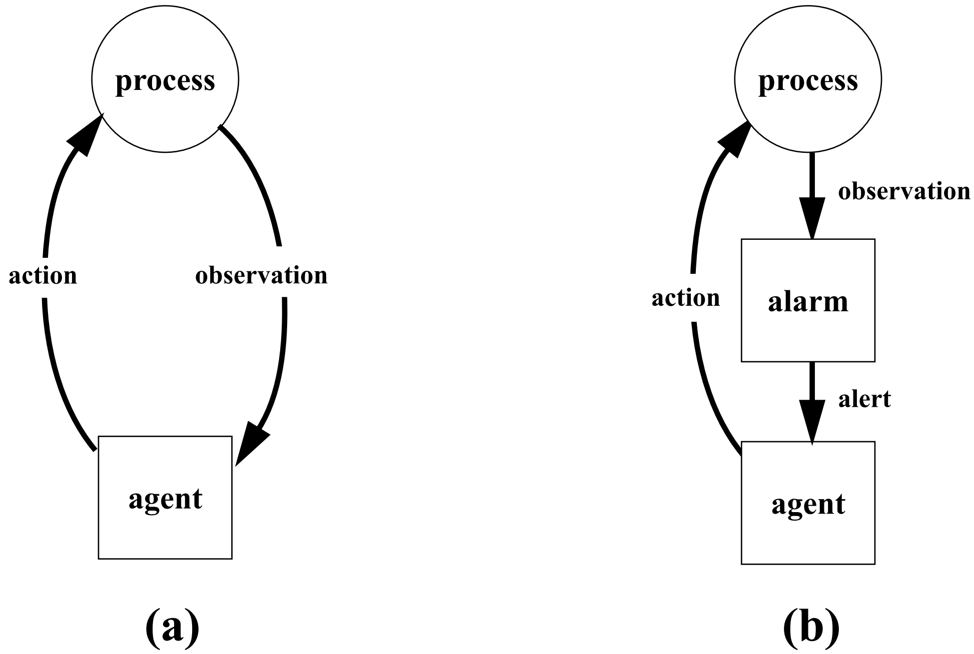


Figure 1.2. Generic monitoring task.
An agent is responsible for managing an ongoing process. In (a), the agent attends to the process at all times. In (b), an alarm attends to the process continuously, alerting the agent whenever the process needs the agent's active intervention.

By acknowledging the role of alarms as monitoring assistants, we immediately conclude that an intelligent alarm is **action based:** It generates alerts based on a process's need for an agent's attention.

1.1.2 Alarms as multitasking assistants

An agent manages a process with the help of an alarm because she is unable (or unwilling) to devote all her attention to that process. In general, the agent manages several concurrent processes to which she cannot attend simultaneously. Figure 1.3 illustrates how a busy agent may divide her attention among a set of three processes: At any given moment, she attends to one process, while leaving the other two processes unattended.

Alarms can help our agent to multitask among the processes that she manages. Suppose that each process is monitored by an action-based alarm. Then this system of alarms can issue alerts according to the attention needs of particular processes. Problems arise,

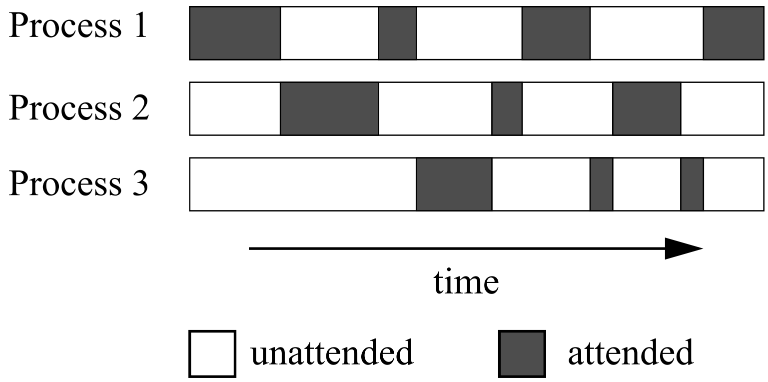


Figure 1.3. Monitoring multiple processes under attention constraints. An agent multitasks among three processes. Over the interval of time shown, the agent switches her attention among the processes, in an effort to manage optimally the collection of processes.

however, whenever there are simultaneous alerts that each request the agent's attention. Despite the agent's best intentions, she must attend to only one process, while temporarily forgoing the others. This problem of managing competing alerts illustrates the need for alerts that communicate not only a process's need for attention, but also the **urgency** of that need. Such alerts would enable the agent to attend to the most urgent process, while maintaining an awareness of the imminent attention needs of the other processes.

1.2 Real-world alarms

Despite their ubiquitous importance, alarms in the real world often fall short of the ideal. They can detect conditions of varying complexity, but they do not issue alerts that necessarily reflect the need for an agent's intervention. Furthermore, the intrusiveness of these alerts does not necessarily reflect the urgencies of the corresponding situations.

1.2.1 Current alarm approaches

I have been motivated, in particular, by the inadequacies of current ICU alarms. The most prevalent ICU alarms monitor a set of phyiologic parameters over time; an alert is issued whenever any of the parameters fall outside its normal range. Unfortunately, this simple alert strategy has been shown to be ineffective: False-alert[3] rates as high as 93 percent have been reported in various studies [Deller et al., 1992; Kestin et al., 1988; Meijler, 1987]. False alerts are problematic because they can divert the attention of clinicians away from more important tasks. Furthermore, they are a source of irritation, often causing clinicians to switch the alarms off—a phenomenon known as the cry wolf syndrome [Sykes, 1989]. Other examples of nonideal alerts are false-negative alerts and true-positive alerts with inappropriate delays [Páte-Cornell, 1986]. In general, nonideal alerts impede the efforts of ICU clinicians to multitask optimally among their patients.

Researchers have developed numerous techniques for addressing the inadequacies of range-checking alarm approaches. These enhanced alarm approaches range from filtering algorithms that remove spurious transients from individual signals, to approaches that perform higher level inferences by integrating the information obtained from multiple signals. (I survey current alarm approaches in Chapter 2.) Although such approaches allow current alarm systems to detect events of varying complexity, these systems lack an integrated understanding of the *competition* among processes for attention. For example, consider an ICU clinician who is confronted by a set of simultaneous alerts that originate independently from several different patients (Figure 1.4). The alerts may draw attention to potentially interesting events concerning the patients, but the clinician must still decide how to act, given the competing alerts.

1.2.2 Normative alarm approaches

Quinn [1989] envisioned an alarm system that organizes and displays alerts in a linear hierarchy, according to each alert's relative need for attention. A clinician using such an alarm system would service the alert at the top of the queue, while maintaining an awareness of other, pending alerts. My thesis research was inspired by this vision of an intelligent alarm. In pursuing this vision, however, I discovered that none of the existing alarm

3. *False alerts* are those that are retrospectively judged to provide no added value to the care of the patient.

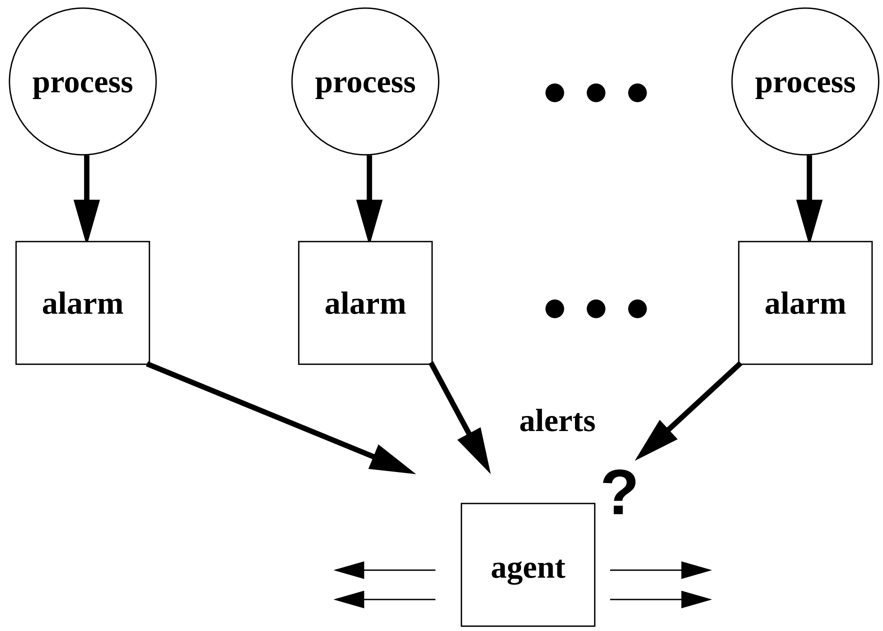


Figure 1.4. The problem of multiple, independent alerts.
An agent is confronted by simultaneous alerts that originate independently from multiple processes. Although each alert may result from a sophisticated inference procedure, the alerts, taken together, do not help the agent to allocate her attention among the processes.

approaches explicitly allocated scarce attention resources among concurrent processes. Thus, my research aims were twofold: (1) to develop a domain-independent framework for the attention-allocation problem, based on normatively sound reasoning principles; and (2) to evaluate this method's potential usefulness in a real-world ICU setting.

1.3 SIMON: A prototype, normative alarm system

I developed a program, SIMON, that generates decision-theoretic alarm signals for a ward of simulated ICU patients who are being ventilated mechanically. SIMON's patient models are based, in part, on the outputs of a validated prototype ventilator-management advisor.

1.3.1 Clinical scenario

Consider the following scenario of an ICU with three simulated patients who have been admitted for separate, potentially life-threatening illnesses (Table 1.1).[4] Each patient receives ongoing life support, which enables his body a chance to heal itself. The physiologic status of each patient is monitored through telemetry; changes in physiology may warrant a change in the standing treatment protocol. Any change in treatment, however, must be administered by the ICU clinician, Dr. Seiver, who physically attends to each patient at the bedside.

ID #	Name	Condition
1	Mr. Dickinson	congestive heart failure
2	Ms. Di'Anno	pulmonary embolism
3	Mr. Bayley	pneumonia

Table 1.1. Three patients and their medical conditions.

Despite his best intentions, Dr. Seiver can attend to only one patient at a time. Thus, he relies on alarm systems that monitor each patient's telemetry, to help him determine where his attention is needed. Under the old alarm system, harsh, audible alerts originated independently and frequently from each bedside, alerting Dr. Seiver to conditions ranging from a loosely attached oxygen-saturation monitor (usually not an emergency), to a precipitous, unexplained drop in cardiac output (usually an emergency). Regardless of an alert's actual urgency, Dr. Seiver was obligated to service it immediately; in most cases, the urgency arose more from a desire to stop the noise and return to his original task, than from the cause of the alert.

4. I describe the models for these patients in Section 6.4.

1.3.2 Alarm signals computed by SIMON

Now, Dr. Seiver uses a new alarm system, SIMON, that helps him to set priorities for allocating his limited attention among the three patients. SIMON integrates physiologic data obtained from *all* the patients, and summarizes the alert status of each patient with two numerical measures:

- A **priority measure,** indicating the *order* in which the patient needs attention
- An **attention allocation,** indicating *how much* attention the patient needs

These alarm signals are then displayed on a portable, integrated monitor typified by Figure 1.5, with the higher priority patients listed farther up. Note that, in addition to the physiologic measurements, two input values affect the computed alarm metrics:

- The **available attention:** the effective number of available ICU clinicians
- The **attention horizon:** the duration over which the available attention is to be allocated

For the scenario illustrated in Figure 1.5, there is 1 available attention unit—because Dr. Seiver is working alone[5]—and the attention horizon is set to 6 minutes. Upon consulting the monitor, Dr. Seiver decides to attend to patient 2, to whom he expects to devote 76 percent of his attention over the next 6 minutes. During this same time interval, Dr. Seiver also expects to devote 16 percent of his attention to patient 1, and 8 percent of his attention to patient 3.

A change in either input would alter SIMON's output. For example, suppose that Dr. Seiver wants to know how he should allocate his attention over the next 60 minutes. Figure 1.6 illustrates the results of changing the attention horizon from 6 minutes to 60 minutes. Dr. Seiver would still first attend to patient 2, but his background awareness of the other patients would increase: Over the next 60 minutes, he would now expect to devote 26 percent of his attention to patient 1, and 24 percent of his attention to patient 3.

5. The available attention would increase if Dr. Seiver recruits additional clinicians.

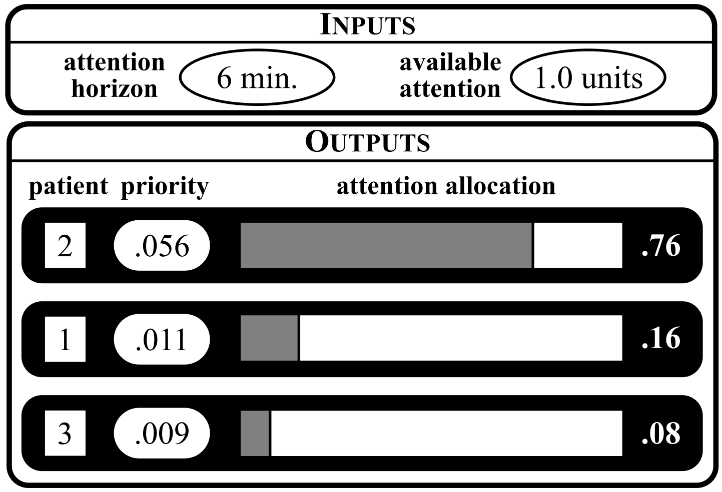


Figure 1.5. Displayed alarm signals for three patients (T = 6 minutes). Displayed are SIMON's alarm signals for the patients of Table 1.1, given an attention horizon $T = 6$ minutes and 1 available attention unit. The priorities are displayed in microutils per attention unit, where a util denotes the probability of a patient's survival. The attention allocations are a recommended partitioning of the available attention units. In this example, higher-priority patients have higher attention allocations; this correspondence does not hold in general.

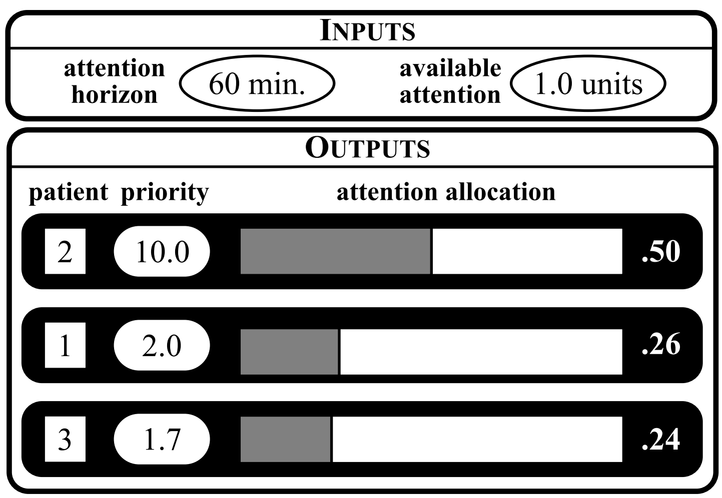


Figure 1.6. Displayed alarm signals for three patients (T = 60 minutes). Displayed are SIMON's alarm signals for the patients of Table 1.1, given an attention horizon $T = 60$ minutes and 1 available attention unit. The values differ from those of Figure 1.5: The priority measures increase with T, but the relative priorities remain the same. Also, with increased T, the attention allocations become closer to one another.

1.3.3 Time-varying alarm signals

SIMON's alarm signals are time-varying, in general, because they are based on inputs that may vary over time. Changes in input can occur in several ways. A patient may experience a change in physiology; he may undergo a change in treatment; his illness may evolve from one type to another. He may be discharged from the ICU: His condition may stabilize to the point that he no longer needs ICU treatment, or he may die despite his receiving aggressive life support. The ICU could admit an additional patient; the ICU could gain or lose clinician resources. In general, one or more of these changes would change the optimal attention allocation, and would thus cause SIMON's output to change.

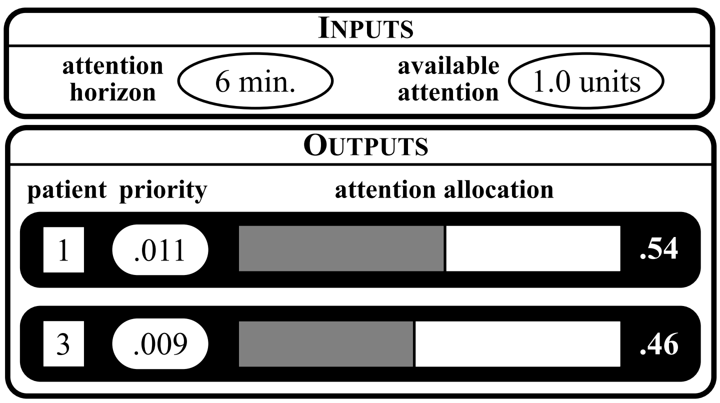


Figure 1.7. Displayed alarm signals for two patients (T = 6 minutes).
Displayed are SIMON's alarm signals for an ICU that contains only patients 1 and 3 (see Table 1.1), given an attention horizon $T = 6$ minutes and 1 available attention unit.

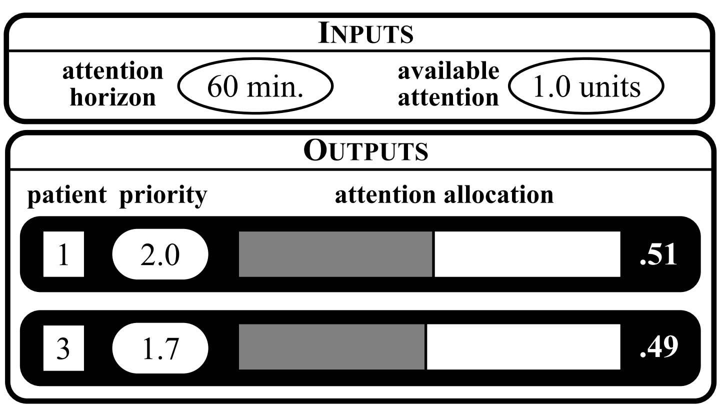


Figure 1.8. Displayed alarm signals for two patients (T = 60 minutes).
Displayed are SIMON's alarm signals for an ICU that contains only patients 1 and 3 (see Table 1.1), given an attention horizon $T = 60$ minutes and 1 available attention unit.

To witness an update in SIMON's output, let us return to Dr. Seiver's ICU scenario. Recall that the alarm signals for $T = 6$ minutes and $T = 60$ minutes are shown in Figures 1.5 and 1.6, respectively. Suppose that Ms. Di'Anno (patient 2) is successfully treated and discharged. Dr. Seiver receives an alert on his portable monitor; upon checking the monitor, he notes that only two patients remain in the ICU. With fewer patients, Dr. Seiver is able to devote more attention to each of them than previously possible. In other words, the attention that had been allocated to Ms. Di'Anno is now reallocated to Mr. Dickinson and Mr. Bayley. Figures 1.7 and 1.8 depict the updated alarm signals for attention horizons of 6 minutes and 60 minutes, respectively.

SIMON has additional capabilities, and can produce interesting behaviors, beyond those that I have demonstrated here. I discuss SIMON's architecture in Chapter 5 and characterize its behavior in Chapter 6.

1.4 SIMON's underlying approach

SIMON's outputs originate from the alarm approach that I develop in this dissertation. Figure 1.9 illustrates a schematic of my alarm approach. Each process has a corresponding model that interprets the measurements obtained from it. The process models, in turn, are considered together by an overarching framework. Given an available attention amount and an attention horizon, this framework computes a priority measure and an attention allocation for each process. Although I use SIMON to simulate various ICU scenarios, the underlying approach is domain independent: ICU clinicians are examples of agents, and patients are examples of processes.

Note that my framework produces *two* quantitative alarm measures for each process, in contrast to the linear alert hierarchy suggested by Quinn [1989]. My research demonstrates, in fact, that the general notion of urgency *cannot* be captured by a single quantitative measure. In Section 2.5.3, I review prior work on a one-dimensional urgency measure that has been developed for a specific class of processes.

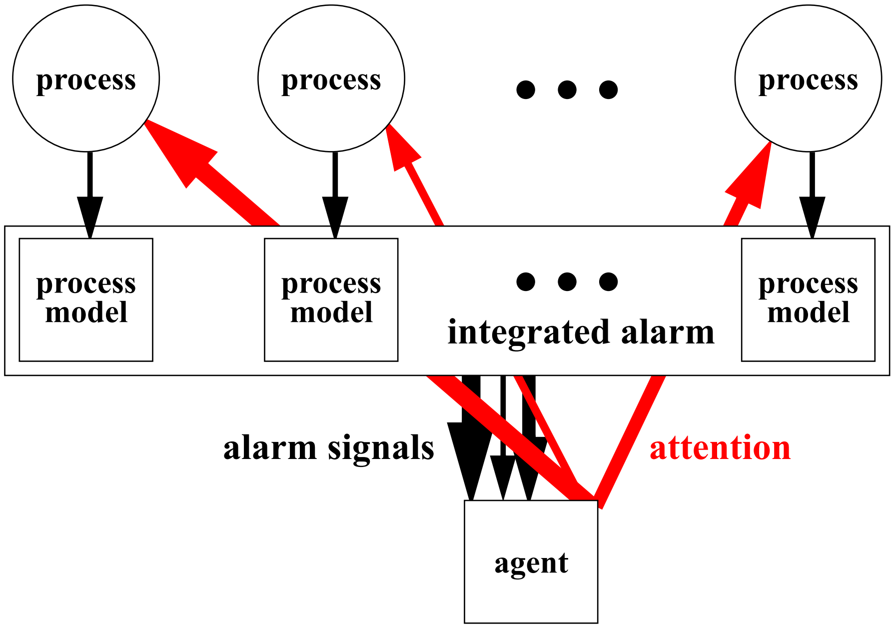


Figure 1.9. Schematic of SIMON's underlying approach.
Observations from each process are interpreted by a corresponding model. An integrated alarm considers the collection of processes taken as a whole. The integrated alarm computes, for each process, a pair of alarm signals; each pair of alarm signals is represented as a black arrow. The agent considers the set of alarm signals and allocates her attention among the processes, as indicated by the grey arrows.

1.5 Guide for the reader

In Chapter 2, I present a survey and an analysis of methodologies that have been developed to address the alarm problem. Building on decision-theoretic foundations, I develop my alarm approach in Chapters 3 and 4. In Chapter 5, I describe SIMON, a program that uses my approach to generate alarm signals for a simulated ICU. In Chapter 6, I describe how I used SIMON to test my research hypothesis. Finally, in Chapter 7, I reflect on my work's contributions to various disciplines, and point to directions for future research.

Chapter 2

Alarm Approaches

The intelligence of an alarm system depends on the underlying reasoning, or methodology, that it uses to generate alerts. Previous approaches to intelligent alarms have relied on methods that vary in sophistication. In this chapter, I survey these alarm approaches. I provide examples from medicine and other domains, and I place my thesis research in the context of this previous work.

The majority of alarm approaches can be subdivided into groups that are characterized by a small set of methodologic distinctions. My exposition is guided, although not restricted, by the hierarchy of distinctions shown in Figure 2.1. For instance, when I discuss diagnostic alerts, I include both static and temporal examples.

2.1 Input-signal processing

Alarm approaches can be distinguished broadly by how they handle multiple, simultaneous signals acquired from a single process. In **observation-based approaches,** each signal is processed independently; alerts are based on features recognized in the individual signals (Figure 2.2a). **Event-based approaches,** on the other hand, process observations

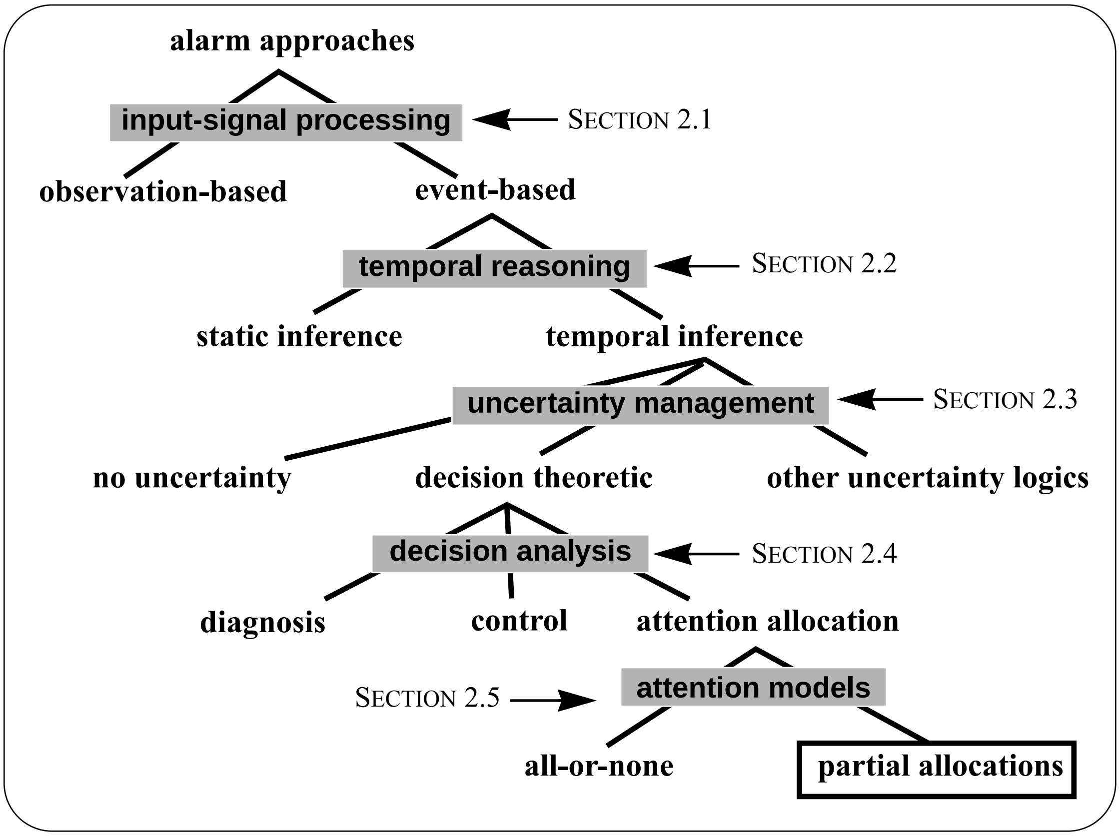


Figure 2.1. Methodologic distinctions.
I survey alarm approaches according to this hierarchy of distinctions. The alarm approach developed in this dissertation is indicated by the bold-faced rectangle.

from multiple sources in an integrated manner; alerts are based on features identified in the signals, which are taken together (Figure 2.2b).

In the following subsections, I investigate various observation-based approaches to generating alerts. I describe event-based approaches in Section 2.2.

2.1.1 Simple range checking

The simplest, and most widely implemented alarm models are based on thresholds of measured signals. An inventory-control alarm triggers when the number of stocked items falls below a predetermined threshold; a smoke alarm is activated when the density of smoke

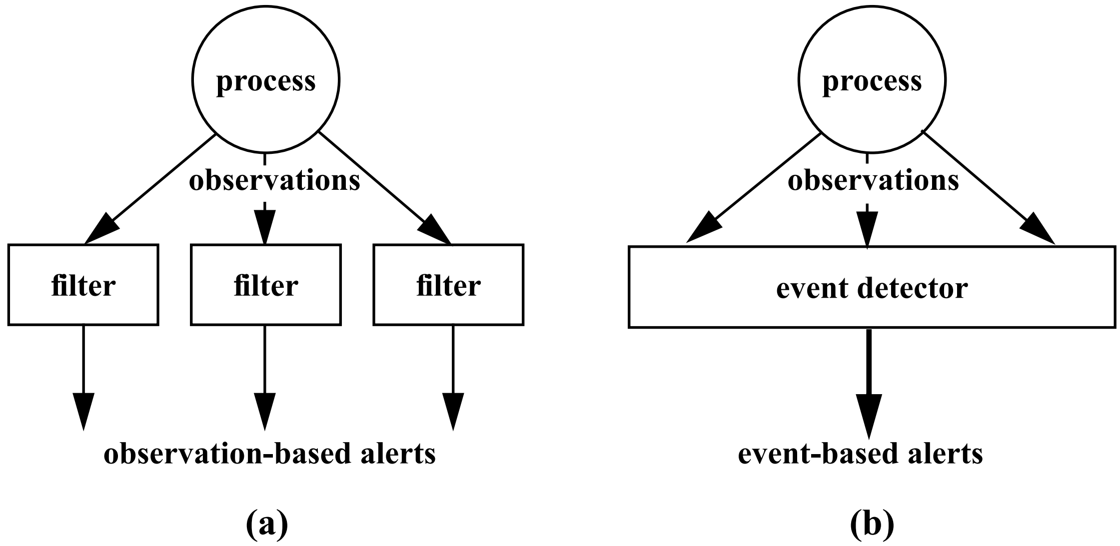


Figure 2.2. Observation-based versus event-based alarm approaches. Observation-based (a) and event-based (b) alarm approaches differ in how they handle multiple observations from the same source. Observation-based alerts use features present in the individual observations, whereas event-based alerts use process characteristics inferred from the observations considered together.

particles exceeds a predetermined threshold; an intensive-care unit (ICU) alarm sounds when one or more physiologic parameters fall outside their normal ranges.

Although sufficient for some domains, the simple-threshold alarm approach has been shown to be ineffective in many others. For example, false alerts that occur in the monitoring of critically ill patients are well documented—false-alert rates of 33, 75, and 93 percent have been reported in various critical-care settings [Deller et al., 1992; Kestin et al., 1988; Meijler, 1987]. It is possible to increase a simple-threshold alarm's specificity (the probability of an alert condition, given an alert) by adjusting the thresholds to broaden the normal range; however, such an adjustment also decreases the alarm's sensitivity (the probability of an alert, given an alert condition). Clinicians typically tolerate a false-alert rate that balances these opposing effects.

2.1.2 Intelligent range checking

To improve the performance of simple-threshold alarms, researchers have employed methods that address the various sources of false alerts.

2.1.2.1 Real-time filters for noise and transients

The signals that are obtained from ICU patients or other physical systems often contain noise and transients that obscure the underlying quantities of interest. For example, momentary data spikes often originate from artifacts in the measurement process. Also, parameter values may drift for reasons that do not necessarily indicate an alert. Restless patients may exhibit erratic variations in their physiologic measurements; patients with complex illnesses such as multiple-organ failure may exhibit characteristic, low-frequency hemodynamic oscillations [Dagum et al., 1995].

Median or low-pass filters can be used to reduce noise and transients whenever data are sampled frequently and inexpensively. Although such filtering may be useful in critical-care settings, the resulting loss of information makes the technique less appropriate for domains that are characterized by relatively sparse data, such as monitoring of children's growth [Haimowitz, 1994].

2.1.2.2 Custom-tailored thresholds for heterogeneous patients

A generic set of normal parmeter ranges may not apply to any specific patient. In fact, critically ill patients often have "normal ranges" that are abnormal for the healthy patient population. For example, we would expect an elderly man who has a longstanding history of cardiovascular disease to have lower cardiac output and higher arterial blood pressure, compared to a young, healthy woman with no cardiovascular disease.

To address the diverse monitoring needs of patients, several researchers developed methods that assign custom-tailored thresholds for specific categories of patients [Beneken & van der Aa, 1989]. Moret-Bonillo and associates [1992] used a symbolic rule set to assign ventilator-assisted patients into categories, each associated with a predetermined set of lower and upper thresholds. van Oostrom and colleagues [1993] employed a statistical clustering approach to perform a similar classification task for patients undergoing intra-operative anesthesia.

2.1.2.3 Thresholds that incorporate temporal risk factors

Many other real-world phenomena confound the search for optimum thresholds. For example, true alerts that are issued too late may result in losses that could have been avoided with earlier warning. For example, a cardiac-arrest alert is a true alert, but of more value are alerts for cardiac rhythm disturbances that, if untreated, will lead to cardiac arrest. Also, false alerts can have consequences beyond the immediate cost of a needless response: Individuals who have responded to numerous false alerts may discount or even ignore future alerts (the cry-wolf phenomenon).

Páte-Cornell [1986] considered the effects of delayed warnings, the cry-wolf phenomenon, and other real-world factors in developing a probabilistic model for a threshold-based warning system. Her model focuses primarily not on alert sensitivities and specificities, but rather on the costs and benefits associated with an alert; the optimal threshold maximizes an aggregate measure of these costs and benefits.

2.2 Temporal reasoning

Compared to their observation-based counterparts, event-based alarm approaches typically involve more-complex reasoning, and they generally elucidate more about the underlying process. Event-based approaches are thus often described as *intelligent.* Intelligent alarm approaches vary widely, however, and can be differentiated by how the underlying models handle the notion of time. This difference is illustrated in Figure 2.3 and is discussed next.

2.2.1 Static inference

Models that perform **static inference** do not consider time explicitly.[1] Static models interpret observations via mechanisms that do not distinguish between measurements or events

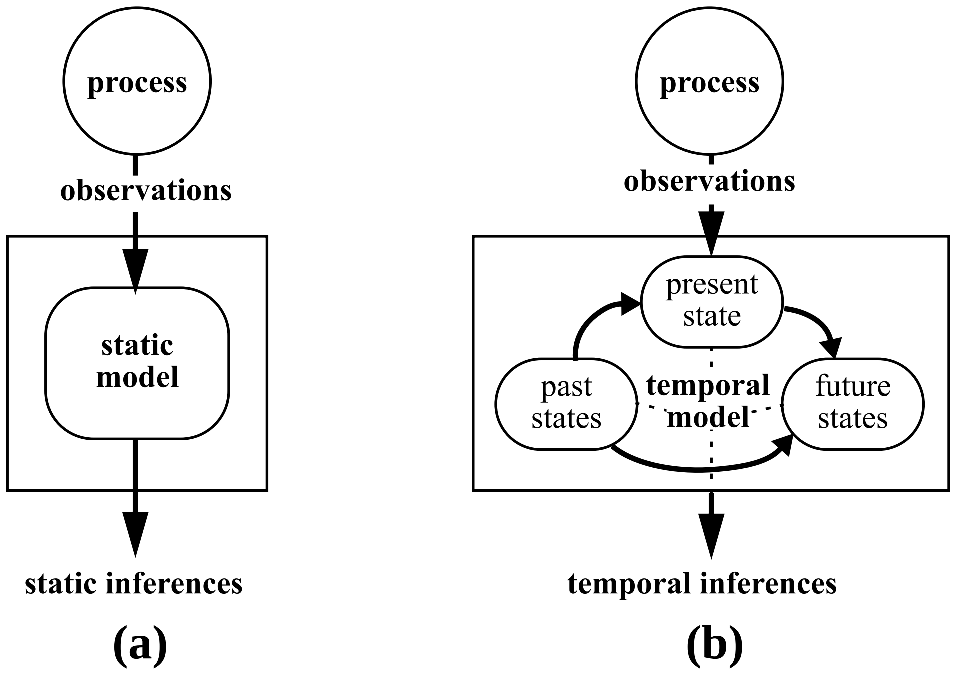


Figure 2.3. Static versus temporal models.

Static and temporal models differ in how they interpret observations from a process at a given time. (a) Static models interpret a constellation of observations by determining which present, underlying conditions best explain those observations. (b) Temporal models, in contrast, may consider past or future states of the world, or both. Whereas static models can be used to represent notions of time, temporal models are distinguished by their underlying formalisms, which are designed to deal with time explicitly.

that occur at different times. Note that this characterization does not imply that static models cannot represent temporal concepts. For example, a prediction model may have a variable labeled "death in 5 years." What makes this model static is its inability to reason about death that occurs within any time other than five years.

Researchers have used myriad static modeling formalisms to prototype intelligent medical alarms. van der Aa [1990] used a rule-based approach to develop an expert alarm system that diagnosed faults in the anesthesia breathing circuit, a network of conduits that enables ventilated patients who are undergoing surgery to rebreathe oxygen and anesthetic gases that they have exhaled. Anesthesia alarms based on neural networks [Orr & Westenskow,

1. Other phrases used to describe static models include *domain models, time-invariant models, atemporal models,* and *memoryless models.*

1994] and belief networks [Beinlich & Gaba, 1989] also were prototyped. Hayes-Roth and colleagues [1992] employed a blackboard architecture that supported heterogenous modes of reasoning. Other intelligent approaches to patient monitoring are reviewed in various sources [see, e.g., Gravenstein et al., 1983; Gravenstein et al., 1987; Uckun, 1994].

In addition to using them to assist in the acute monitoring of bedridden patients, researchers have also used static alarm models to infer various pathological conditions from medical records or other databases. Alerts from these sytems are typically presented as reminders that help physicians to cope with ever-increasing amounts of information [McDonald, 1976]. Alert rules have been used in medical information systems for drug-interaction monitoring [Speedie et al., 1987] and laboratory-test interpretation [Connelly et al., 1996], and in the HELP hospital information system [Haug et al., 1994].

Applications of static alarm models also include decision-support systems that make explicit predictions; these predictions, in turn, help experts to focus limited attention resources. A classic example is the APACHE prognostic scoring system, which predicts the survival of ICU patients [Knaus et al., 1991]. APACHE and its successors, APACHE II and APACHE III, are based on statistical prediction models that are constructed from large databases of cases. Other scoring systems designed to allocate limited resources effectively are reviewed by Esserman and associates [1995]. Note that these statistical models are essentially static, despite their ability to predict the future. For example, APACHE III can produce risk estimates for patients who have different lengths of stay in the ICU, but, in doing so, it does not consider temporal relationships among these different risk estimates.

2.2.2 Temporal inference

Models that perform **temporal inference,** or dynamic models, are distinguished from static models by their explicit representation and manipulation of time. Through various representations, temporal models capture inferential or mechanistic relationships between the values of process parameters at different times. Examples of temporal inference include the formulation of a diagnosis based on temporal patterns of observations [Shahar, 1994; Haimowitz, 1994], and the prediction of future process values from past observations [West & Harrison, 1989; Weigend & Gershenfeld, 1993].

An intelligent alarm can benefit in several ways from the ability to reason about time. Many monitoring applications emphasize the importance of detecting *changes* in relevant parameter values; such strategies go beyond declaring particular situations normal or abnormal [Philip, 1989]. In addition, many diagnostic alarms depend on their ability to interpret multivariate temporal trends [Haimowitz, 1994; Shahar, 1994]. Finally, many alarms depend on early warnings that allow respondents to avert the consequences of impending disasters [Páte-Cornell, 1986; Horvitz & Seiver, 1997]. For example, warnings issued by a fire alarm should allow enough time for building occupants to evacuate, and for firefighters to keep property damage in check.

The simplest temporal-inference methods are single-variable trend detectors. These methods can enhance the alert capabilities of range-checking methods by detecting pathologic changes that occur within the normal ranges. For example, a falling mean arterial pressure (MAP) may indicate an internal bleeding process, which commences before the MAP drops below the normal range. Unfortunately, these enhanced detection capabilities come at the expense of additional false alerts—for example, those that result from artifactual measurement spikes or other physiologically meaningless trends. Certain such artifacts can be eliminated with the same filtering techniques that augment range-checking alarms. Single-parameter filters, however, can improve the specificity of trend-detecting alarms to only a certain extent; more specific alerts require the coordinated analysis of multiple parameter values and trends. Avent and Charlton [1990] review biomedical applications of trend detection. Shahar [1994] and Haimowitz [1994] describe novel, artificial-intelligence (AI) approaches to temporal inference.

Models that perform temporal inference do not need to rely on formalisms that are distinct from those used for static inference. Fagan [1980] used data-directed rule sets to represent temporal relations in the design of VM, one of the earliest ICU-monitoring prototypes. Fukui and Masuzawa [1989] describe another ICU-monitoring system that uses a rule-based approach to temporal inference. I discuss temporal extensions of belief networks and decision models in Sections 2.4.1 and 2.4.2.

The alarm methodology presented in this dissertation builds on the Markov decision process (MDP): a temporal decision model with decision-theoretic foundations. I further characterize this methodology in Sections 2.4 and 2.5. But first, let us ponder the problem of reasoning under uncertainty, and my reasons for choosing the decision-theoretic basis over other uncertain-reasoning formalisms.

2.3 Uncertainty management

Probability and statistics have been studied in centuries past, but uncertain reasoning did not receive attention from the AI community until the late 1960s and early 1970s. Before that time, AI was largely regarded as symbolic reasoning built on a foundation of logic; *intelligence* meant the ability to manipulate concepts, rather than to crunch numbers. Enthusiasm for AI grew as symbolic-reasoning methods found their way from artificial block worlds to more complex real-world domains such as medicine and oil wildcatting. One of the first expert systems to incorporate an uncertain-reasoning mechanism was MYCIN, an antibiotic-therapy advisor developed by Shortliffe and colleagues during the early 1970s [Buchanan & Shortliffe, 1984]. Shortliffe and other researchers recognized that AI had to address uncertainties that were inherent in real-world reasoning tasks, if AI was to model those tasks with fidelity.

2.3.1 Uncertainty in the real world

Uncertainty abounds. We monitor a physical system to learn the values of various quantities about which we are uncertain. Often, however, our measurements serve only to reduce, rather than to eliminate, such uncertainties; the quantities in which we are interested express themselves only in part through the quantities that we can observe. Even when we believe that our understanding of a particular process is accurate, we cannot always predict that process's future evolution with absolute certainty. Unmodeled, exogenous influences, called **stochastic influences,** can confound our most detailed prediction algorithms [West & Harrison, 1989; Dagum et al., 1995]; even when a natural system is largely deterministic, infinitesimal perturbations conspire to make the system virtually unpredictable beyond a limited time horizon [Gleick, 1987].

It is in the face of such uncertainties that we deliberate and act. When we must choose among alternatives, we may ponder and evaluate the future consequences of each alternative in the context of our present understandings. Such decisions would be straightforward if we could asscociate a predictable, certain outcome with each alternative, and if we had a way of assessing the relative desirability of different outcomes. In reality, the uncertainty in prospective outcomes forces us to resort to some form of approximate reasoning.

2.3.2 Uncertain reasoning in AI

The simplest way to deal with uncertainty is to abstract it away. We can reason solely from heuristics expressed in binary logic, provided that we have a means of transforming real-world observations and concerns into rules and queries that are composed of only true or false assertions. Rule-based expert systems use such an approach [Buchanan & Shortliffe, 1984]. The use of expert heuristics can assist powerfully in our endeavors to reason about a complex, deterministic domain; however, as Tversky and Kahneman [1974; 1981] demonstrated through psychological studies, people making even simple decisions, in the face of uncertainty, are prone to errors in reasoning if they adhere rigidly to commonly held rules of thumb. Thus, expert systems can fail if, when applied to situations characterized by uncertainty, they rely on the heuristics that people use.

The alternative to abstracting away uncertainty is to deal with it explicitly. The premise here is that uncertainty, when it cannot be eliminated, can and should be quantified. An uncertain-reasoning formalism provides a structured language that we can use to represent varying degrees of belief or confidence, and a set of computation rules that dictates how we should combine and otherwise manipulate such uncertain propositions. Early examples of uncertain-reasoning formalisms in AI are certainty factors [Shortliffe & Buchanan, 1975] and fuzzy logic [Zadeh, 1983; Becker et al., 1997]; there are many others [Shafer & Pearl, 1990].

2.3.3 The emergence of decision theory

Despite its early roots and routine use in engineering applications, decision theory[2] [Bayes, 1958[3]; von Neumann & Morgenstern, 1947; Savage, 1954] was not widely embraced as a viable uncertain-reasoning formalism by the AI community before the mid-1980s. Early demonstrations of decision-theoretic expert systems in medicine [Gorry & Barnett, 1968; de Dombal et al., 1972] did not deter the widespread belief that the required computations could not scale up to large, complex domains without the introduction of grossly unrealistic assumptions of independence. Decision theory was thus judged by many researchers in the 1970s and early 1980s as a number-crunching formalism of theoretical interest, impractical for the "intelligent" reasoning tasks being attacked by AI techniques [Rich, 1983; Horvitz et al., 1988].

Advances in computation and a renewed critique of uncertain-reasoning formalisms brought decision theory into the AI spotlight. Fundamental inconsistencies and other weaknesses in non–decision-theoretic formalisms were elucidated [Heckerman, 1986; Horvitz & Heckerman, 1986; Heckerman & Shortliffe, 1992]. The developments of belief networks and influence diagrams enabled the structuring of domain knowledge [Jensen, 1996; Pearl, 1988; Shachter, 1988; Howard & Matheson, 1981]; researchers invented techniques to perform efficient inference within such structures [Huang & Darwiche, 1996; Dagum & Luby, 1997]. Sequential decision models, already well established in the engineering literature [Howard, 1960; Howard, 1971; Ross, 1983], found renewed interest among AI researchers [Puterman, 1994; Dean et al., 1995]. High-profile, real-world applications of decision theory were deployed [Horvitz & Barry, 1995].

2. Decision theory encompasses probability theory.
3. This reference is a reprint of the original work published in 1763.

Heckerman [1990] argued, concisely and persuasively, that decision theory stands out among other uncertain-reasoning formalisms for its self-consistency, its unambiguous representation of assumptions, its generality, its well-developed theory, and its normative basis. The normative basis is crucial, in light of demonstrated human tendencies to think about uncertainty incorrectly (see Section 2.3.2). Howard [1988] asserted the need for normative reasoning "when we are tempted to think what may not be so," and emphasized a strict adherence to the axioms in the following excerpt:

> Some decision theorists have questioned the normative concepts. They desire to weaken the norms until the normative behavior agrees with the descriptive behavior of human beings, to construct theories of decision making that are both normative and descriptive. A moment's reflection shows that if we have a theory that is both normative and descriptive, we do not need a theory at all. If a process is natural, like breathing, why would you be even tempted to have a normative theory? [Howard, 1988, p. 683]

2.4 Decision analysis

Decision analysis is the application of decision theory [Howard, 1988; Heckerman, 1990]. In developing the alarm framework presented in this dissertation, I chose to adopt a decision-theoretic foundation of reasoning. My work can therefore be regarded as decision-analytic research.

The computations involved in a decision analysis depends on the underlying reasoning objective. I discuss briefly the objectives of diagnosis, control, and attention allocation, showing how these objectives relate to one another.

2.4.1 Diagnosis

Diagnosis involves applying expert knowledge to a set of observations with the goal of determining the set's most probable explanation. Whether the explanation is a known cause or merely a recognized syndrome, diagnostic reasoning typically requires the expert interpretation of multiple observations in concert. Diagnostic models thus fall under the category of event-based approaches.

Decision-theoretic diagnosis in AI typically involves inference on a belief network [Horvitz et al., 1988; Heckerman & Shortliffe, 1992]. Examples of diagnostic belief networks in medicine include the ALARM network for the detection of critical events in anesthesia [Beinlich & Gaba, 1989], and QMR-DT, a comprehensive belief network of diseases and findings in internal medicine [Shwe et al., 1991]. Investigators have also endeavored to incorporate temporal reasoning into the belief-network paradigm. Laursen [1994] developed a belief-network to interpret constellations of ICU-measurement data and their recent trends. Other researchers have also investigated the application of temporal extensions of belief network models [Dagum et al., 1995; Long, 1996; Tudor et al., 1998].

2.4.2 Control

Control is the purposeful, ongoing manipulation of a process. A typical control problem is framed around a set of actions, one of which is applied to the process at any given time. The goal is to determine the action that best satisfies a prespecified objective. The best action may vary with process state; for example, when we are managing the ventilator settings of a mechanically ventilated ICU patient, the optimal ventilator settings depend on the patent's physiologic state, which may vary over time. Control encompasses diagnosis, in that the best action usually depends on what is happening with the process (e.g., a progressive illness). The diagnostic step may be performed separately, or it may be implicitly embedded in the control procedure.

Decision-theoretic control typically relies on a temporally extended decision model. The various approaches all involve the prediction and valuation of future effects of present actions—or inactions. The most general approach of practical interest is the MDP, a multistage decision model in which states of the world at time t influence states of the world at time $t+1$ [Howard, 1960; Howard, 1971; Puterman, 1994]. Figure 2.4 illustrates an influence diagram for a generic MDP. To compute optimal actions for an MDP, we

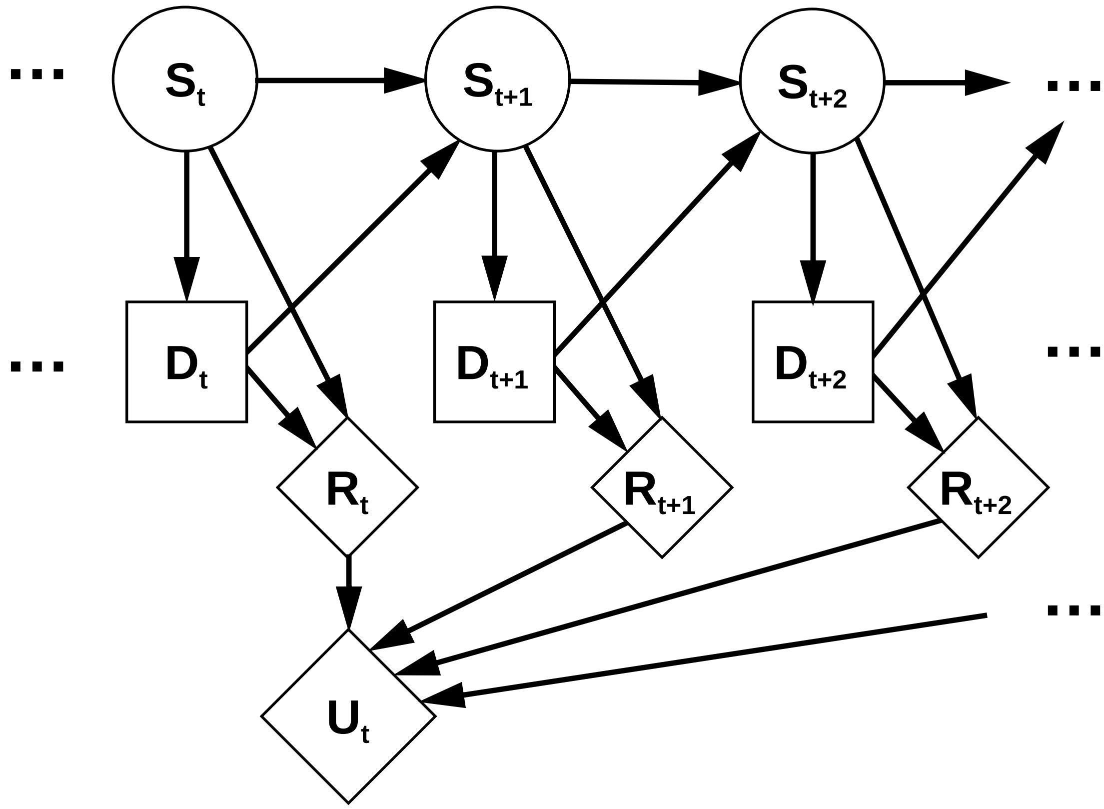


Figure 2.4. Markov decision process.

In this influence diagram of a generic Markov decision process (MDP), circles denote states of the world, squares denote actions to be decided on, and diamonds denote value parameters. Time is divided into discrete epochs, indexed by integers in this figure. Suppose that t denotes the present epoch. The instantaneous reward R_t is influenced by the current state S_t and the current action D_t; the latter two quantities influence the next state S_{t+1}. The total value at time t, U_t, is a discounted sum of the instantaneous rewards from time t onwards. MDPs are explained in Section 3.2.

must consider how various alternative actions affect future decision scenarios, whose chosen actions have ramifications even further into the future, and so on.

The complexity that arises from this interleaving of future states and future actions has driven researchers to develop special modeling techniques that circumvent these difficulties. Horvitz and colleagues characterized a particular class of time-critical decision problems for which time-dependent utilities could be assessed directly [Horvitz & Seiver, 1997; Horvitz & Rutledge, 1991]. Seiver [1992] compressed future decisions and their effects into aggregate variables, which were then assessed directly by domain experts. Although Rutledge and associates [1993], in their design of a ventilator-management advisor, also abstracted away future states and future decisions, they used a physiologic model (as opposed to an expert's direct assessment) to predict the steady-state effects of alternative ventilator settings; these steady-state effects were then, in turn, evaluated according to a utility model developed by Farr [1991].

2.4.3 Attention allocation

Many real-world processes require human attention to ensure optimal control. In an attention-allocation problem, there are several processes competing for a limited amount of human attention resource. Under such circumstances, not every process can be managed optimally; the goal, instead, is to manage the *collection* of processes optimally, given the constraints on attention resource.

The attention-allocation problem introduces additional modeling challenges beyond those of the control problem. Whereas performing control involves determining an optimal action, the decision whether to allocate attention depends on what action is currently being carried out: Whenever the current, or default, action is the optimal action, no attention is needed. Thus, in evaluating the effects of attention, an attention-allocation model must take into account such default actions, in addition to computing the optimal action.

Another modeling challenge, more serious than that of considering default actions, concerns the attention horizon—that is, the interval of time during which limited attention is allocated. Suppose that we are allocating an amount of attention that is to be expended only during the present instant (i.e., and no further into the future). For each process, we can determine the optimal action, compare it with the current action, and compute the amount by which the process would benefit from receiving attention. We would then grant attention to the processes that stand to benefit the most.[4]

4. This scenario assumes that we value each process independently, and that our objective is to maximize the sum total value of the individual processes.

The difficulty arises when we wish to consider attention horizons beyond the present instant. In ascertaining a process's need for attention in the future, we need to reason about future states (which have not occurred) and future actions (which have not been chosen). Therefore, when, at the present time, we allocate attention to be used for a limited duration into the future, we are making assumptions about how these future actions will be chosen. A decision-analytic, attention-allocation model makes these assumptions explicit; the assumptions need to be chosen with care.

2.5 Attention models

How to allocate attention is the heart of the alarm problem. Compared to the diagnosis and control problems, however, the attention-allocation problem has received little attention from decision-analytic researchers. This discrepancy can be attributed to the aforementioned difficulties inherent in formulating and solving such a problem. Recall that, to model the effects of attention over a future duration, we have to make assumptions about actions that have not been chosen. Particular sets of assumptions define particular problem formulations, which in turn are suitable for particular classes of monitoring domains. In other words, the first step in designing an attention-allocation model is to formulate a clear problem definition—a difficult but necessary task that does not confront researchers who are commited exclusively to the computational challenges of diagnosis and control.

We can appreciate the necessity and benefit of making such assumptions by understanding the exact problem that attention-allocation formulations attempt to approximate. In Sections 2.5.1 and 2.5.2, I examine a strict MDP formulation of a single decision maker who is multitasking among several processes. This exact formulation is difficult to solve, and I argue that even readily available solutions to this problem are not useful as alarm signals. In Section 2.5.3, I review pioneering decision-analytic work on an important class of attention-allocation problems, pathologic-process problems, and examine the assumptions that enable this problem formulation. Finally, in Section 2.5.4, I propose a generalization of the model of Section 2.5.3, which leads to the problem formulation (and solution) featured in this dissertation.

2.5.1 Multiple processes in concert

Consider a collection of independently operating processes, each modeled as an MDP. We can control this set of processes optimally by solving the control problem for each process separately. The situation changes, however, when attention resources that are necessary for optimal control become scarce. For example, suppose that we have a decision maker who is responsible for managing these processes, and who can attend to only one of them at a time. In this scenario, the processes become linked: The decision maker cannot pay attention to one process without neglecting the others. Thus, the processes must be considered in concert, even though they are physically disconnected from one another.

We can construct an amalgamated MDP that captures our decision maker as she is multitasking among these processes (Figure 2.5). We define the state space as the cross-product of the state spaces of the constituent processes, and the set of alternatives as the processes themselves. The optimal (or recommended) process to which to attend depends on the combination of states occupied by the constituent processes. The optimal policy is a set of recommendations, each corresponding to a unique state combination; we compute the optimal policy using the policy iteration algorithm developed by Howard [1960].

2.5.2 Problems with the amalgamated MDP

Although amalgamating a collection of processes as a single, continuous-time MDP is theoretically sound, there are several pragmatic concerns associated with building and using an alarm system based on this approach:

- *Computational complexity.* The state space of the amalgamated MDP increases exponentially with the number of concurrent processes; given N processes of M states each, there are a total of M^N states to consider. In practice, the computational effort required for policy iteration is polynomial in the number of states [Littman et al., 1995]. This computational burden renders policy iteration impractical for problems that have nontrivial magnitudes of N or M.

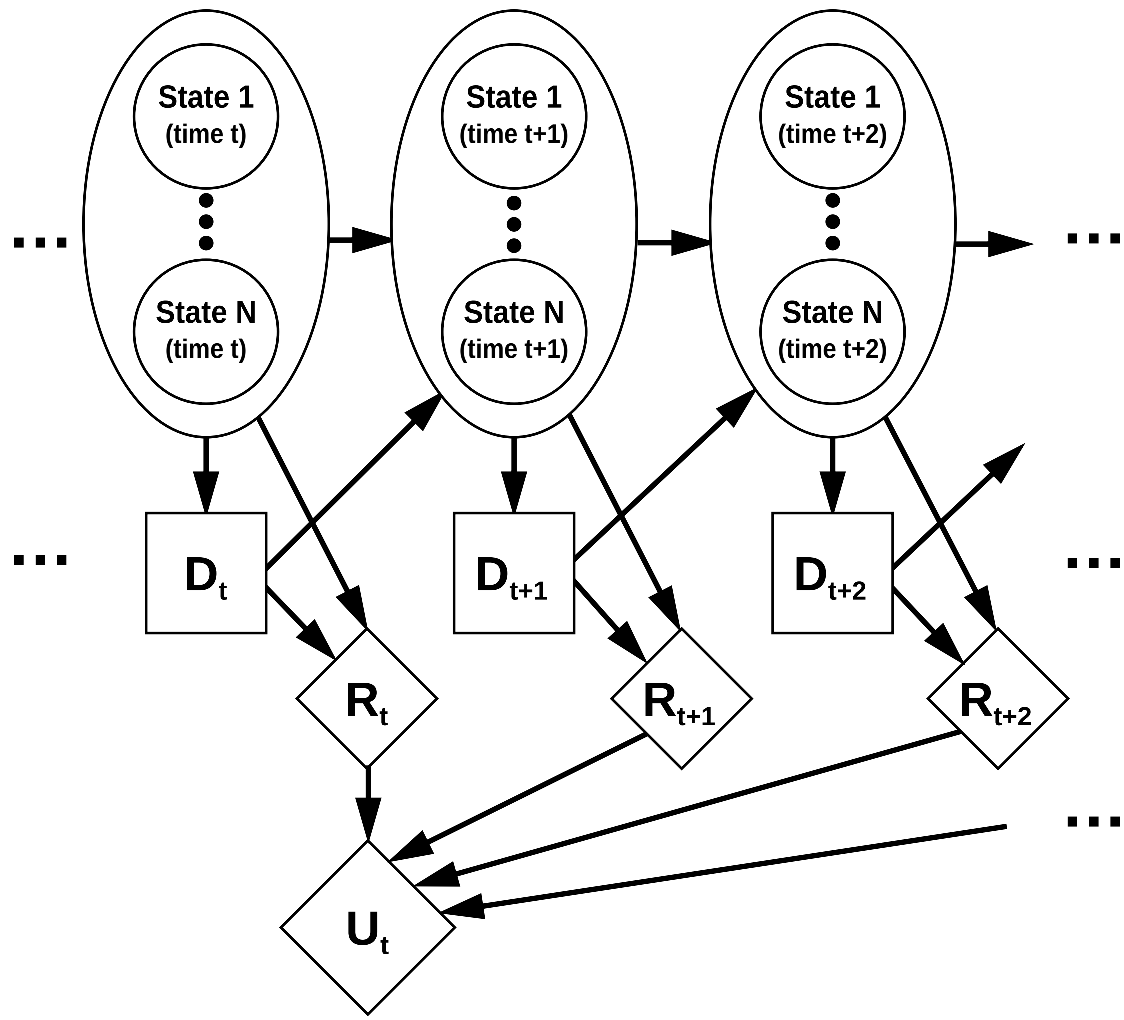


Figure 2.5. Modeling of several processes as a single MDP.

A decision maker monitors N processes, but can attend to only one of them at a time. This decision scenario is modeled as a single MDP. Each constituent process is in turn an MDP with default alternatives; each chance node *State i (time t)* represents the state of process i at time t. The decision to select a process at time t is represented by the decision node D_t; each decision D_t, along with the process states at time t, influences the process states at time $t + 1$. Immediate rewards R_t and the accumulated reward U_t pertain to the entire collection of processes.

- *Nonintuitive explanations.* Decision rules that make up the policy are of the form, “If process 1 is in state j_1, process 2 is in state j_2, …, process N is in state j_N, then pay attention to process k.” Such rules are unwieldy as alarm signals because they force users to think collectively about processes that are physically separate from one another.
- *Consultation and switching overhead.* Optimal policies yield maximum expected benefits only when they are followed. However, it is unrealistic to expect users to consult an alarm that updates its recommendations continually. First, the mere act of consulting the alarm diverts attention from the processes under management. Second, people usually take time to switch their attention between processes (e.g., the processes may be located at some distance apart from each other). During these momentary diversions, no process receives attention; therefore the optimal policy cannot be followed in practice.
- *Disempowerment through micromanaging.* There are also philosophical concerns associated with an alarm that micromanages a decision maker’s every step. Alarms are generally meant to facilitate, rather than to dictate, the management of an expert’s responsibilities. In medicine, in particular, decision-support systems that function as passive consultants are more readily accepted than are those that behave as intrusive advisors [Shortliffe, 1987]. Unfortunately, the micromanaging advice that the amalgamated MDP produces (“go to process 1; now attend to process 3,” etc.) is disempowering and intrusive by nature.

In summary, there are serious, practical problems associated with the amalgamated MDP. Its computational complexity is daunting. The exact recommendation, even if available, is too confusing, too difficult to follow in practice, and too intrusive.

2.5.3 Binary attention allocations

Working around the difficulties presented by the amalgamated MDP, Horvitz developed attention-allocation models for an important class of time-critical decision problems, which he referred to as **pathological-process problems** [Horvitz, 1995; Horvitz & Seiver, 1997]. A pathological process is one whose fate depends largely on the presence and timing of an initial, rapid burst of corrective action; the effects of subsequent actions and observations are comparatively negligible. For example, the fate of a patient who suffers an abdominal hemorrhage from blunt trauma depends on the duration of bleeding as well as on interventions designed to counteract the effects of such bleeding, such as the administration of fluids. For a given process state, the cost of delaying action for a duration T can be represented as a time-dependent utility that is directly assessed.

For such pathological processes, Horvitz demonstrated that the expected cost of delayed action (ECDA) can be readily computed for various delays T. A decision maker who is tending to several processes simultaneously can identify which processes need immediate attention most urgently by ranking them in decreasing order of ECDA. An application of ECDA is the Vista project at the NASA Mission Control Center, which displays selectively propulsion-system information that is most relevant to decisions that operators must make urgently [Horvitz & Barry, 1995]. Horvitz and Seiver [1997] discuss the potential use of ECDA in trauma-care triage and transportation.

Although the ECDA measure is useful because of its simplicity, it is built on numerous assumptions that limit its application scope. Models for computing ECDA were developed for pathological processes, for which time-dependent utilities had been assessed previously. However, many processes are not pathological processes: We often *do* want to consider the interleaving of future actions and observations in our attention-allocation decisions. For example, in deciding how to treat a patient with congestive heart failure now, we should take into account how we would treat her under various future circumstances that, in turn, are influenced by our present choice of treatment.

Related to the pathological-process assumption is the independent computation of ECDA measures for each process. Although convenient, this property of ECDA lies in stark contrast to the intricate interdependence among processes in the exact, multiple-process formulation of Section 2.5.1. We must consider this interdependence whenever we want to reason about the effects of shifting attention between different processes; otherwise, whatever attention we allocate to a process must remain with that process for the entire duration of consideration. Thus, the ECDA measure forces us to allocate attention to processes in an all-or-none fashion: Each process either is controlled optimally, or is completely neglected during the attention horizon T.

2.5.4 Partial attention allocations

My alarm methodology improves the ECDA method of allocating attention in two important ways. First, I represent the constituent processes as MDPs; thus, I enable my attention-allocation model to consider the shifting of attention resources between different processes. Second, my model permits the allocation of partial amounts of attention among the processes. For example, a decision maker could allocate one-third of her attention to one process and two-thirds to another. This commitment of attention means that she switches her attention between the two processes frequently and randomly for a duration T, such that one-third of her time is expended on the first process and two-thirds of her time is expended on the second. Although this partial-attention model is an approximation of real-world actions, it is a more precise approximation than the ECDA model, which permits our decision maker to spend the entire duration T on only one of the processes.

These generalizations of ECDA are significant from both a functional and a methodological standpoint. Functionally, my alarm methodology allocates partial amounts of attention to a set of concurrent processes. To allocate attention in an all-or-none fashion, as ECDA does, closely approximates the optimal allocation in only certain catastrophic situations. Methodologically, my model attacks the interdependencies among processes that occur when attention resources are scarce, while avoiding the difficulties associated with an exact, dictatorial problem formulation (see Sections 2.5.1 and 2.5.2). ECDA, in contrast, assumes that these interdependencies are negligible. There are additional points of comparison in Section 4.7; we shall be in a better position to appreciate them after we learn about the inner workings of my alarm methodology in Chapters 3 and 4.

2.6 Summary

In this chapter, I reviewed previous designs for intelligent alarms. I guided my survey by differentiating these approaches according to a small set of methodologic distinctions. Despite their variety, most of these approaches did not address the problem of allocating limited attention resources among concurrent processes that are in need of such attention.

An intelligent alarm determines where scarce attention resources should be directed. We found that such an alarm needs to infer higher-level events from raw process observations; to consider impending criticalities in addition to those present; and to compute, in the face of various uncertainties, how much a process can expect to benefit from receiving various amounts of attention. In addition, we discovered that, when attention resources are scarce, we must consider the future evolution of each a set of concurrent processes together—rather than separately. We reviewed recent work that addresses this difficult problem, and we previewed how the alarm methodology developed in this dissertation advances this state of the art.

Chapter 3

Utility of Partial Attention

In this chapter, I present a model of the utility of partial attention applied to an ongoing process. I develop the concepts of attention and partial attention using key elements from the theory of Markov decision processes (MDPs). The resulting utility formulation informs busy decision makers how partial compromises in short-term attention affect the overall utilities of the processes they are managing.

3.1 Overview

To model the effects of attention on a process, we need a framework for reasoning about processes in general. In Section 3.2, I introduce the MDP, a general model that is useful for studying time-varying stochastic systems that respond to externally applied actions. I present key results from the theory of MDPs—results that concern the long-term, overall utility of a process under different action strategies. These developments provide a foundation for the remaining sections. In Section 3.3, I investigate, within the MDP framework, what it means to provide or withhold attention from a process. The effects of withholding attention for various durations T are studied in Section 3.4. Finally, in Section 3.5, I develop the concept of partial attention. I construct a computational model of the utility of partial attention applied to a process, and I reflect on the nature and the potential applicability of this model.

3.2 Background

The MDP is a model for describing complex systems that evolve over time. Although the basic theory is well established [Howard, 1960; Howard, 1971; Ross, 1983], MDP-based models remain a subject of active investigation [Puterman, 1994; Dean et al., 1995]. A few key features of the MDP are worth noting at the outset. An MDP is **stochastic** in that its future behavior cannot be predicted with certainty: The system's behavior is *influenced*—rather than necessarily determined—by its present state and by any actions that are presently being applied. The decisions in an MDP are indexed over time: In each decision, a decision maker observes the state of the system, then decides on actions to be applied to the system at that time.

In this section, I develop and illustrate the concept of an MDP, emphasizing notation and highlighting results that are used throughout this dissertation. I adapt the development and running example presented by Howard [1960]. The study of MDPs is predicated on a basic understanding of Markov processes and their associated reward structures, alternatives, and policies. Therefore, let us begin by examining these components.

3.2.1 Markov processes

Our first goal is to describe the dynamic behavior of a system that changes over time. We use the basic concepts of the state of a system and transitions that occur between different states. The **state** is defined by one or more **state variables** that assume specific values at each point in time; each combination of values defines a distinct state. We say that the system occupies a particular state whenever the state variables assume the unique combination of values corresponding to that state. Thus, at any particular time, the system occupies exactly one state; at different times, the system may occupy different states. We say that a **transition** occurs whenever the system changes state.

3.2.1.1 States and transitions

A **Markov process** is a mathematical model that is useful for describing the transition behavior of many complex systems. We specify a particular Markov process as

- *A set of states*: We assume a finite number of states 1, …, N.
- *A transition matrix:* For an N-state Markov process, we specify an N-by-N matrix **A** with components a_{ij}, where the indices i and j refer to the ith row and jth column, respectively. We assume that the matrix **A** is constant over time.

How we specify and interpret the values of the transition matrix depends on whether we are describing a continuous-time or a discrete-time process. In a **discrete-time process,** state transitions occur only at specific time points between equally spaced intervals—for example, every hour. In a **continuous-time process,** the state can change at random intervals. The theories for continuous-time and discrete-time Markov processes are parallel, as are the methods used to analyze those processes. We henceforth consider only continuous-time Markov processes.

We describe the transition behavior of a continuous-time Markov process by specifying the off-diagonal elements of the **A** matrix. Let us call a_{ij} the **transition rate** from state i to state j, for $i \neq j$. The transition rates are nonnegative and are interpreted as follows. Over a short time interval dt, a system that is in state i transitions to a different state j with probability $a_{ij}\,dt$. The diagonal elements of the **A** matrix are defined such that each row sums to zero:

$$a_{ii} = -\sum_{i \neq j} a_{ij}, \qquad \text{for } i = 1, \ldots, N.$$

Let us illustrate the continuous-time Markov process by introducing an example called the **foreman's dilemma** [Howard, 1960]. A machine-shop foreman has a machine that may either be working (state 1) or not working (state 2). When it is working, the machine breaks down with probability $5dt$ in a short time interval dt; when it is not working, the

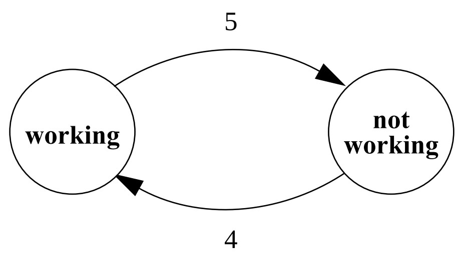


Figure 3.1. A Markov process.
This figure depicts a transition diagram for a two-state, continuous-time Markov process. The nodes *working* and *not working* represent the states. A possible transition between two states is represented by a directed arc and by an associated transition rate. For example, the transition rate from *working* to *not working* is 5.

machine is repaired with probability $4dt$ in dt.[1] Figure 3.1 illustrates a transition diagram for this process. Formally, we describe this two-state process with the transition matrix

$$\mathbf{A} = \begin{bmatrix} -5 & 5 \\ 4 & -4 \end{bmatrix}. \tag{3.1}$$

3.2.1.2 Predicted future states

We can use the transition matrix **A** to answer questions about the future of the process. Because we cannot predict with certainty how the process will unfold, we must ask about *probabilities* of different states at various times in the future.

Let $p_j(t)$ denote the probability that the process is in state j at time t. We use the row vector[2] $\mathbf{p}(t) = \begin{bmatrix} p_1(t) & \dots & p_N(t) \end{bmatrix}$ to summarize the state probability distribution at time t. By convention, we consider $t = 0$ to be the present instant; we are interested in predicting $\mathbf{p}(t)$ for future times $t > 0$.

From the interpretation of transition rates as transition probabilities over an infinitesimal time interval, it can be shown that the state probabilities at time t are governed by the linked differential equations

1. The unit of time is irrelvant for this working example.
2. The state probability vector $\mathbf{p}(t)$ and its variants are row vectors. All other vectors mentioned in this document are column vectors.

$$\frac{d}{dt}p_i(t) = \sum_{i=1}^{N} p_i(t)a_{ij}, \qquad \text{for } i = 1, 2, \ldots, N. \tag{3.2}$$

These equations can be written in matrix form as

$$\frac{d}{dt}\mathbf{p}(t) = \mathbf{p}(t)\mathbf{A}. \tag{3.3}$$

We can solve Equation 3.3 conveniently by using Laplace transformations. Let $\boldsymbol{p}(s)$ be the Laplace transform of the state probability vector $\mathbf{p}(t)$. By taking the Laplace transform and rearranging, we obtain

$$\boldsymbol{p}(s) = \mathbf{p}(0)(s\mathbf{I} - \mathbf{A})^{-1}, \tag{3.4}$$

where the vector $\mathbf{p}(0)$ corresponds to our knowledge of the state at the present time. We then obtain the state probability vector $\mathbf{p}(t)$ by taking the inverse Laplace transform of Equation 3.4.

Let us apply these results to the foreman's example. Suppose that, at time $t = 0$, we are interested in the probability that the machine will be working at time t in the future. We instantiate Equation 3.4 with the matrix $\mathbf{A}$ from Equation 3.1:

$$s\mathbf{I} - \mathbf{A} = \begin{bmatrix} s+5 & -5 \\ -4 & s+4 \end{bmatrix}; \tag{3.5}$$

$$(s\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} \frac{s+4}{s(s+9)} & \frac{5}{s(s+9)} \\ \frac{4}{s(s+9)} & \frac{s+5}{s(s+9)} \end{bmatrix}. \tag{3.6}$$

Expanding each component using partial fractions yields

$$(s\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} \frac{\frac{4}{9}}{s} + \frac{\frac{5}{9}}{s+9} & \frac{\frac{5}{9}}{s} + \frac{-\frac{5}{9}}{s+9} \\ \frac{\frac{4}{9}}{s} + \frac{-\frac{4}{9}}{s+9} & \frac{\frac{5}{9}}{s} + \frac{\frac{4}{9}}{s+9} \end{bmatrix}, \tag{3.7}$$

which can be rewritten as

$$(s\mathbf{I}-\mathbf{A})^{-1} = \frac{1}{s}\begin{bmatrix}\frac{4}{9} & \frac{5}{9}\\ \frac{4}{9} & \frac{5}{9}\end{bmatrix} + \frac{1}{s+9}\begin{bmatrix}\frac{5}{9} & -\frac{5}{9}\\ -\frac{4}{9} & \frac{4}{9}\end{bmatrix}. \tag{3.8}$$

Designating $\mathbf{H}(t)$ as the inverse Laplace transform of the matrix $(s\mathbf{I}-\mathbf{A})^{-1}$, we obtain

$$\mathbf{H}(t) = \begin{bmatrix}\frac{4}{9} & \frac{5}{9}\\ \frac{4}{9} & \frac{5}{9}\end{bmatrix} + e^{-9t}\begin{bmatrix}\frac{5}{9} & -\frac{5}{9}\\ -\frac{4}{9} & \frac{4}{9}\end{bmatrix}. \tag{3.9}$$

The desired solution for the foreman's machine is thus the inverse Laplace transform of Equation 3.4, with $\mathbf{H}(t)$ given by Equation 3.9:

$$\mathbf{p}(t) = \mathbf{p}(0)\mathbf{H}(t). \tag{3.10}$$

Note that an exact numerical solution requires that we specify $\mathbf{p}(0)$, our knowledge of the state at the present time. For example, if the machine is currently working (state 1), we apply $\mathbf{p}(0) = \begin{bmatrix}1 & 0\end{bmatrix}$ to Equation 3.10 and obtain

$$\mathbf{p}(t) = \begin{bmatrix}\frac{4}{9} & \frac{5}{9}\end{bmatrix} + e^{-9t}\begin{bmatrix}\frac{5}{9} & -\frac{5}{9}\end{bmatrix},$$

which can be written as two separate functions $p_1(t) = \frac{4}{9} + \frac{5}{9}e^{-9t}$ and $p_2(t) = \frac{5}{9} - \frac{5}{9}e^{-9t}$ (Figure 3.2a). If the machine is currently not working (state 2), we use $\mathbf{p}(0) = \begin{bmatrix}1 & 0\end{bmatrix}$ and obtain $p_1(t) = \frac{4}{9} - \frac{4}{9}e^{-9t}$ and $p_2(t) = \frac{5}{9} + \frac{4}{9}e^{-9t}$ (Figure 3.2b).

Note that $\mathbf{p}(t)$ approaches $\begin{bmatrix}\frac{4}{9} & \frac{5}{9}\end{bmatrix}$ for large t, whether the machine is presently in state 1 or in state 2. In general, we say that a Markov process is **ergodic** whenever the quantity $\lim_{t\to\infty}\mathbf{p}(t)$ is independent of the starting state $\mathbf{p}(0)$. For an ergodic process, we refer to the **limiting state probabilities** $p_j = \lim_{t\to\infty} p_j(t)$ for $j = 1, \ldots, N$, or simply $\mathbf{p} = \begin{bmatrix}p_1 & \ldots & p_N\end{bmatrix}$. Note that we can calculate $\mathbf{p}$ directly by zeroing the left-hand side of Equation 3.3 and solving the resulting system of equations $\mathbf{pA} = \mathbf{0}$. We employ this technique whenever we are interested in only the limiting state probabilities.

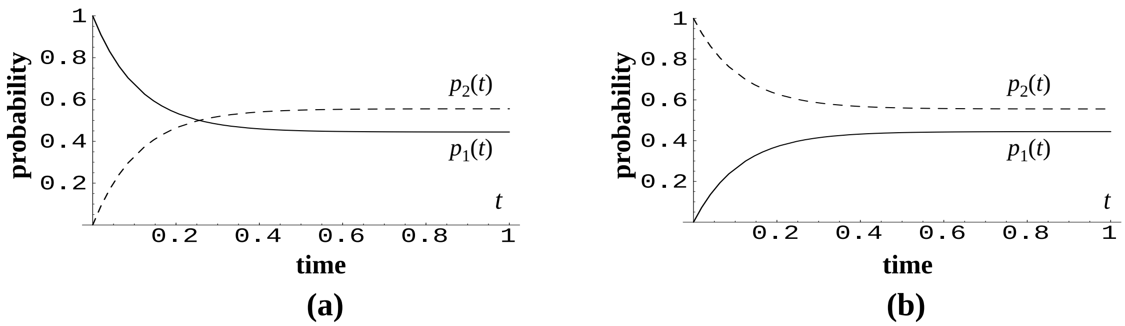


Figure 3.2. State-probability functions (foreman's example).
These graphs illustrate the state probabilities $p_1(t)$ and $p_2(t)$ as a function of time t, for when (a) the machine is in state 1 at time $t = 0$, and (b) the machine is in state 2 at $t = 0$.

The analytical developments for multichain (nonergodic) Markov processes are fundamentally similar to those of their ergodic counterparts; indeed, many multichain processes can be reformulated as ergodic through a redefinition of state [Howard, 1960]. We henceforth consider only those Markov processes that are of the ergodic variety.

3.2.2 Markov processes with rewards

We are now ready to add rewards to our working model. In this subsection, I describe the stream of rewards that a Markov process generates over time. The reward at a given instant depends on the transition behavior of the associated process at that instant.

3.2.2.1 Reward-matrix formulation

We define an N-state **Markov reward process (MRP)** as a Markov process with an associated N-by-N reward matrix **R.** The components r_{ij} of **R** are specified independently, and are interpreted as follows:

- *Diagonal elements:* For $i = 1, \ldots, N$, the quantity r_{ii} is the reward per unit time that the process earns while in state i.
- *Off-diagonal elements:* Each off-diagonal element r_{ij} ($i \neq j$) denotes the reward earned whenever the process transitions from state i to state j. Note that the units of the off-diagonal elements differ from those of the diagonal elements.

3.2.2.2 Earning-rate vector formulation

For purposes of analysis, it is convenient to consider the expected reward per unit time that the system earns at each state. We define an earning-rate vector $\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_N \end{bmatrix}$, where each **earning rate** q_i is defined in terms of the transition matrix and reward matrix elements, according to

$$q_i = r_{ii} + \sum_{i \neq j} a_{ij} r_{ij}, \qquad \text{for } i = 1, \dots, N. \tag{3.11}$$

An MRP can be completely described by its transition matrix **A** and its earning-rate vector **q.** In many processes, specifying the vector **q** is more natural than is specifying the reward matrix **R.** For example, the machine in the foreman's example might earn \$6 per hour in state 1 and –\$3 per hour in state 2. The MRP is then specified completely as

$$\mathbf{A} = \begin{bmatrix} -5 & 5 \\ 4 & -4 \end{bmatrix} \qquad \mathbf{q} = \begin{bmatrix} 6 \\ -3 \end{bmatrix}, \tag{3.12}$$

where the matrix **A** comes from Equation 3.1. This MRP is illustrated in Figure 3.3.

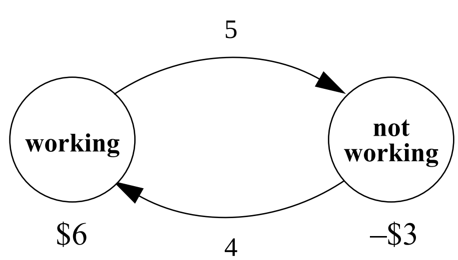


Figure 3.3. A Markov reward process.
The process generates rewards that accumulate over time. It generates \$6 per unit time when in the *working* state, and –\$3 per unit time when in the *not working* state.

3.2.3 Rewards accumulated over time

How much we can expect an MRP to earn over a given time interval? We can answer such questions by using knowledge of the process state at the start of the interval. We now consider two separate, closely related criteria for quantifying accumulated rewards.

3.2.3.1 Undiscounted rewards

For an MRP that is currently operating in state i, let $v_i(t)$ denote the **total reward** that we expect the process to generate between now ($t = 0$) and time t in the future. From the definition of earning rate in Section 3.2.2.2, it can be shown that the total rewards $v_i(t)$ satisfy the linked differential equations

$$\frac{d}{dt}v_i(t) = q_i + \sum_{j=1}^{N} a_{ij}v_j(t), \qquad \text{for } i = 1, 2, \ldots, N. \tag{3.13}$$

Using the **total-reward vector** $\mathbf{v}(t) = \begin{bmatrix} v_1(t) & \ldots & v_N(t) \end{bmatrix}^{\mathrm{T}}$, we can rewrite (3.13) as

$$\frac{d}{dt}\mathbf{v}(t) = \mathbf{q} + \mathbf{A}\mathbf{v}(t). \tag{3.14}$$

As in our prior development of the Markov process, the Laplace transform provides a convenient method of solution. Let $\boldsymbol{v}(s)$ be the Laplace transform of the total-reward vector $\mathbf{v}(t)$. By taking the Laplace transform of Equation 3.14 and solving for $\boldsymbol{v}(s)$, we find that

$$\boldsymbol{v}(s) = \frac{1}{s}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{q} + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{v}(0). \tag{3.15}$$

The inverse Laplace transform of Equation 3.15 then yields the total-reward vector $\mathbf{v}(t)$.

The vector $\mathbf{v}(0) = \begin{bmatrix} v_1(0) & \ldots & v_N(0) \end{bmatrix}^{\mathrm{T}}$ from Equation 3.15 deserves special attention. It materializes from the process of solving Equation 3.14, and must be manually specified as part of any solution $\mathbf{v}(t)$. We interpret $\mathbf{v}(0)$ as follows: When the process terminates in state i, it generates the **terminal reward** $v_i(0)$, for $i = 1, \ldots, N$; we call $\mathbf{v}(0)$ the **terminal-reward vector.** In considering the total reward $\mathbf{v}(t)$ accumulated between now (t = 0) and time t, the terminal-reward vector $\mathbf{v}(0)$ stipulates an amount received at time t, the end of the interval under consideration. Note that the actual terminal reward that we expect at time t depends on the state probabilities $\mathbf{p}(t)$ at time t, and can be computed as the dot product of $\mathbf{p}(t)$ and $\mathbf{v}(0)$.

Let us apply the analysis of Equation 3.15 to the foreman's MRP from Equation 3.12. If we assume that $\mathbf{v}(0)$ is identically zero, then $\mathbf{v}(t)$ is the inverse Laplace transform of the vector $\frac{1}{s}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{q}$. We can perform this compuation methodically by employing a strategy analogous to Equations 3.5 through 3.9. We obtain

$$\mathbf{v}(t) \;=\; t\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}\frac{5}{9}\\-\frac{4}{9}\end{bmatrix} + e^{-9t}\begin{bmatrix}-\frac{5}{9}\\\frac{4}{9}\end{bmatrix},$$

or $v_1(t) \;=\; t+\frac{5}{9}-\frac{5}{9}e^{-9t}$ and $v_2(t) \;=\; t-\frac{4}{9}+\frac{4}{9}e^{-9t}$. The total expected reward in time t is $v_1(t)$ if the machine is presently in state 1, and is $v_2(t)$ if the machine is in state 2. These functions are displayed in Figure 3.4.

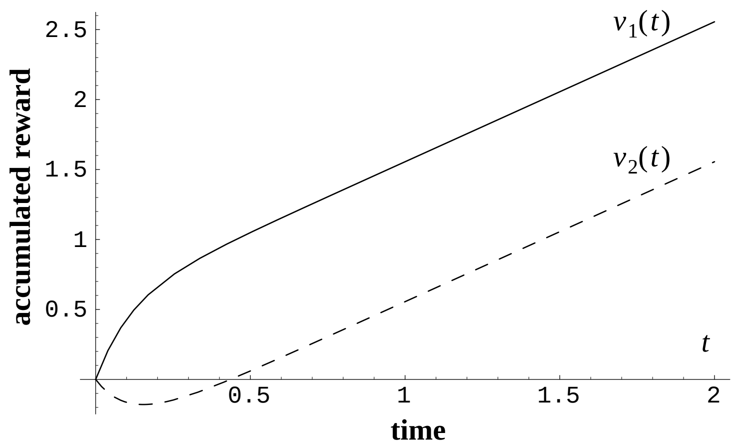


Figure 3.4. Total-value functions (foreman's example).
This graph depicts $v_1(t)$ and $v_2(t)$, the total expected rewards that are accumulated between now and time t from now, when the present state is 1 or 2, respectively.

Note that, as $t \to \infty$, both $v_1(t)$ and $v_2(t)$ asymptotically approach straight lines. The slopes of these lines are called the **gains** of the process. In an ergodic process, the gains for all of the $v_i(t)$'s are identical, so we can properly refer to a single **gain** g. Thus, for an N-state ergodic MRP, we can express the asymptotic total values as

$$v_i(t) = tg + v_i, \qquad \text{for } i \;=\; 1, 2, \ldots, N. \tag{3.16}$$

When we are interested in only the total values accumulated over a long time, we can afford to bypass the exact analysis of Equation 3.15. If we take the time derivative of

Equation 3.16 and substitute the result into the left-hand side of Equation 3.13, we obtain the following system of N equations with $N+1$ unknowns:

$$g = q_i + \sum_{j=1}^{N} a_{ij} v_j, \qquad \text{for } i = 1, 2, \ldots, N. \tag{3.17}$$

By solving this system with v_N set to zero, we can compute the gain g and the values v_i to within a mutually shared additive constant; we call each resulting v_i the **relative value** of the process for state i. For an ergodic process, the gain g is the incremental reward per additional unit of time of operating the process, whereas the quantities $v_i - v_j$ (for $i \neq j$) tell us how much better it is to be in state i than in state j. It is important to remember that these interpretations are valid only when we are considering rewards far into the future.

Let us apply this procedure to the foreman's MRP (Equation 3.12). We set v_2 to zero and solve the system (3.17), obtaining the solutions $g = 1$ and $v_1 = 1$. The gain of 1 tells us that the process is expected to earn \$1 for each additional hour that it operates, regardless of its present state. The relative values $v_1 = 1$ and $v_2 = 0$ indicate that the foreman should be willing to pay up to \$1 to trade in a nonworking machine for a working machine.

3.2.3.2 Discounted rewards

When considering future rewards, we often consider a particular reward amount more valuable if we receive it sooner. For example, \$100 in our posession now is worth more than the same amount promised a year from now, because the money that we possess can be used to earn a year's worth of compounded interest. Also, when we extol the value of delayed gratification in colloquial terms, we are typically referring to a future reward that is worth the wait—in other words, a future reward that is sufficiently greater than the present reward that we forgo. We can model this time preference by incorporating a **discount rate** α into our preceding discussion ($0 < \alpha < \infty$). We define α such that a reward x that is received after an infinitesimal time interval dt is now worth $(1 - \alpha dt)x$. If we are modeling the cost of money, we can interpret α as a continuous compounding rate.

The analysis for this type of discounting parallels that of the undiscounted scheme. For an MRP in state i, let us now call $v_i(t)$ the **present value** of the rewards expected between now and time t. (We also call these quantities **discounted total rewards.**) It can be shown that the present values $v_i(t)$ satisfy

$$\frac{d}{dt}v_i(t) + \alpha v_i(t) = q_i + \sum_{j=1}^{N} a_{ij}v_j(t), \qquad \text{for } i = 1, 2, \ldots, N, \tag{3.18}$$

or, in matrix form,

$$\frac{d}{dt}\mathbf{v}(t) + \alpha\mathbf{v}(t) = \mathbf{q} + \mathbf{A}\mathbf{v}(t). \tag{3.19}$$

The Laplace transform of $\mathbf{v}(t)$ is then

$$\boldsymbol{v}(s) = \frac{1}{s}((s+\alpha)\mathbf{I} - \mathbf{A})^{-1}\mathbf{q} + ((s+\alpha)\mathbf{I} - \mathbf{A})^{-1}\mathbf{v}(0), \tag{3.20}$$

where $\mathbf{v}(0)$ is the terminal-reward vector (see page 43).

Suppose that, in the foreman's example, we are interested in computing discounted total rewards with a discount rate of $\frac{1}{9}$. We apply Equation 3.20 using $\alpha = \frac{1}{9}$ and the values of $\mathbf{A}$ and $\mathbf{q}$ from Equation 3.12, and we again assume that $v_1(0) = v_2(0) = 0$. Taking the inverse Laplace transform of the resulting expression then yields

$$\mathbf{v}(t) = \begin{bmatrix} \frac{783}{82} \\ \frac{351}{41} \end{bmatrix} + e^{-\frac{82}{9}t}\begin{bmatrix} -\frac{45}{82} \\ \frac{18}{41} \end{bmatrix} + e^{-\frac{1}{9}t}\begin{bmatrix} -9 \\ -9 \end{bmatrix},$$

where $\mathbf{v}(t) = \begin{bmatrix} v_1(t) & v_2(t) \end{bmatrix}^{\mathrm{T}}$. These solutions are graphed in Figure 3.5.

Note that as $t \to \infty$, the present values $v_1(t)$ and $v_2(t)$ approach constants, which we refer to as the **limiting present values** v_1 and v_2, respectively. In general, this behavior holds for the discounted total values generated by an MRP, provided that $0 < \alpha < \infty$.

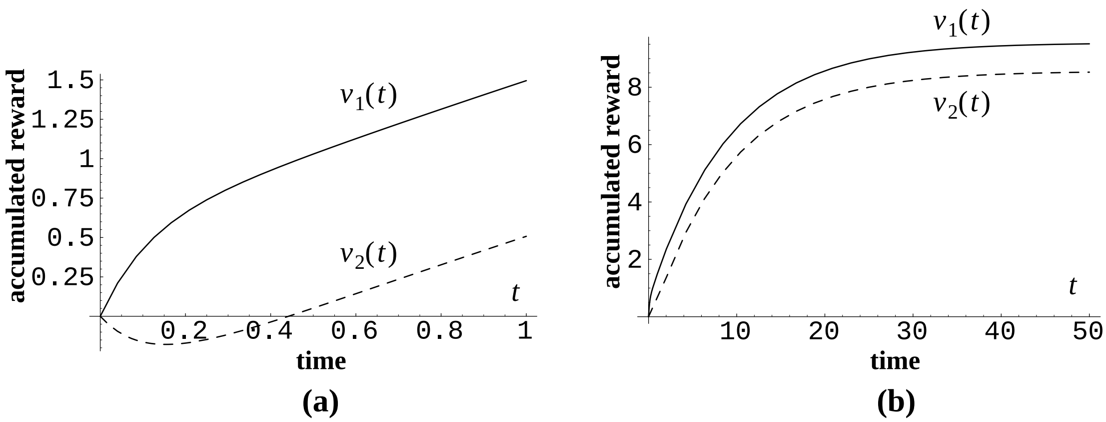


Figure 3.5. Discounted total rewards (foreman's example).
(a) For small values of t, the plots for $v_1(t)$ and $v_2(t)$ resemble those of Figure 3.4. (b) As t becomes very large, both of these functions asymptotically approach constant values.

When we are interested in solely the limiting present values, we can bypass the analysis prescribed by Equation 3.20. For large t, $v_i(t) \to v_i$; substituting v_i for $v_i(t)$ in Equation 3.18 yields the system

$$\alpha v_i = q_i + \sum_{j=1}^{N} a_{ij} v_j, \qquad \text{for } i = 1, 2, \ldots, N, \tag{3.21}$$

which can be solved for the limiting present values v_i. In matrix form, Equation 3.21 reduces to $\mathbf{v} = (\alpha\mathbf{I} - \mathbf{A})^{-1}\mathbf{q}$. Again, it is important to remember that the concept of limiting present values applies to only those processes that have long duration.

3.2.4 Markov decision processes

Thus far, we have investigated processes that generate rewards that depend only on their state trajectories. We are now ready to consider processes whose transition and reward properties can be influenced by the ongoing activities of a decision maker. To consider such processes, we introduce alternatives and decisions into our working model.

In a **Markov decision process (MDP)**, each state is associated with a finite set of alternatives.[3] We describe an N-state MDP by specifying, for each alternative k associated with state i, the following:

- *Transition rates:* For each state $j \neq i$, we specify a transition rate a_{ij}^k. We then assign $a_{ii} = -\sum_{i \neq j} a_{ij}$, for each state i (See Section 3.2.1.1 on page 37.)
- *Earning rate:* We specify the earning rate q_i^k. Alternatively, we can first assess a reward rate r_{ii}^k and transition rewards r_{ij}^k $(j \neq i)$, and then use Equation 3.11 to compute q_i^k. (See Section 3.2.2 on page 41.)

We now extend our story of the foreman's dilemma to incorporate alternatives. An MDP for this modified foreman's dilemma is presented in Table 3.1; its diagrammatic counterpart is shown in Figure 3.6. When the machine is working (state 1), the foreman can choose one of two maintenance regimens. He can employ a normal maintenance strategy, under which the machine will earn \$6 per hour and break down with probability $5dt$ in a short time interval *dt*. Alternatively, he can employ an expensive maintenance program, which would reduce the machine's earnings to \$4 per hour but decrease to 2 per hour the rate of breaking down. When the machine is not working (state 2), the foreman can choose between a normal repair program and an expensive repair program. The transition and earning rates for these latter alternatives are enumerated in Table 3.1.

STATE	ALTERNATIVE	TRANSITION RATE	
i	k	a_{i1}^k	a_{i2}^k
1 (machine working)	1 (normal maintenance)	–5	5
	2 (expensive maintenance)	–2	2
2 (machine not working)	1 (normal repair)	4	–4
	2 (express repair)	7	–7

Table 3.1. The foreman's dilemma.

3. Markov decision processes with continuous alternative sets also exist (e.g., [Ross, 1983]), but they are beyond the scope of our consideration.

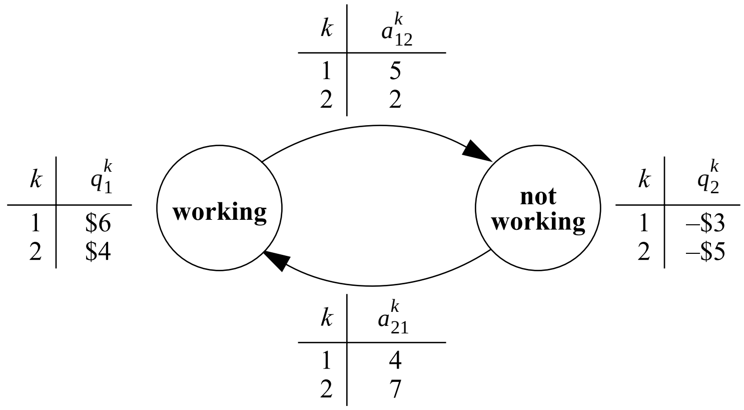


Figure 3.6. A Markov decision process.
Shown is a transition diagram for the MDP specified in Table 3.1. For a given state i, the choice of alternative, k, determines the transition properties and the earning rate. For example, when the process is in the *not working* state (state 2), the first alternative ($k = 1$) yields an earning rate of –$3 and a transition rate of 4 to the *working* state, whereas the second alternative ($k = 2$) yields earning and transition rates of –$5 and 7, respectively.

3.2.5 Policies

We cannot predict how an MDP will behave a priori because its evolution over an interval of time depends on the alternatives chosen during that interval. However, if we decide in advance how alternatives are to be chosen, then we can make predictions about the MDP's future behavior. We call such an imposed alternative-choosing strategy a policy.

A **policy** is a set of decision rules that specify, for each state, which alternative is to be chosen.[4] We encode a policy for an N-state MDP using the vector $\mathbf{d} = \begin{bmatrix} d_1 & \dots & d_N \end{bmatrix}^T$, where each d_i is the numerical index of the alternative prescribed for state i. In the foreman's dilemma, for example, the policy $\begin{bmatrix} 1 & 2 \end{bmatrix}^T$ stipulates that the machine undergo normal maintenance when it is operational ($d_1 = 1$), and undergoes express repair when it is broken ($d_2 = 2$).

4. This definition corresponds to that of a *stationary* policy. More complex policy structures exist. However, when we are considering accumulated rewards earned by an MDP over an infinite horizon, the optimal policy is always a stationary policy.

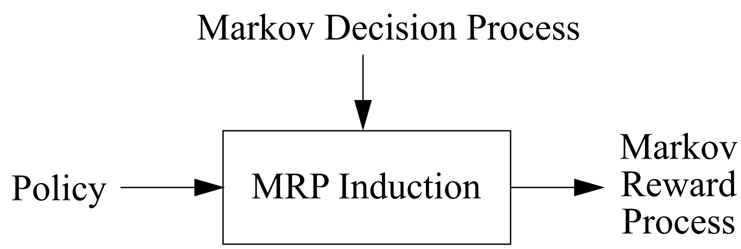


Figure 3.7. The relationship among policies, MDPs, and MRPs.
A Markov reward process results from applying a policy to a Markov decision process.

3.2.5.1 MRP induction

Policies, MDPs, and MRPs are related to one another in an important manner (Figure 3.7). Under a specified policy, the transition and reward properties of an MDP are characterized by a corresponding MRP. For example, the policy $d_1 = 1$, $d_2 = 1$ applied to the foreman's dilemma results in the MRP of Equation 3.12. In general, for an MDP, we say that a policy **induces** an MRP. For example,

$$\mathbf{d} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ induces } \mathbf{A} = \begin{bmatrix} -5 & 5 \\ 7 & -7 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} 6 \\ -5 \end{bmatrix};$$

$$\mathbf{d} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \text{ induces } \mathbf{A} = \begin{bmatrix} -2 & 2 \\ 4 & -4 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} 4 \\ -3 \end{bmatrix}.$$

3.2.5.2 Comparison of policies

Given two policies A and B, we say that policy A is **superior** to policy B if the MRP induced by A yields a greater accumulated reward than the MRP induced by B. Clearly, the mechanics of such a comparison involve specifying the present state, the time interval of interest, and the criterion that we use to value the rewards accumulated during that interval (see Section 3.2.3 on page 42).

As an example, let us consider the total earnings in the foreman's dilemma under an infinite horizon ($t \to \infty$) and a discount rate $\alpha = \frac{1}{9}$. We investigate each policy by computing the present values for the MRP induced by that policy (Equation 3.21). The present values for each policy are shown in Table 3.2.

Policy (**d**)		Present Values (**v**)	
d_1	d_2	v_1	v_2
1	1	9.5488	8.5610
1	2	13.1284	12.2202
2	1	15.3818	14.2364
2	2	18.2195	17.2317

Table 3.2. Comparison of candidate policies.
The present values of candidate policies for the foreman's dilemma are shown. Rewards are assumed to be accumulated over an infinite horizon with discount rate $\frac{1}{9}$. The quantity v_i for a given policy is the expected value if the process is in state i. For example, under the policy $d_1 = 1$, $d_2 = 2$, the process is worth \$13.13 if it is in state 1.

Comparing the present values under each policy, we see that if we are in state 1, then the policy $d_1 = 2$, $d_2 = 2$ yields the maximum return (\$18.22). If we are in state 2, the same policy $\begin{bmatrix} 2 & 2 \end{bmatrix}^T$ is superior to all others. Thus, the policy $\begin{bmatrix} 2 & 2 \end{bmatrix}^T$ is in some sense optimal, because it produces a maximum return regardless of in which state we happen to be. This policy suggests that the machine should receive expensive maintenance when it is working and express repair when it is not working, and that the long-term rewards justify these more expensive alternatives.

We might ask whether the existence of an optimal policy in this problem is mere coincidence. It turns out that, for an infinite horizon MDP, we can always find a policy that produces a maximum return, regardless of the present state [Howard, 1960]. This guarantee holds for both undiscounted and discounted criteria for appraising streams of predicted rewards.

3.2.5.3 Optimal policies

We now formally define optimal policies for an infinite-horizon MDP. Let us begin with the discounted scenario. For an infinite-horizon MDP with a nonzero discount rate, we say that a policy is **optimal** if there is no policy that is superior to it, in the following sense: Given policies A and B with corresponding present values v_i^A and v_i^B, policy A is superior to policy B if at least one $v_i^A > v_i^B$, and no $v_i^B > v_i^A$. For the undiscounted scenario, we compare policies first according to their gains, then according to their relative values: Given policies A and B with respective gains g^A, g^B and relative values v_i^A, v_i^B, policy A is superior to policy B if $g^A > g^B$. If $g^A = g^B$, then policy A is superior to policy B if at least one $v_i^A > v_i^B$, and no $v_i^B > v_i^A$.

How do we find an optimal policy for an infinite-horizon MDP? In general, it is impractical to search exhaustively as we did in Table 3.2. If $k(i)$ is the number of alternatives for state i, then there are $\prod_{i=1}^{N} k(i)$ policies to consider; in other words, for a fixed number of alternatives, the number of policies grows exponentially with the number of states. Fortunately, we do not need to perform such an unguided search: A procedure has been developed that takes a policy A and produces a superior policy B, if one exists [Howard, 1960]. The method of **policy iteration** involves repeated applications of this procedure, and is always guaranteed to produce an optimal policy—usually in a small number of steps. Variants of policy iteration have been developed for the undiscounted and discounted scenarios.

3.2.5.4 Constant-value policies

For an MDP with undiscounted future rewards, the MRP induced by a policy generates a total reward amount that increases linearly with the amount of time t that the process operates, for large values of t (see Figure 3.4 on page 44). For these processes, we have seen that the method of policy iteration attempts to maximize the gain and, if necessary, the relative values. On the other hand, when we discount future rewards, the total reward increases toward a constant as $t \to \infty$ (see Figure 3.5 on page 47). In these scenarios, the policy-iteration algorithm searches for a policy that optimizes the relative values.

The ensuing developments are based on MDPs whose infinite-horizon policies have constant values. We henceforth restrict our attention to discounted processes, and to undiscounted processes with a gain of zero.

3.3 Attention

The preceding discussion on MDPs lays the groundwork for our investigation of processes that receive varying amounts of short-term attention. We now begin using this foundation to describe formally the notion of attention. Our concept of what it means to give a process attention requires only a modest extension of the existing MDP framework.

3.3.1 Default alternatives and policies

Recall that an MDP associates a finite set of alternatives with each state (see Section 3.2.4). For each state, let us distinguish one of the associated alternatives as the **default alternative** for that state. Since the alternatives for a given state may be ordered arbitrarily, let us adopt the convention of designating alternative 1 as the default alternative for that state.

For the foreman's dilemma (Table 3.1 on page 48), we can specify the default alternatives as follows. When the machine is working, the default alternative is to employ a normal maintenance regimen (alternative 1 for state 1). When the machine is not working, the default alternative is to employ a normal repair program (alternative 1 for state 2). We can of course specify other alternatives as default alternatives by simply reassigning the numeric indices associated with the alternatives for a given state.

For any MDP with default alternatives specified, we can talk about its default policy. The **default policy** of an MDP is the set of decision rules that specifies the default alternative for each state. According to our numbering convention, the default policy for an N-state MDP is $d_1 = 1$, $d_2 = 1$, …, $d_N = 1$, or simply an N-element vector of 1s. For example, the default policy for the foreman's dilemma is encoded as $\begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathrm{T}}$.

3.3.2 Attention and the optimal policy

The notion of attention can now be defined as the enabling of an MDP's optimal policy: *The MDP operates under the optimal policy whenever attention is applied; otherwise, the MDP operates under the default policy.* For example, in the foreman's dilemma under an infinite horizon and a discount rate of $\frac{1}{9}$, deciding whether or not to apply attention amounts to choosing between the optimal policy $\begin{bmatrix}2 & 2\end{bmatrix}^{\mathrm{T}}$ and the default policy $\begin{bmatrix}1 & 1\end{bmatrix}^{\mathrm{T}}$.[5]

In general, we can apply this model of attention to *any* MDP that has specified default alternatives and discount criteria (Figure 3.8). We use policy iteration to compute, for a given discount criterion, the optimal policy and the **optimal values**—the total-value vector that results from executing this policy for all time. The optimal policy then induces an optimal MRP, whereas the default policy induces a default MRP (see Section 3.2.5.1).

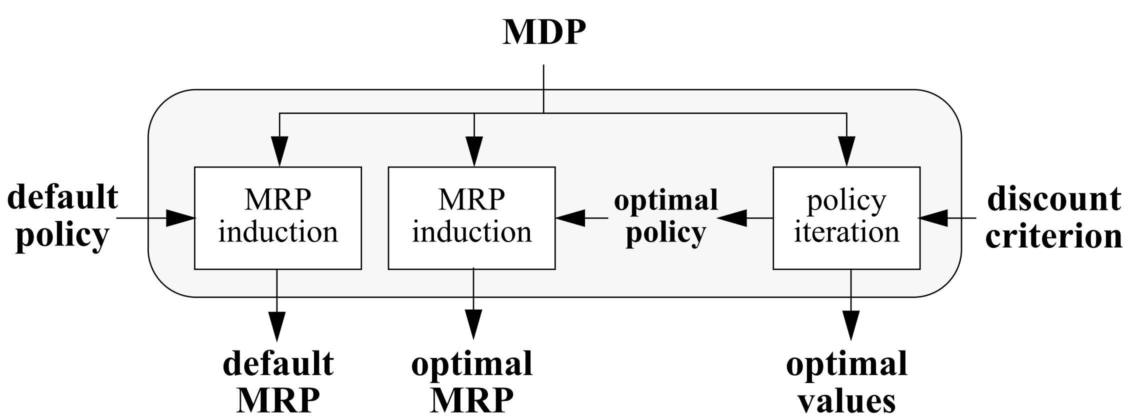


Figure 3.8. Conversion of an MDP into default and optimal reward processes. This figure illustrates the steps involved in converting an MDP to a form that is useful for our working process model. The default MRP is the reward process that prevails whenever the process is unattended; the optimal MRP prevails when attention is applied. The optimal values are the expected total rewards for each starting state when the optimal policy is applied for all time. Only the default MRP, optimal MRP, and optimal values are needed for subsequent analysis.

5. In general, it is possible for the default and optimal alternatives for one or more states to coincide. Whenever the process is in such a state, its immediate transition and reward behaviors do not depend on the attention received (or not received) at that time.

Note that the preceding procedure always produces an equivalent MDP with the alternatives *no attention* and *attention* specified appropriately for each state. The default MRP describes the transition and reward properties of the process when that process is not receiving attention; the optimal MRP describes these process properties when attention is applied. We employ the default MRP, the optimal MRP, and the optimal values in our subsequent analysis.

3.4 Delayed attention

An MDP generates rewards according to the policy that is applied to the process. In the special case of a **stationary policy** (one that is applied over all time), we can compute the total expected reward by analyzing the MRP that is induced by that policy: solving the system (3.17) in the case of an undiscounted process, or solving (3.21) for a discounted process. The situation is more complicated when a policy may vary over time. To compute total rewards for such **nonstationary policies**, we must apply methodically the mathematical machinery of Section 3.2.

In this section, we analyze a particular class of nonstationary policies. We consider an MDP that is unattended for a period of time T before receiving attention thereafter. From our development of the notion of attention in Section 3.3, we saw that leaving a process unattended is the same as operating the process under its default policy, whereas giving attention is the same as operating the process under its optimal policy. Thus, the situation that we wish to consider corresponds to applying the default policy for T, then applying the optimal policy thereafter (Figure 3.9). Our goal is to compute the total expected reward under these conditions.

3.4.1 Notation

Our analysis builds on the development of Section 3.3. To faciliate our subsequent exposition, we let the pairs $\mathbf{A^0}, \mathbf{q^0}$ and $\mathbf{A^1}, \mathbf{q^1}$ denote the default MRP and optimal MRP, respectively. Furthermore, we let $\mathbf{v^1}$ denote the vector of optimal values that results from

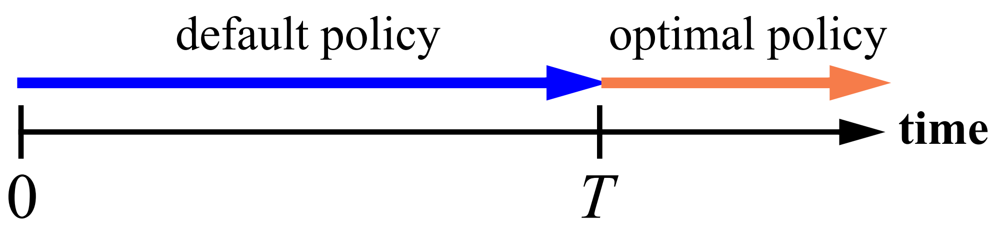


Figure 3.9. Delayed attention.
Delaying attention for time T is implemented in the MDP framework as application of the default policy for time T, and application of the optimal policy thereafter. Note that when T is zero, there is no delay of attention, and the optimal policy is applied over all time.

applying the MRP $\mathbf{A^1}$, $\mathbf{q^1}$ over all time. In the case of the foreman's dilemma with a discount rate of $\frac{1}{9}$, we have

$$\mathbf{A^0} = \begin{bmatrix} -5 & 5 \\ 4 & -4 \end{bmatrix}, \mathbf{q^0} = \begin{bmatrix} 6 \\ -3 \end{bmatrix}; \tag{3.22}$$

$$\mathbf{A^1} = \begin{bmatrix} -2 & 2 \\ 7 & -7 \end{bmatrix}, \mathbf{q^1} = \begin{bmatrix} 4 \\ -5 \end{bmatrix}; \tag{3.23}$$

$$\text{and } \mathbf{v}^1 = \begin{bmatrix} 18.2195 \\ 17.2317 \end{bmatrix}, \tag{3.24}$$

where $\mathbf{v^1}$ is obtained by applying the system of equations (3.21) to the MRP $\mathbf{A^1}$, $\mathbf{q^1}$.

3.4.2 Value of delayed attention

We can now state our problem as follows. Given a process that is in state i with discount rate α, what total reward can we expect to accumulate over the interval $0 \le t < \infty$, if we operate the MRP $\mathbf{A^0}$, $\mathbf{q^0}$ for $0 \le t < T$, then operate the MRP $\mathbf{A^1}$, $\mathbf{q^1}$ for $t \ge T$?

Let us designate $v_i(T)$ as our target quantity. We decompose $v_i(T)$ additively into two subquantities: the rewards accumulated during the interval $0 \le t < T$, and the rewards accumulated for $t \ge T$. However, for a given state j at time $t = T$, the second subquantity is simply the jth element of the optimal value vector $\mathbf{v^1}$. Thus, we can compute $v_i(T)$ by summing the expected rewards from the MRP $\mathbf{A^0}$, $\mathbf{q^0}$ over the *finite* interval $0 \le t < T$, and incorporating a terminal reward amount that is determined by $\mathbf{v^1}$ and the state-probability distribution of the process at time T. Figure 3.10 depicts this computational strategy.

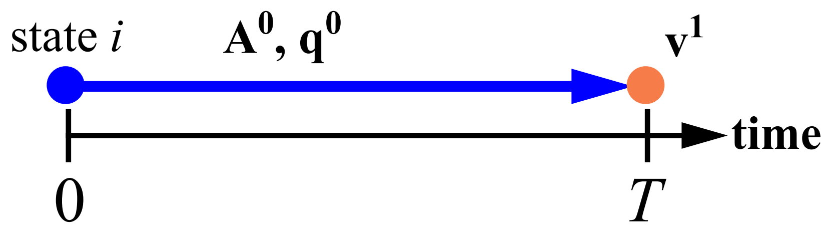


Figure 3.10. Computational model for the value of delayed attention. Consider a process that is in state i at $t = 0$. During the interval $0 \le t < T$, state transitions occur according to the transition matrix $\mathbf{A^0}$, and rewards are expected according to the reward vector $\mathbf{q^0}$ and the state-probability distributions during this interval. At time T, the process terminates and yields a terminal reward that is determined in part by the optimal values $\mathbf{v^1}$. This terminal reward incorporates the rewards generated by the MRP $\mathbf{A^1}$, $\mathbf{q^1}$ for times beyond T.

To make the strategy concrete, we adapt the analysis of Equation 3.20. Using $\mathbf{A^0}$ as the transition matrix, $\mathbf{q^0}$ as the reward vector, and $\mathbf{v^1}$ as the terminal-reward vector, we have

$$\boldsymbol{v}(s) = \frac{1}{s}((s+\alpha)\mathbf{I} - \mathbf{A^0})^{-1}\mathbf{q^0} + ((s+\alpha)\mathbf{I} - \mathbf{A^0})^{-1}\mathbf{v^1}; \tag{3.25}$$

$$\mathbf{v}(t) = \mathcal{L}^{-1}\{\boldsymbol{v}(s)\}. \tag{3.26}$$

Substituting $t = T$ into Equation 3.26 yields the vector $\mathbf{v}(T) = \begin{bmatrix} v_1(T) & \dots & v_N(T) \end{bmatrix}^{\mathrm{T}}$, from which we obtain the desired quantity $v_i(T)$.

Figure 3.11 illustrates the results of applying Equations 3.25 and 3.26 to the foreman's dilemma (Equations 3.22 through 3.24, $\alpha = \frac{1}{9}$). Whether the machine is currently in state 1 or in state 2, the value of delaying attention decreases asymptotically as the delay amount increases. In other words, the cost of delaying attention increases with increasing delay.

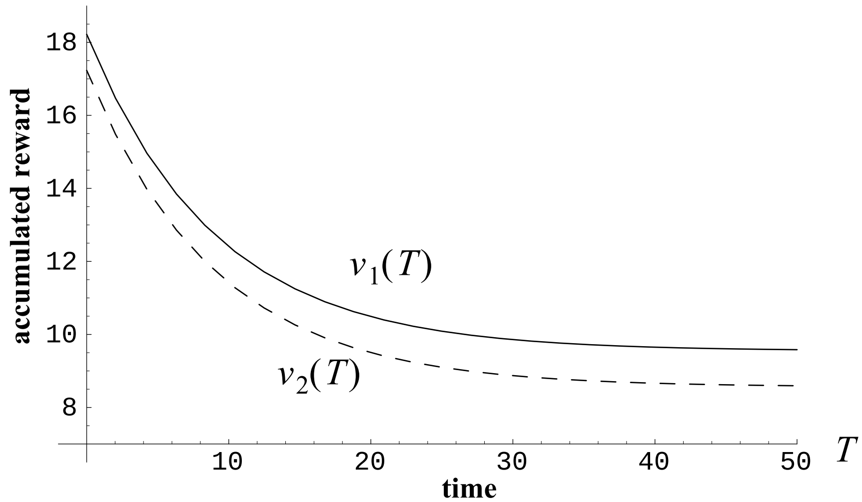


Figure 3.11. Value of delayed attention (foreman's example).
Each quantity $v_i(T)$ is the total discounted reward expected from a delay of attention for a period T when the machine is currently in state i. As the delay T increases, the default policy prevails for longer periods of time before the optimal policy takes over, and the expected total reward decreases.

3.5 Partial attention

We have reasoned about the effects of leaving a process unattended for a limited period of time. Let us now extend our present conversation by examining various degrees of **partial attention** administered during that time period. For example, we would like to talk about the foreman devoting one-third of his attention to the machine over the next hour.

The phrase *partial attention* is a misnomer when we consider time at a microscopic level, because at any instant in time a process is either receiving attention or not receiving attention. When we are considering more measurable units of time, however, we can speak meaningfully about applying partial attention to a process. Suppose that our foreman devotes 20 minutes of his attention to the machine over the next hour. Then, in an imprecise sense, we can say that the machine receives one-third of the foreman's attention during that hour. Our statement would be imprecise in that those 20 minutes could be distributed in any one of an infinite number of ways.

In this section, we develop the concept of partial attention by building on our working analysis of the MDP. Our goal is to derive a measure of the total expected reward generated by a process that initially receives a fraction of attention θ (where $0 \le \theta \le 1$) for a duration T, before receiving full attention thereafter (Figure 3.12). To derive this exact numeric quantity from the inexact notion of partial attention, we must assume a particular distribution of attention during the interval $0 \le t < T$. We now delve into the mechanics of this assumption, then ponder its meaning and potential usefulness.

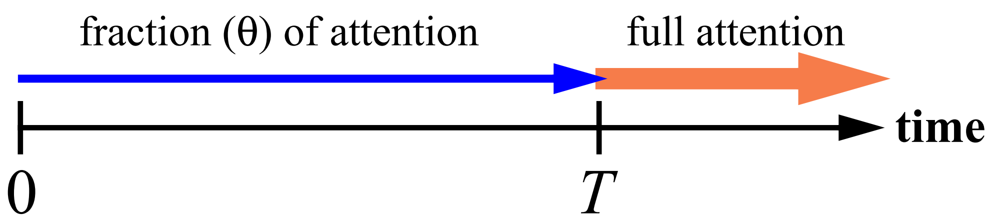


Figure 3.12. Partial attention.
A process receives a fraction of attention θ for a duration T before receiving full attention. This formulation of partial attention is illustrated in more detail in Figures 3.13 and 3.14.

3.5.1 Uniform distribution of attention

Consider a fraction of attention θ that is applied during the time interval $0 \le t < T$. Although our concept of attention fraction is thus far imprecise, we can immediately conclude that the process somehow receives attention for θT time units (and consequently no attention for $(1-\theta)T$ time units). What remains to be specified is how the applied attention is distributed over the specified interval.

Our model of partial attention assumes that the available attention θT is temporally distributed in a *uniform* manner over the interval $0 \le t < T$. We can picture this interval partitioned into a large number of equally spaced subintervals of length Δt, each subinterval containing $\theta \Delta t$ time units of attention (Figure 3.13). In the limit as $\Delta t \to 0$, the desired, effective partial level of attention θ is applied throughout the interval $[0, T)$.

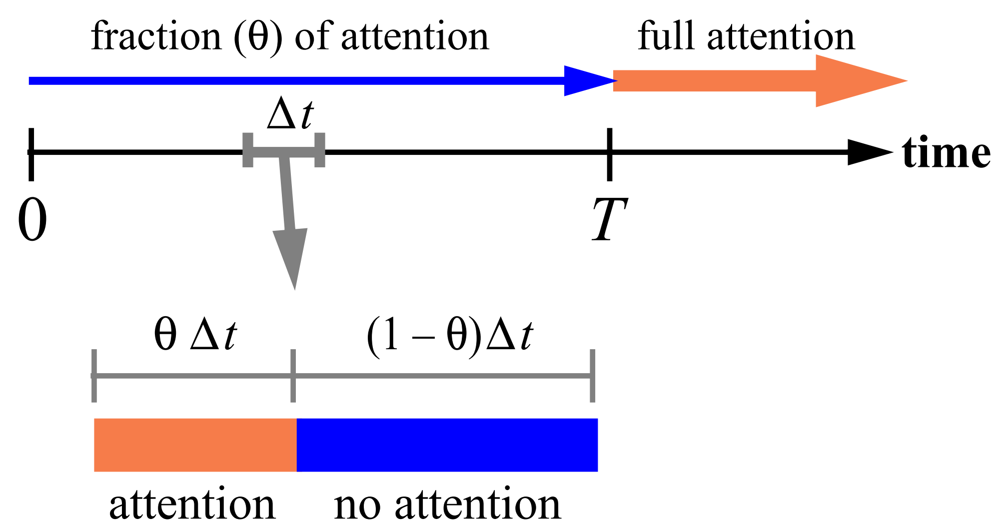


Figure 3.13. Uniform distribution of partial attention.
Each infinitesimal subinterval Δt within $[0,\ T)$ contains $\theta\Delta t$ time units of attention.

3.5.2 Hybrid default-optimal policies

Now how does the process actually behave under partial attention? In the model outlined in the previous paragraphs, we saw that the attention fraction θ applies to every *time point* within the interval $[0, T)$. We are thus confronted with the problem of modeling partial levels of attention applied to instants in time—a concept that we have previously declared invalid (see page 58).

Fortunately, we can obtain an effective attention fraction of θ by modeling the process dynamics at each instant as an uncertain prospect. When the available time is uniformly distributed, the level of attention at each time point is characterized by θ: For every instant $0 \le t < T$, the process receives attention with a probability θ and no attention with probability $1 - \theta$. Equivalently, for these instants, the optimal MRP $\mathbf{A}^1, \mathbf{q}^1$ transpires with probability θ; the default MRP $\mathbf{A}^0, \mathbf{q}^0$ transpires with probability $1 - \theta$ (Figure 3.14).

Reflection shows that these dynamics are summarized by a hybrid MRP $\mathbf{A}^\theta,\ \mathbf{q}^\theta$ whose elements are weighted sums of the corresponding elements, the default and optimal MRPs. Therefore, we have, in matrix form,

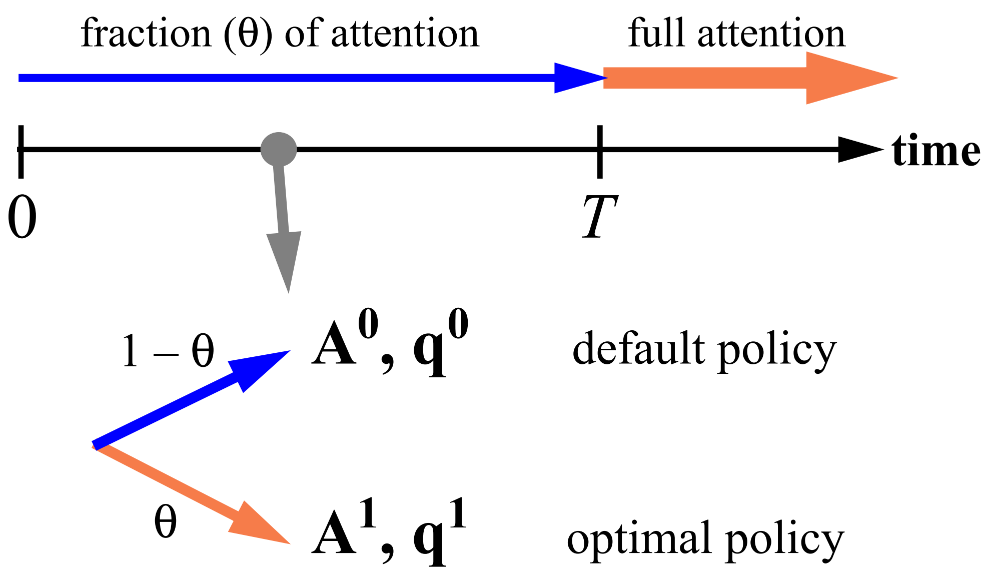


Figure 3.14. Partial attention at an instant in time.
At each instant, the process receives attention with probability θ and no attention with probability $1 - \theta$. The expected attention amount for an infinitesimal interval dt is θdt; integrating this expression over $[0, T)$ yields θT expected time units of attention, corresponding to an effective attention fraction of θ.

$$\mathbf{A}^{\theta} = (1-\theta)\mathbf{A}^{\mathbf{0}} + \theta\mathbf{A}^{\mathbf{1}}; \tag{3.27}$$

$$\mathbf{q}^{\theta} = (1-\theta)\mathbf{q}^{\mathbf{0}} + \theta\mathbf{q}^{\mathbf{1}}. \tag{3.28}$$

3.5.3 Value of partial attention

Now we are ready to compute the value of applying partial attention for a limited amount of time. Let $v_i^{\theta}(T)$ denote the total expected reward of a process in state i that receives partial attention θ for a duration T before receiving full attention. Assume a discount rate α. We adapt the analysis for the value of delaying attention (Equations 3.25 and 3.26), with the hybrid MRP $\mathbf{A}^{\theta}$, $\mathbf{q}^{\theta}$ taking the place of the default MRP $\mathbf{A}^{\mathbf{0}}$, $\mathbf{q}^{\mathbf{0}}$. We obtain:

$$\boldsymbol{v}^{\theta}(s) = \frac{1}{s}((s+\alpha)\mathbf{I} - \mathbf{A}^{\theta})^{-1}\mathbf{q}^{\theta} + ((s+\alpha)\mathbf{I} - \mathbf{A}^{\theta})^{-1}\mathbf{v}^{\mathbf{1}}; \tag{3.29}$$

$$\mathbf{v}^{\theta}(t) = \mathcal{L}^{-1}\{\boldsymbol{v}^{\theta}(s)\}. \tag{3.30}$$

Substituting $t = T$ into Equation 3.30 yields the vector $\mathbf{v}^{\theta}(T) = \left[v_1^{\theta}(T) \; \ldots \; v_N^{\theta}(T) \right]^{\mathrm{T}}$, from which we obtain the desired quantity $v_i^{\theta}(T)$ (see Figure 3.15).

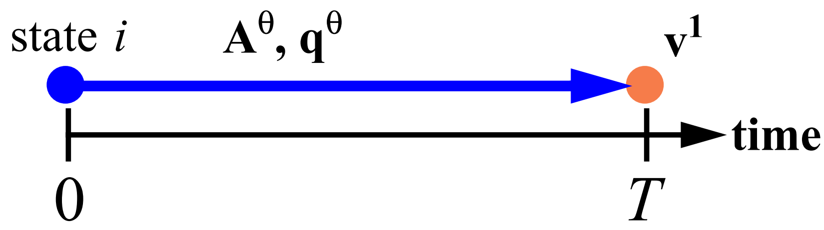


Figure 3.15. Computational model for the value of partial attention. Starting from state i at $t = 0$, the process transitions and generates rewards according to the hybrid MRP $\mathbf{A}^{\theta}$, $\mathbf{q}^{\theta}$. At time T, the process terminates and yields a terminal reward that incorporates the rewards generated by the MRP $\mathbf{A}^{1}$, $\mathbf{q}^{1}$ for times greater than T.

In general, Equation 3.30 cannot be computed in closed form for an unspecified attention fraction θ: Small changes in θ may cause the inverse Laplace transform to vary not only in its numerical elements, but also in its functional form—sometimes abruptly. However, a closed-form expression can be obtained from Equation 3.30 for any *specific* value of θ. So, if we are interested in how the quantity $v_i^{\theta}(T)$ varies with θ, we must compute $v_i^{\theta}(T)$ numerically for a specified, discrete sample set $\{\theta_1, \theta_2, \ldots\}$.

3.5.4 Examples

Let us invoke once again our running example of the foreman's dilemma (Equations 3.22 through 3.24) with a discount rate $\alpha = \frac{1}{9}$. Figure 3.16 plots the machine's expected returns for different levels of attention θ over the next hour ($T = 1$). Whether the machine is in the *working* ($i = 1$) or *not working* ($i = 2$) state, the total returns $v_i^{\theta}(1)$ increase with θ in an approximate linear fashion.[6]

We can simultaneously vary the delay T and the level of attention θ. An example is shown in Figure 3.17.

6. In Section 4.6 and Chapter 5, we shall examine more complex models—models for which an asymptotic exponential function characterizes more accurately the change in value with respect to θ.

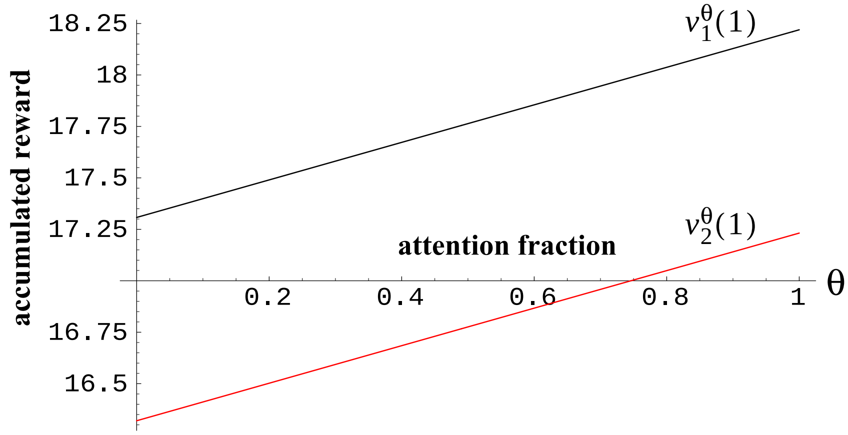


Figure 3.16. Total rewards as a function of attention fraction.
This figure plots the total expected rewards $v_1^\theta(1)$ and $v_2^\theta(1)$ as a function of the attention fraction θ. To generate these plots, I instantiated Equations 3.29 and 3.30 iteratively with values of θ ranging from 0 to 1 in increments of 0.01.

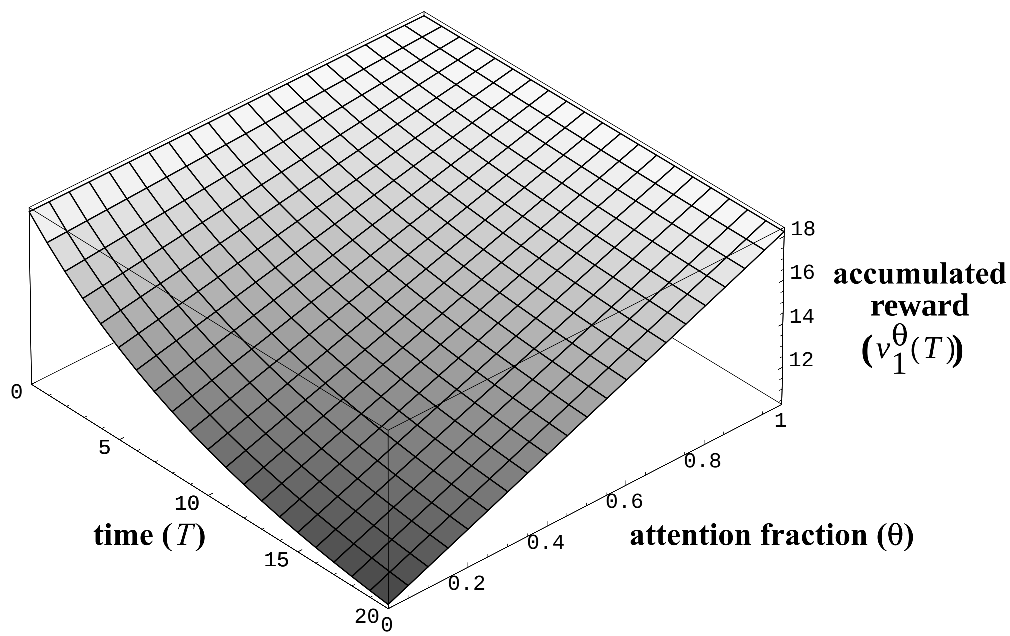


Figure 3.17. Varying of delay and attention level simultaneously.
This surface diagram displays the value of the machine in state 1, as a function of the delay T and the attention level θ. The value decreases with decreased attention level and increased delay.

3.5.5 Discussion

In the analysis that culminated with Equation 3.30, we constructed exact formulae for the value of partial attention applied to a process for a duration of time. In obtaining these exact expressions, we assumed that the applied attention was distributed uniformly—that the process received the same partial level of attention for the entire duration of interest.

Let us reflect on what this approximation means for the decision maker. In Section 3.5.2, we adopted a probabilistic interpretation of the uniformly distributed attention fraction θ. At every instant, the decision maker is either attending or not attending to the process; whether or not the process receives attention may differ between neighboring instants. Thus, a decision maker characterized by our model randomly switches between applying attention and withholding attention at an infinite rate, in a manner that provides the process an expected attention fraction of θ over the duration of interest.

In reality, decision makers cannot multitask in such a idealized, frenzied manner. Attention is typically granted to the process or withheld in favor of another task for a measurable length of time. Also, there is usually a time cost associated with switching between tasks. Thus, Equation 3.30 is an *exact* implementation of an *approximate* model of a decision maker's limited attention. The approximation is unrealistic when the process transitions much faster than the decision maker's ability to switch to or away from the process. However, when process dynamics are slow in comparison to the decision maker's multitasking abilities, the model of partial attention represented by Equation 3.30 is plausible.

3.6 Summary

In this chapter, I described the development of formulae for the utility of processes that receive partial levels of attention over a specified temporal horizon. My analysis applies to processes that can be modeled as a continuous-time MDP. To model the notion of attention, I augmented the MDP model with default alternatives and default policies. With these added elements, I defined giving attention as applying the optimal policy, and withholding attention as applying the default policy. Moving beyond policies that are applied uniformly over all time, I derived formulas for the utility of a process that receives no attention for a limited duration before receiving full attention thereafter. I then extended these analyses to incorporate the notion of applying partial attention over a limited duration. My model of partial attention is plausible for processes whose transition rates are sufficiently small, relative to the decision maker's ability to switch tasks quickly.

The need to compromise attention is often unavoidable in situations that involve multiple processes competing for limited amounts of available attention. Knowing how partial compromises in short-term attention affect the long-term health of different processes can help decision makers to allocate their limited attention rationally. The manner in which this allocation occurs is the subject of the following chapter.

Chapter 4

Alarm Signals for Concurrent Processes

Having studied a single process under various degrees of partial attention, we are now ready to consider the simultaneous operation of a collection of such processes. In this chapter, I present a model of the management of concurrent, continuing processes under constraints on the available attention resources. The model is constructed as an optimization problem, and is grounded in decision theory, as the components of the optimization build on ideas and developments from the previous chapter.

In analyzing this model, I highlight particular numerical metrics as alarm signals. These metrics serve to inform, rather than to dictate. They can guide the attention of busy agents who are managing several processes to which they cannot attend all at once.

4.1 Overview

This chapter is built on the construction and analysis of a particular optimization problem formulation, the convex separable problem. In Section 4.2, I construct a separable problem that captures the dilemma of a single decision maker who must allocate her attention among a collection of processes. In Section 4.3, I extend this basic problem formulation to incorporate various scenarios that involve the allocation of multiple attention units. The heart of the chapter is presented in Sections 4.4 and 4.5, in which I use the theory of convex separable optimization to model the allocation of attention among a set of concurrent processes. In analyzing the convex separable problem, I extract a priority measure and an attention-allocation measure for each process; these measures constitute the alarm outputs of my framework. In Section 4.6, I explain how we can engineer the problem formulation of Section 4.3 as a convex separable problem, for a large class of processes. I also demonstrate how the best attention-allocation strategy can depend on how far into the future we choose to look. Finally, in Section 4.7, I reflect on recent work on the allocation of scarce attention among concurrent processes, and I show how my framework advances the state of the art.

4.2 From single processes to multiple processes

Suppose that a decision maker is managing a particular, ongoing process. We can use the model presented in Chapter 3 to compute the utility[1] of applying a partial level of attention to this process. The notion of partial attention becomes especially relevant when we introduce other tasks or processes that can also benefit from this decision maker's attention. By representing the other processes as separate instances of the model of Chapter 3, we can begin to discuss meaningfully the problem of allocating a person's limited attention among multiple, concurrent processes (Figure 4.1).[2]

4.2.1 Notation

We assume that there are N independently operating processes; we index them using the variable j, with $j \in \{1, \ldots, N\}$. Our discussion uses the following notation:

1. Throughout the remainder of the dissertation, I use *utility* and *expected utility* synonymously.

2. *Person* and *people* can be interpreted more generally as *agent* and *agents,* respectively. I use the former terms for the sake of readability.

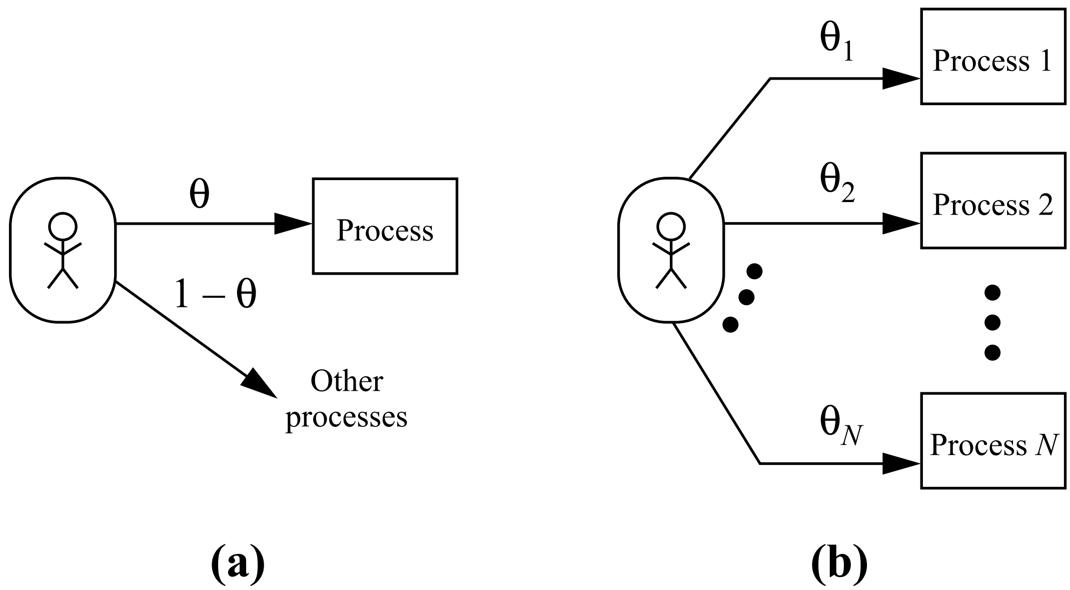


Figure 4.1. Implicit and explicit representations of competing processes. In (a), a decision maker devotes a partial level of attention θ to a process; the remaining attention $1 - \theta$ is devoted to unspecified, other processes. In (b), a decision maker allocates attention fractions $\theta_1, \ldots, \theta_N$ to corresponding processes $1, \ldots, N,$ where the θ_js add up to 1. Note that (b) can be viewed as an explicit rendition of (a), where the unspecified "other processes" are now specified. (In the remainder of the chapter, we speak of allocating *attention amounts* x_j instead of attention fractions θ_j.)

- x_j: The number of **attention units** assigned to process j
- $u_j(x_j)$: The **expected utility** of process j under an **attention amount** x_j
- T: The **attention horizon**

The attention amount x_j is applied to process j for a duration $T,$ starting from the present time $t = 0$. Each process j shares the same attention horizon T.

We are interested in computing and using the utility function $u_j(x_j)$. In addition to the attention amount x_j and the global parameter $T,$ we employ the following process-specific quantities in our calculation:

- κ_j: The **attention capacity** of process j, or the maximum number of attention units assignable to it
- $\mathbf{A}_j^{x_j}$, $\mathbf{q}_j^{x_j}$: The **transition matrix** and **reward vector,** respectively, for process j when it is receiving x_j units of attention
- α_j: The **discount rate** for rewards accumulated by process j (Section 3.2.3.2)
- i_j: The **current state** of process j (Section 3.2.1)

Now suppose that each $\kappa_j = 1$, that is, that up to 1 attention unit can be assigned to each process.[3] Then, we can compute $u_j(x_j)$ through a straightforward adaptation of the developments of Chapter 3, by regarding x_j as the attention fraction applied to process j. We begin the computation with the following fundamental quantities:

- $\mathbf{A}_j^0$, $\mathbf{q}_j^0$: The transition matrix and reward vector, respectively, of the default policy for process j (i.e., when it is receiving no attention)
- $\mathbf{A}_j^1$, $\mathbf{q}_j^1$: The transition matrix and reward vector, respectively, of the optimal policy for process j (i.e., when it is receiving 1 attention unit)
- $\mathbf{v}_j^1$: The **optimal-value vector** for process j (under 1 attention unit)

Here, we compute $\mathbf{A}_j^1$, $\mathbf{q}_j^1$, and $\mathbf{v}_j^1$ from $\mathbf{A}_j^0$, $\mathbf{q}_j^0$, and α_j using policy iteration (Sections 3.3 and 3.4). We substitute these quantities into Eqs. 3.27 and 3.28, obtaining the transition and reward matrices for the hybrid policy corresponding to the attention amount x_j:

$$\mathbf{A}_j^{x_j} = (1 - x_j)\mathbf{A}_j^0 + x_j \mathbf{A}_j^1 ; \tag{4.1}$$

$$\mathbf{q}_j^{x_j} = (1 - x_j)\mathbf{q}_j^0 + x_j \mathbf{q}_j^1 . \tag{4.2}$$

Next, we use the Laplace transform to compute the **total-value vector** that results from operating the hybrid policy for a time t, then operating the optimal policy thereafter (Section 3.5.3):

3. I relax this assumption in Section 4.3.

$$\boldsymbol{v}_j^{x_j}(s) = \frac{1}{s}((s+\alpha_j)\mathbf{I} - \mathbf{A}_j^{x_j})^{-1}\mathbf{q}_j^{x_j} + ((s+\alpha_j)\mathbf{I} - \mathbf{A}_j^{x_j})^{-1}\mathbf{v}_j^1; \tag{4.3}$$

$$\mathbf{v}_j^{x_j}(t) = \mathcal{L}^{-1}\{\boldsymbol{v}^{x_j}(s)\}. \tag{4.4}$$

Finally, we instantiate the solution of Equation 4.4 with the attention horizon T, and we extract the scalar component corresponding to the current state i_j:

$$u_j(x_j) = [\mathbf{v}_j^{x_j}(T)]_{i_j}, \tag{4.5}$$

where $[\mathbf{v}_j^{x_j}(T)]_{i_j}$ denotes the i_jth element of $\mathbf{v}_j^{x_j}(T)$.[4]

We use the notation $u_j(x_j)$ to emphasize this relationship between utility and applied attention, acknowledging that u_j also depends on the time-varying process state i_j, the process parameters $\mathbf{A}_j^0$, $\mathbf{q}_j^0$, $\mathbf{A}_j^1$, $\mathbf{q}_j^1$, $\mathbf{v}_j^1$, α_j, κ_j, and the global parameter T.

4.2.2 Independent processes with additive utilities

We are interested in the overall utility of the N processes. We assume that we are dealing with separate processes that operate independently; thus, changes in an attention amount x_j affect only the corresponding process utility $u_j(x_j)$. Furthermore, we assume that the overall utility comprises the individual process utilities in an additive manner. Using the symbol U to denote the overall utility, we can therefore write

$$U(x_1, \ldots, x_N) = \sum_{j=1}^{N} u_j(x_j). \tag{4.6}$$

Although the individual utilities $u_1, \ldots, u_N$ are independently determined by their corresponding attention amounts $x_1, \ldots, x_N$, they are, in general, *not* independent of one another. We shall see that the constraint on the total available attention introduces

4. To preserve clarity of exposition, I do not distinguish between *value* and *utility* in this dissertation. The equating of utility with value corresponds to the assumption of risk neutrality in decision analysis.

dependencies among the individual x_js, and therefore introduces dependencies among the individual u_js.

4.2.3 Basic optimization formulation

Using the notation of Sections 4.2.1 and 4.2.2 for describing individual processes, I now present an initial formulation of the multiple-process monitoring problem. A decision maker must allocate an amount of attention x_j to each process j, where $j = 1, \ldots, N$. The attention amounts are constrained to sum to 1. Each process j yields a utility $u_j(x_j)$. The overall utility $U(x_1, \ldots, x_N)$ is the sum of the component utilities $u_j(x_j)$. The objective is to maximize this overall utility:

$$\text{maximize } U(x_1, \ldots, x_N) \tag{4.7}$$

$$\text{subject to } \sum_{j=1}^{N} x_j = 1 \tag{4.8}$$

$$\text{and } 0 \le x_j \le 1 \qquad \text{for } j = 1, \ldots, N, \tag{4.9}$$

where $U(x_1, \ldots, x_N)$ is given by Equation 4.6.

I present methods for solving Eqs. 4.6 through 4.9 in Section 4.4. But let us first extend our current problem formulation to handle a larger class of resource-constrained, process-monitoring problems.

4.3 Allocation of multiple attention units

Thus far, we have thought of the multiple-process problem as the optimal partitioning of a single person's attention. However, by manipulating the right-hand side of Equation 4.8, we can talk about partitioning the attention of more than one person. We would also like to talk about allocating more than one person to a process. For example, a certain process might require the attention of two people to operate optimally.

In this section, I generalize our existing framework to incorporate such possibilities. These generalizations are simple from a mathematical-modeling standpoint, but the semantics of the revised framework merit careful consideration.

4.3.1 Incorporation of different resource constraints

Suppose that C people are available to manage a set of N tasks. We can generalize our current problem formulation by substituting C into the right-hand side of Equation 4.8. Larger values of C allow more favorable attention allocations, which in turn permit greater overall utility.

When $C = N$, the only solution to the optimization problem is to assign $x_j = 1$ for each process j. Problems arise when $C > N$—when the amount of available attention resource exceeds the combined resource capacities of all the processes. This technical difficulty can be resolved easily with an additional variable x_0 that accounts for any unused attention ($0 \leq x_0 \leq C$); we modify the left-hand side of Equation 4.8 to incorporate x_0.

4.3.1.1 Attention units versus people units

The optimization-problem formulation treats C as a fungible, continuous resource that is partitioned, into fractional amounts as necessary, to achieve maximum overall utility. It can therefore be problematic to speak of C as an available number of people, since people typically come in discrete units. Therefore, I refer to C as an *attention amount,* measured in units of attention, or *attention units*—rather than people units. The magnitude of C conveys the attention resources of that equivalent number of people, with the understanding that such resources can be partitioned continuously.

4.3.1.2 Meta-level decision maker

Now that we have allowed multiple attention units to be allocated, we must clarify the identity of the decision maker. For $C = 1$, we described a single decision maker striving to multitask among several processes. However, this story does not adapt readily to situations where $C \neq 1$, because the allocation of multiple attention units is performed by one decision maker. Thus, although the quantity C represents available human activity, it is easier to think of C as a fungible resource that is allocated by an intelligent decision maker at the meta-level. This situation is akin to a manager partitioning and assigning a large workforce to different tasks.

4.3.2 Incorporation of different attention capacities

Thus far, we have assumed that it takes the full attention of one person, or 1 attention unit, to operate a process optimally. However, we can imagine a process that requires multiple units of attention to operate optimally. We can accommodate such processes by relaxing the previous assumption that the attention capacities κ_j all equal 1.

We must decide exactly what it means for a process to receive arbitrary amounts of attention. Suppose a process j operates optimally when it receives κ_j units of attention. We can construct hybrid default-optimal matrices for x_j ($\leq \kappa_j$) units of attention according to the effective attention fraction x_j/κ_j. We start with the following quantities:

- $\mathbf{A}_j^0$, $\mathbf{q}_j^0$: The transition matrix and reward vector, respectively, of the default policy for process j (i.e., when it is receiving no attention)
- $\mathbf{A}_j^{\kappa_j}$, $\mathbf{q}_j^{\kappa_j}$: The transition matrix and reward vector, respectively, of the optimal policy for process j (i.e., when it is receiving κ_j units of attention)
- $\mathbf{v}_j^{\kappa_j}$: The optimal-value vector for process j (under κ_j units of attention)

For an attention amount x_j, we compute the transition and reward matrices for the hybrid default-optimal policy under a partial level of attention x_j/κ_j (Section 3.5.2):

$$\mathbf{A}_j^{x_j} = \left(1 - \frac{x_j}{\kappa_j}\right)\mathbf{A}_j^0 + \frac{x_j}{\kappa_j}\mathbf{A}_j^{\kappa_j}; \tag{4.10}$$

$$\mathbf{q}_j^{x_j} = \left(1 - \frac{x_j}{\kappa_j}\right)\mathbf{q}_j^0 + \frac{x_j}{\kappa_j}\mathbf{q}_j^{\kappa_j}. \tag{4.11}$$

We then compute $u_j(x_j)$ in a manner analogous to Eqs. 4.3 through 4.5, substituting $\mathbf{A}_j^{\kappa_j}$, $\mathbf{q}_j^{\kappa_j}$, and $\mathbf{v}_j^{\kappa_j}$ for $\mathbf{A}_j^1$, $\mathbf{q}_j^1$, and $\mathbf{v}_j^1$, respectively:

$$\boldsymbol{v}_j^{x_j}(s) = \frac{1}{s}((s+\alpha_j)\mathbf{I} - \mathbf{A}_j^{x_j})^{-1}\mathbf{q}_j^{x_j} + ((s+\alpha_j)\mathbf{I} - \mathbf{A}_j^{x_j})^{-1}\mathbf{v}_j^{\kappa_j}; \tag{4.12}$$

$$\mathbf{v}_j^{x_j}(t) = \mathcal{L}^{-1}\{\boldsymbol{v}^{x_j}(s)\}; \tag{4.13}$$

$$u_j(x_j) = [\mathbf{v}_j^{x_j}(T)]_{i_j}. \tag{4.14}$$

Although our use of the notation $u_j(x_j)$ emphasizes the relationship between utility and attention allocation, it is important to remember that u_j also depends on the current state i_j, the process parameters $\mathbf{A}_j^0$, $\mathbf{q}_j^0$, $\mathbf{A}_j^{x_j}$, $\mathbf{q}_j^{\kappa_j}$, $\mathbf{v}_j^{\kappa_j}$, α_j, κ_j, and the global parameter T. Figure 4.2 illustrates the dynamic determinants of u_j.

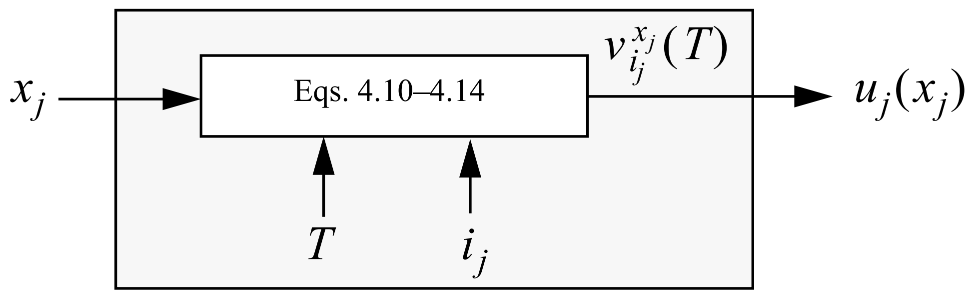


Figure 4.2. Determinants of process utility.
The utility of process j, u_j, depends on the attention amount x_j and on numerous of other quantities, including i_j, the current state of process j, and T, the attention horizon of interest. The other determinants of u_j—κ_j, $\mathbf{A}_j^0$, $\mathbf{q}_j^0$, $\mathbf{A}_j^{x_j}$, $\mathbf{q}_j^{\kappa_j}$, $\mathbf{v}_j^{\kappa_j}$, and α_j—are intrinsic process parameters, and are not shown explicitly. The total expected utility from these inputs is denoted by $v_{i_j}^{x_j}(T)$; we use $u_j(x_j)$ to denote this quantity to emphasize our interest in how this utility changes with the attention amount x_j.

4.3.3 Revised problem formulation

The added notions of variable attention constraints and variable attention capacities permit the following generalization of the basic problem formulation of Section 4.2. We are given C attention units, and we must optimally allocate x_j of them to each process j, for $j = 1, \ldots, N$. Each process j has an attention capacity κ_j and yields a utility $u_j(x_j)$ that is obtained from Eqs. 4.10 through 4.14. The overall utility $U(x_1, \ldots, x_N)$ to be maximized is the sum of the component utilities $u_j(x_j)$. We write

$$\text{maximize } \sum_{j=1}^{N} u_j(x_j) \tag{4.15}$$

$$\text{subject to } \sum_{j=1}^{N} x_j = C \tag{4.16}$$

$$\text{and } \; 0 \le x_j \le \kappa_j \qquad \text{for } j = 1, \ldots, N. \tag{4.17}$$

Note that problems arise when the total available attention C exceeds the total attention capacity $\sum_{j=1}^{N} \kappa_j$. We can remedy such scenarios by introducing an additional variable x_0, to which all excess avilable attention units are allocated, and modifying the left-hand side of Equation 4.16 accordingly. (Note that $0 \le x_0 \le C$.) Figure 4.3 depicts the resulting optimization scenario. Because the need for x_0 arises only when there is no meaningful constraint on the available attention, I henceforth assume that $0 \le C \le \sum_{j=1}^{N} \kappa_j$.

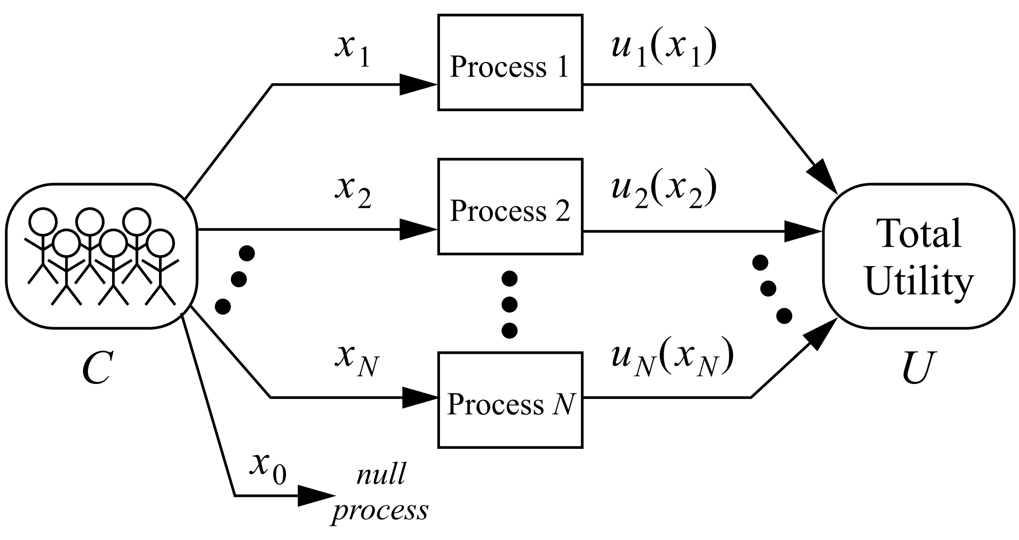


Figure 4.3. Distribution of attention among concurrent processes.
A limited amount of attention C is to be distributed among N processes. When process j receives an attention amount x_j, it yields an expected utility $u_j(x_j)$. The total utility U depends on the values assigned to $x_1, \ldots, x_N$, and is equal to the sum of the process utilities $u_j(x_j)$ for $j = 1, \ldots, N$. Unused attention resources are incorporated into the special variable x_0, where $0 \le x_0 \le C$. The x_js must add up to C.

4.3.4 Overview of solution methodology

Eqs. 4.15 through 4.17 typify a **separable optimization problem:** The constrained resource (the available attention) separates into a sum of components $x_1, \ldots, x_N$, and the objective function U separates into a sum of functions u_j of the respective components x_j. This separability permits efficient solutions to what would otherwise be an intractable search for optimal solutions in the joint space of $x_1, \ldots, x_N$.

Our framework for managing concurrent process relies on the analysis of **convex separable problems**—separable problems with additional assumptions about the form of the utility functions. In Section 4.4, I begin our investigation of convex separable problems by considering only the equality constraint (4.16). Then, in Section 4.5, I incorporate the inequality constraints (4.17). Finally, in Section 4.6, I describe practical methods for modeling the utility of a process under various levels of attention. We are particularly interested in Markov decision processes whose utility functions satisfy the assumptions of the convex separable problem.

4.4 Convex separable problems

We now explore the solution of convex separable problems with a single equality constraint [Luenberger, 1984]. Our problem is this:

$$\text{maximize } \sum_{j=1}^{N} u_j(x_j) \tag{4.18}$$

$$\text{subject to } \sum_{j=1}^{N} x_j = C \tag{4.19}$$

by strategically choosing $x_1, \ldots, x_N$. We assume that each $u_j(x_j)$ is increasing: that u_j increases as x_j increases. Furthermore, we assume that each $-u_j(x_j)$ is strictly **convex:**

$$\partial^2 u_j / \partial x_j < 0 \text{ for all } x_j.^5 \tag{4.20}$$

In other words, u_j increases with diminishing returns as x_j increases, for $j = 1, \ldots, N$.

We can rewrite (4.18) and (4.19) as

$$\text{maximize } U(\mathbf{x}) \tag{4.21}$$

$$\text{subject to } h(\mathbf{x}) = 0, \tag{4.22}$$

where $\mathbf{x} = [x_1 \ldots x_N]^{\mathrm{T}}$, $U(\mathbf{x}) = \sum_{j=1}^{N} u_j(x_j)$, and $h(\mathbf{x}) = (\sum_{j=1}^{N} x_j) - C$. When conveying basic results from optimization theory, we use the notation of (4.21) and (4.22); when we exploit separability, we recruit the more specific form of (4.18) and (4.19).

4.4.1 First-order necessary conditions

Suppose that a point $\mathbf{x}$ satisfies the constraint $h(\mathbf{x}) = 0$ and is a local extremum (maximum or minimum) of the objective function $U(\mathbf{x})$, where both h and U have continuous first-order partial derivatives. Classic optimization theory asserts the following first-order necessary condition: For any such point **x,** there exists a real number λ such that

$$\nabla U(\mathbf{x}) - \lambda \nabla h(\mathbf{x}^*) = \mathbf{0}. \tag{4.23}$$

Equation 4.23, together with the constraint (4.22), is a system of $N + 1$ equations in $N + 1$ unknowns (comprising $\mathbf{x}$ and λ). Instantiated with the separable problem (4.18) and (4.19), this system of equations becomes

$$\lambda = \frac{\partial}{\partial x_j} u_j(x_j) \qquad \text{for } j = 1, \ldots, N; \tag{4.24}$$

$$\sum_{j=1}^{N} x_j = C. \tag{4.25}$$

5. Setting this condition is equivalent to asserting that each $u_j(x_j)$ is strictly concave. The optimization literature typically adheres to a cost-minimization framework with convexity assumptions; here, I adopt a utility-maximization framework with concavity assumptions. I use the term *convex,* however, to remain consistent with the language of the optimization literature.

The parameter λ is called a **Lagrange multiplier** for historical reasons. I discuss the meaning and significance of λ in Section 4.4.5.

4.4.2 Second-order sufficient conditions

Suppose that $\mathbf{x}^*$ and λ^* satifies the system (4.22) and (4.23). If h and U have continuous second-order partial derivatives, then $U(\mathbf{x}^*)$ is guaranteed to be a strict local maximum under the following second-order condition: The matrix

$$\mathbf{L}(\mathbf{x}^*) = \nabla^2 U(\mathbf{x}^*) - \lambda^* \nabla^2 h(\mathbf{x}^*) \tag{4.26}$$

is negative definite on the plane $M = \{\mathbf{y}\mathbf{:}\ \nabla h(\mathbf{x}^*)\,\mathbf{y} = \mathbf{0}\}$, that is, $\mathbf{y}^{\mathrm{T}}\mathbf{L}(\mathbf{x}^*)\mathbf{y} > 0$, for $\mathbf{y} \in M$, $\mathbf{y} \neq \mathbf{0}$.

Note that, for the separable problem (4.18) and (4.19), $\nabla^2 h(\mathbf{x})$ vanishes for all $\mathbf{x}$, and $\nabla^2 U(\mathbf{x})$ reduces to a diagonal matrix. Furthermore, the convexity requirement (4.20) causes the diagonal elements of $\nabla^2 U(\mathbf{x})$ to be negative. From these observations, we see that the matrix $\mathbf{L}(\mathbf{x}^*)$ is diagonal with negative elements, which automatically renders it negative definite on the entire space of $\mathbf{x}^*$, and therefore on M. Thus we can be confident that any solution $\mathbf{x}^*, \lambda^*$ to the system (4.24) and (4.25) locally maximizes the overall utility $U(\mathbf{x}) = \sum_{j=1}^{N} u_j(x_j)$ under the constraint $\sum_{j=1}^{N} x_j = C$.

4.4.3 Local duality

We have established that solving the constrained optimization problem involves solving a system of $N + 1$ equations with $N + 1$ unknowns—the N-component vector $\mathbf{x}$ and the Lagrange multiplier λ. Thus, any solution $\mathbf{x}^*$ to the optimization problem has an associated Lagrange multiplier λ^*. It turns out that, under certain local convexity assumptions, we can formulate an unconstrained, dual optimization problem in which the parameter λ is the primary unknown. Any solution λ^* to the dual problem would then have an associated point $\mathbf{x}^*$ that we can determine by invoking the first-order conditions (4.23) with $\lambda = \lambda^*$.

Although duality plays an important role in the theory of optimization, we are more interested in its practical ramifications. Specifically, the dual problem is often easier to solve than is the original problem. Let us examine the theory of duality to see how we can use it to solve the convex separable problem (4.18) and (4.19) efficiently.

4.4.3.1 Local-duality theorem

Suppose that the constrained problem

$$\text{maximize } U(\mathbf{x}) \tag{4.27}$$

$$\text{subject to } h(\mathbf{x}) = 0 \tag{4.28}$$

has a local solution at $\mathbf{x}^*$ with corresponding value $r^* = U(\mathbf{x}^*)$ and Lagrange multiplier λ^*. Suppose also that the matrix $\mathbf{L}(\mathbf{x}^*) = \nabla^2 U(\mathbf{x}^*) - \lambda^* \nabla^2 h(\mathbf{x}^*)$ is negative definite. Let the dual function ϕ be defined according to

$$\phi(\lambda) = \max_{\mathbf{x}}[U(\mathbf{x}) - \lambda h(\mathbf{x})]. \tag{4.29}$$

Then, the local-duality theorem asserts that the unconstrained dual problem

$$\text{minimize } \phi(\lambda) \tag{4.30}$$

has a local solution at λ^* with corresponding value $r^* = \phi(\lambda^*) = U(\mathbf{x}^*)$.

4.4.3.2 Exploitation of local duality

A prominent obstacle to putting local duality to work is the task of computing the dual function $\phi(\lambda)$. Performing this computation involves finding, for a given λ, the point $\mathbf{x}$ that maximizes the quantity $U(\mathbf{x}) - \lambda h(\mathbf{x})$. Setting the gradient of this quantity to 0 yields the following system of N equations in N unknowns (with λ held constant):

$$\nabla U(\mathbf{x}) - \lambda \nabla h(\mathbf{x}) = \mathbf{0}, \tag{4.31}$$

which is identical to the set of first-order necessary conditions in Equation 4.23. Solving this system yields a locally unique solution, which we designate as $\mathbf{x}(\lambda)$. We obtain the dual function, then, by substituting $\mathbf{x}(\lambda)$ into Equation 4.29:

$$\phi(\lambda) = U(\mathbf{x}(\lambda)) - \lambda h(\mathbf{x}(\lambda)). \tag{4.32}$$

In general, our ability to exploit local duality depends primarily on how easily $\mathbf{x}(\lambda)$ is obtained. Given $\mathbf{x}(\lambda)$, however, it turns out that we do not need to compute $\phi(\lambda)$ at all. To see why, consider what happens when we attempt to minimize $\phi(\lambda)$. Differentiating Equation 4.32 with respect to λ yields, by the chain rule,

$$\phi'(\lambda) = [\nabla U(\mathbf{x}(\lambda)) - \lambda \nabla h(\mathbf{x}(\lambda))]\mathbf{x}'(\lambda) + h(\mathbf{x}(\lambda)),$$

where the gradient ∇ is taken in the space of **x.** Since $\mathbf{x}(\lambda)$ is defined as the point that maximizes $U(\mathbf{x}(\lambda)) - \lambda h(\mathbf{x}(\lambda))$, the gradient of this expression vanishes, and $\phi'(\lambda)$ reduces to $h(\mathbf{x}(\lambda))$. To solve the constrained optimization problem (4.27) and (4.28), then, we take the following steps:

- Compute $\mathbf{x}(\lambda)$ by solving $\nabla U(\mathbf{x}) - \lambda \nabla h(\mathbf{x}) = \mathbf{0}$.
- Solve the dual problem (4.30) by solving $h(\mathbf{x}(\lambda)) = 0$.
- For each root λ^*, compute $\mathbf{x}^* = \mathbf{x}(\lambda^*)$.

Remember that this use of local duality assumes, for each solution $\mathbf{x}^*, \lambda^*$, that the matrix $\mathbf{L}(\mathbf{x}^*) = \nabla^2 U(\mathbf{x}^*) - \lambda^* \nabla^2 h(\mathbf{x}^*)$ is negative definite (see Section 4.4.2).

4.4.3.3 Solution of convex separable problems

Let us invoke the procedure given previously to solve the convex separable problem:

$$\text{maximize } \sum_{j=1}^{N} u_j(x_j) \tag{4.33}$$

$$\text{subject to } \sum_{j=1}^{N} x_j = C. \tag{4.34}$$

Our first step is to compute $\mathbf{x}(\lambda)$. We instantiate the first-order conditions and obtain the following for $j = 1, \ldots, N$:

$$\lambda = \frac{\partial}{\partial x_j} u_j(x_j), \tag{4.35}$$

which can be rewritten as

$$\lambda = \lambda_j(x_j), \tag{4.36}$$

where the function $\lambda_j(x_j)$ is defined as the right-hand side of Equation 4.35. The convexity assumption $\partial^2 u_j / \partial x_j < 0$ implies that $\lambda_j(x_j)$ is monotonic and therefore invertible. Thus, we can invert Equation 4.36 to obtain the desired components of $\mathbf{x}(\lambda)$:

$$x_j(\lambda) = \lambda_j^{-1}(\lambda) \qquad \text{for } j = 1, \ldots, N. \tag{4.37}$$

Armed with $\mathbf{x}(\lambda)$, we construct and solve the constraint $h(\mathbf{x}(\lambda)) = 0$, or

$$\left(\sum_{j=1}^{N} x_j(\lambda)\right) - C = 0. \text{[6]} \tag{4.38}$$

Finally, for a solution λ^*, we use Equation 4.37 to determine the corresponding $\mathbf{x}^*$.

4.4.4 Global optimality

We have found that the required conditions for local duality are satisfied automatically by the convexity assumptions (4.20). It turns out that these convexity assumptions are stronger than what is needed to employ local duality, and that these stronger assumptions lend a stronger interpretation to any solution $\mathbf{x}^*, \lambda^*$ obtained through the local-duality procedure presented in Section 4.4.3.

Equation 4.20 implies that the objective function $U(\mathbf{x}) = \sum_{j=1}^{N} u_j(x_j)$ is concave.[7] Furthermore, the constraint $h(\mathbf{x})$ is affine (linear plus a constant). In general, these two conditions, taken together, guarantee that any locally optimal solution $\mathbf{x}^*, \lambda^*$ to the optimization problem (4.21) and (4.22) is globally optimal. Thus, we can be confident that any solution we obtain from solving (4.37) and (4.38) is a globally optimal solution.

---

6. People typically employ numerical methods to solve this equation. An analytic solution can be achieved only when the functions $u_j(x_j)$ are formulated conveniently.

7. Maximizing the concave objective function $U(\mathbf{x})$ is equivalent to minimizing the convex objective function $-U(\mathbf{x})$.

4.4.5 Sensitivity

Given a maximizing point $\mathbf{x}^*$, what can we say about its associated Lagrange multiplier λ? In this subsection, we find that λ can be interpreted mathematically as a local sensitivity parameter, and economically as a marginal price. We shall also see, in the specific case of the convex separable problem, that λ can be interpreted as an equilibrium price.

4.4.5.1 Sensitivity theorem

Consider the family of constrained problems:

$$\text{maximize } U(\mathbf{x}) \tag{4.39}$$

$$\text{subject to } h(\mathbf{x}) = c\,. \tag{4.40}$$

Suppose that, for $c = 0$, there is a local solution $\mathbf{x}^*$, λ that satisfies the second-order sufficiency conditions for a strict local maximum. Then, there exists an $\mathbf{x}(c)$ that is well-defined and continuous over some interval containing 0, such that $\mathbf{x}(c)$ is a local maximum of (4.39) and (4.40), and $\mathbf{x}(0) = \mathbf{x}^*$. Furthermore, the sensitivity theorem asserts that

$$\left.\frac{\partial}{\partial c} U(\mathbf{x}(c))\right|_{c=0} = \lambda\,. \tag{4.41}$$

Equation 4.41 states that a small increase in resource ∂c increases the quantity $U(\mathbf{x}(c))$ by an amount $\lambda \partial c$. The parameter λ can thus be interpreted as a **marginal price**—a price associated with small changes in the constraint. Since the optimization solution varies with the total resource C, we may also speak of a quantity $\lambda(C)$ that measures the local sensitivity of overall utility to variations in total resource in the neighborhood of C.

4.4.5.2 Equilibrium prices

We can glean more insight into the Lagrange multiplier λ for the specific case of the convex separable problem. Recall that each process j has an associated utility function $u_j(x_j)$ that increases in x_j with diminishing returns, and an associated function $\lambda_j(x_j)$ that we defined in terms of $u_j(x_j)$:

$$\lambda_j(x_j) = \frac{\partial}{\partial x_j} u_j(x_j) \qquad \text{for } j = 1, \ldots, N. \tag{4.42}$$

Equation 4.42 states that the quantity $\lambda_j(x_j)$ measures the sensitivity of the individual utility u_j to variations in resource allocated to process j in the neighborhood of x_j. Because $u_j(x_j)$ is increasing and concave, $\lambda_j(x_j)$ is a positive quantity that decreases in magnitude as x_j increases. In other words, additional resource increments result in progressively smaller increments in process utility, for each process j. We can thus think of λ_j as a **local marginal price**—that is, as a marginal price local to the individual process j.

The global constraint (4.34) introduces a mutual dependence among the local marginal prices. Specifically, for any optimal allocation $x_1^*, \ldots, x_N^*$, the first-order necessary conditions require that the corresponding local marginal prices $\lambda_1^*(x_1^*), \ldots, \lambda_N^*(x_N^*)$ all be equal. Combining (4.35) with (4.42), we obtain

$$\lambda = \lambda_j(x_j^*) \qquad \text{for } j = 1, \ldots, N. \tag{4.43}$$

In other words, we can interpret the global sensitivity parameter λ as an **equilibrium price:** a marginal price that holds across all processes under an optimal attention allocation $\mathbf{x}^*$**.** Stated from the dual point of view, each local marginal price must be set to the global equilibrium price if the set of x_j^*s is to optimize the overall utility and simultaneously satisfy the resource constraint.

4.4.6 Example: Portfolio management

We have seen how duality enables a straightforward procedure for solving a convex separable problem. Let us apply this procedure to the **portfolio manager's problem.** An investor hires four consultants to manage a portfolio for a period of time. The portfolio comprises three funds. From previous experience with similar funds, she assesses, for each fund j, the expected profit u_j (quantified in millions of dollars) resulting from an effective number of consultants x_j managing the fund (Figure 4.4):

$$u_1(x_1) = \ln(x_1 + 1)$$
$$u_2(x_2) = 0.7x_2 - 0.03(x_2)^2$$
$$u_3(x_3) = 1.5(1 - e^{-x_3})$$

She wishes to determine how much of the consultants' combined attention should be allocated to each fund. Her goal is to maximize the total expected profits from this portfolio.

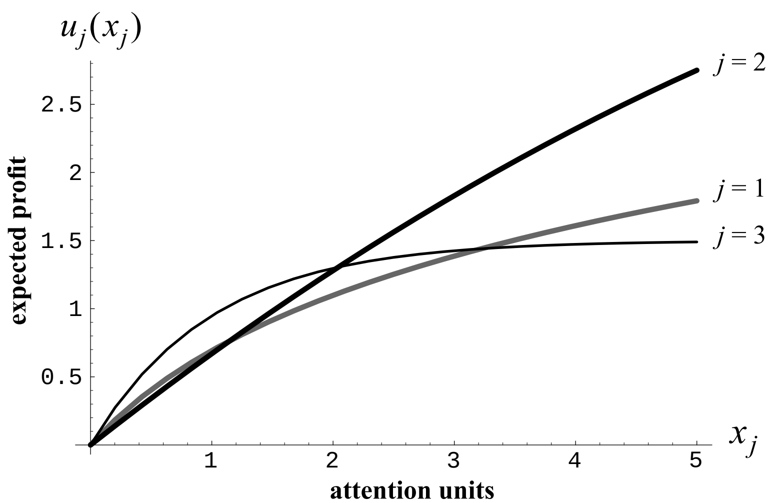


Figure 4.4. Utility functions (portfolio example).
Illustrated for each of the three funds are the expected profits, in millions of dollars, as a function of the effective number of allocated consultants.

After confirming that we have indeed a convex separable problem,[8] our first step is to compute the components of $\mathbf{x}(\lambda)$. We compute $\lambda_j(x_j)$ by differentiating u_j, obtaining

$$\lambda_1(x_1) = \frac{1}{x_1 + 1},$$
$$\lambda_2(x_2) = 0.7 - 0.06x_2,$$
$$\lambda_3(x_3) = 1.5e^{-x_3}.$$

8. Although $u_2(x_2)$ is concave, it does not increase for all x_2. However, this fact does not present a problem for our analysis.

We then invert these relations to obtain the desired functions $x_j(\lambda_j)$. Equating the λ_js as λ and applying the constraint yields the equation $x_1(\lambda) + x_2(\lambda) + x_3(\lambda) = 4$, which is satisfied by $\lambda = 0.564829$. We then compute $x_j^* = x_j(\lambda)$ for each fund j:

$$x_1^* = 0.770448 \qquad x_2^* = 2.25285 \qquad x_3^* = 0.976698,$$

which yields a total profit of $u_1(x_1^*) + u_2(x_2^*) + u_3(x_3^*) = 2.93144$ million dollars. Figure 4.5 illustrates this solution graphically: The relationships between each λ_j and x_j are plotted, and the horizontal line indicates the marginal price at the optimal solution.

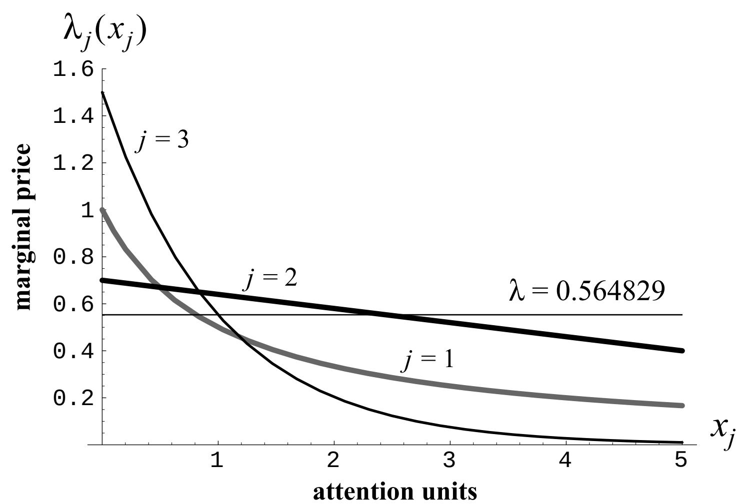


Figure 4.5. Marginal prices (portfolio example).
For each of the three funds, the marginal price decreases as the effective number of consultants increases. The solution to the optimization problem is represented by the horizontal line and its points of intersection with the marginal-profit curves. At the solution, the individual marginal profits are equal, such that the corresponding attention units sum to 4.

Let us briefly illustrate the role of λ as a sensitivity parameter. We can repeat the preceding analysis for constraints that deviate slightly from the original example. The results of one such experiment are shown in Table 4.1. Note that, for these small deviations in total resource, the actual incremental profit is approximated closely by the original equilibrium price of $\lambda = 0.564829$ million dollars per consultant.

C	$U(\mathbf{x}(c))$	$\Delta U/\Delta C$
3.95	2.90284	0.565988
3.97	2.91418	0.565524
3.99	2.92549	0.565061
4.01	2.93679	0.564597
4.03	2.94807	0.564134
4.05	2.95932	0.563670

Table 4.1. Sensitivity experiment (portfolio example).
The effects of small variations in the total resource amount C are shown. Note that the optimized profit $U(\mathbf{x}(c))$ increases with C. Furthermore, the normalized profit response $\Delta U/\Delta C$ closely approximates the equilibrium price of $\lambda = 0.564829$ at $C = 4$.

The procedure of Section 4.4.3.3 breaks down for more extreme values of C. For example, when $C = 1$, the procedure yields the "optimal" solution $x_1^* = 0.409466$, $x_2^* = -0.158142$, $x_3^* = 0.748676$. Unfortunately, our manager cannot allocate negative consultant equivalents. Thus, we need to amend our current framework by incorporating boundary conditions such as $x_2 \geq 0$.

4.5 Incorporation of boundary conditions

With a few technical modifications, we can use the methods of the previous section to solve convex separable problems that stipulate boundary conditions on the resource allocations. Let us revisit our original problem formulation:

$$\text{maximize} \sum_{j=1}^{N} u_j(x_j) \tag{4.44}$$

$$\text{subject to} \sum_{j=1}^{N} x_j = C \tag{4.45}$$

$$\text{and } 0 \leq x_j \leq \kappa_j \qquad \text{for } j = 1, \ldots, N, \tag{4.46}$$

where, for each process j, we have $\partial^2 u_j / \partial x_j < 0$ for all x_j.

It is possible to extend the theoretical framework of Section 4.4 to incorporate inequality constraints. Such an extension, however, requires additional parameters and notation, and does not move us closer to developing efficient algorithms for solving (4.44) through (4.46). Rather than explore this more general theory, let us investigate two specific approaches for dealing with the boundary conditions (4.46).

4.5.1 Methods for dealing with boundaries

The requirement $0 \le x_j \le \kappa_j$ means that it is impossible to allocate less than 0 attention units or more than κ_j attention units to process j. Thus, we must never allow such an allocation to occur during any optimization procedure. Whenever there is an attempt to allocate negative attention units to a process j, we must enforce the minimum allocation of 0 attention units. Similarly, whenever there is an attempt to allocate attention beyond the process's attention capacity, we must enforce the maximum attention allocation κ_j.

The following methods for handling boundary conditions assume that $0 \le C < \sum_{j=1}^{N} \kappa_j$; larger values of C result in a degenerate problem with no effective resource constraint.

4.5.1.1 Method 1: Elimination of boundary processes

One way to satisfy the boundary conditions is first to eliminate processes that do not meet those boundary conditions, then to reoptimize the reduced set of processes. An algorithm for this strategy is presented in the following function, BOUNDED-OPTIMIZE. There are two input parameters: P, the set of processes to be optimized, and C, the number of available attention units. The subroutine OPTIMIZE(P, C) performs the procedure in Section 4.4.3.3 and returns a solution pair $(\mathbf{x}, \lambda)$, where $\mathbf{x}$ denotes a set of assignments of values to variables. The initial call is BOUNDED-OPTIMIZE$(\{1, \ldots, N\}, C)$.

BOUNDED-OPTIMIZE(P, C):

- $\mathbf{z} := \varnothing$
- **Loop**
  1. $(\mathbf{x}, \lambda) :=$ OPTIMIZE(P, C)
  2. $Q := \{j \in P \mid x_j < 0\}$
  3. **If** $Q = \varnothing$ **then break** `(*exit loop *)`
  4. $\mathbf{z} := \mathbf{z} \cup \{x_j \leftarrow 0 \mid j \in Q\}$
  5. $P := P - Q$ `(* set difference *)`
- **Loop**
  1. $R := \{j \in P \mid x_j > \kappa_j\}$
  2. **If** $R = \varnothing$ **then break** `(* exit loop *)`
  3. $\mathbf{z} := \mathbf{z} \cup \{x_j \leftarrow \kappa_j \mid j \in R\}$
  4. $C := C - \sum_{j \in R} \kappa_j$
  5. $P := P - R$ `(* set difference *)`
  6. $(\mathbf{x}, \lambda) :=$ OPTIMIZE(P, C)
- $\mathbf{x} := \mathbf{x} \cup \mathbf{z}$
- **Return** $(\mathbf{x}, \lambda)$

4.5.1.2 Method 2: Redefinition of the dual function

By slightly modifying the inverse marginal functions $x_j(\lambda)$, we can adapt the procedure of Section 4.4.3.3 to handle the boundary conditions $0 \le x_j \le \kappa_j$. This method involves computing and using the **boundary prices** $\lambda_j(0)$ and $\lambda_j(\kappa_j)$; note that these quantities are the maximum and minimum values, respectively, of the marginal price for process j. An attempt to overallocate corresponds to an equilibrium price less than $\lambda_j(\kappa_j)$, whereas an equilibrium price that exceeds $\lambda_j(0)$ leads to unintended negative allocations. The modified procedure is as follows:

- For each process j, compute the function

$$x_j(\lambda) = \begin{cases} \kappa_j & \lambda < \lambda_j(\kappa_j) \\ \lambda_j^{-1}(\lambda) & \lambda_j(\kappa_j) \le \lambda \le \lambda_j(0), \\ 0 & \lambda > \lambda_j(0) \end{cases} \tag{4.47}$$

where $\lambda_j(x_j) = \dfrac{\partial}{\partial x_j} u_j(x_j)$.

- Solve for λ:

$$\sum_{j=1}^{N} x_j(\lambda) = C. \tag{4.48}$$

- Using the solution λ^*, compute $x_j^* = x_j(\lambda^*)$ for each process j.

Note that, in general, the equilibrium price λ^* matches the individual marginal prices $\lambda_j(x_j^*)$ for only those processes in which $0 < x_j^* < \kappa_j$.

4.5.2 Example: A family of portfolio problems

Let us revisit the portfolio manager's problem, this time including boundary conditions in our analysis. We were given $u_1(x_1) = \ln(x_1 + 1)$, $u_2(x_2) = 0.7x_2 - 0.03(x_2)^2$, and $u_3(x_3) = 1.5(1 - e^{-x_3})$. Suppose that each fund can be managed by up to three consultants: $\kappa_1 = \kappa_2 = \kappa_3 = 3$. Then, our goal is to maximize the sum of the $u_j(x_j)$s, subject to $x_1 + x_2 + x_3 = C$ and $0 \le x_j \le 3$, where C is the total number of available attention units.

Using either method from Section 4.5.1, we can verify that the solution for $C = 1$ is $x_1^* = 0.318239$, $x_2^* = 0$, $x_3^* = 0.681761$; this solution has an associated equilibrium price $\lambda = 0.758588$ that is shared by only the first and third funds. More interesting is how the solution varies with the resource constraint C: Figure 4.6 illustrates the optimal attention allocations for values of C ranging from 0 to 5. The corresponding marginal prices and optimized total profits are shown in Figures 4.7 and 4.8, respectively. We can also compute the fractional optimal allocations, $\varphi_j^*(C) \equiv x_j^*(C)/C$, which provide additional insight into the solution (Figure 4.9).

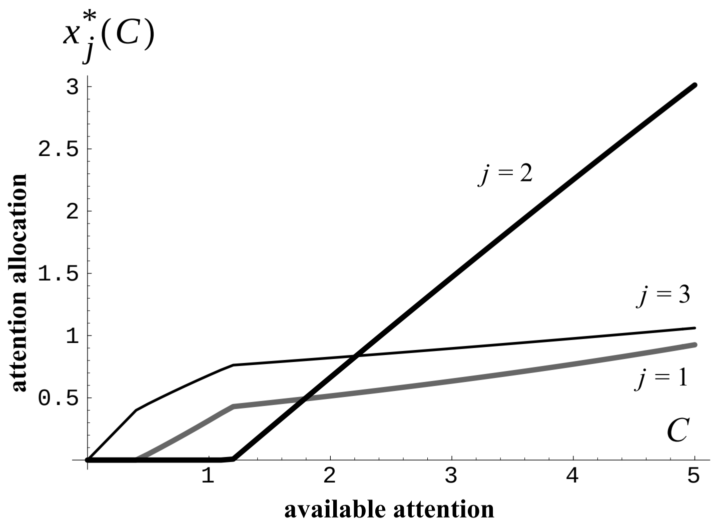


Figure 4.6. Optimal attention allocations (portfolio example).
Graphed are the optimal attention allocations x_1^*, x_2^*, x_3^* for the portfolio management problem as the available number of consultant attention units C is varied between 0 and 5.

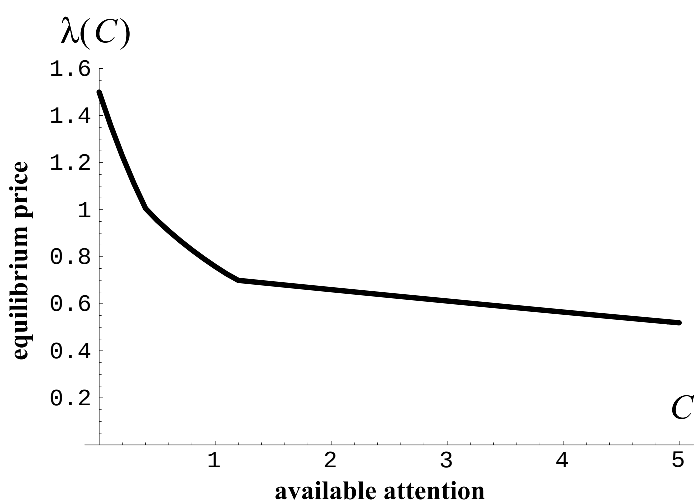


Figure 4.7. Equilibrium prices (portfolio example).
The equilibrium price λ decreases as more consultant attention resources C are made available to the portfolio management problem.

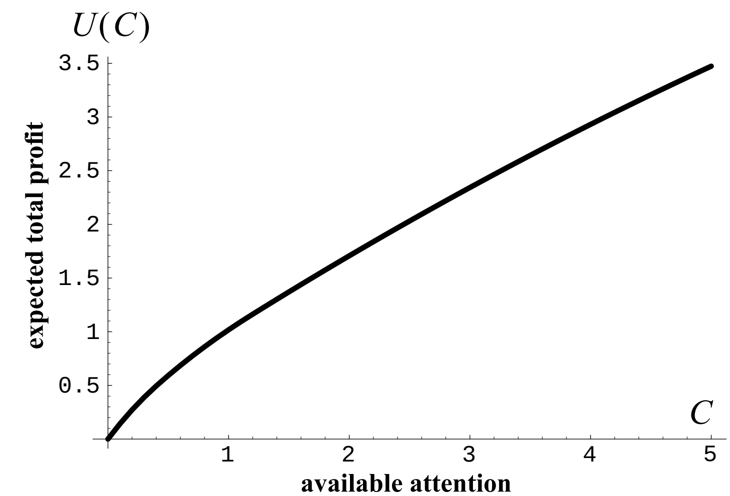


Figure 4.8. Total utility (portfolio example).

An increase in consultant resources C increases the maximum attainable total profit $U(C) = u_1(x_1^*(C)) + u_2(x_2^*(C)) + u_3(x_3^*(C))$.

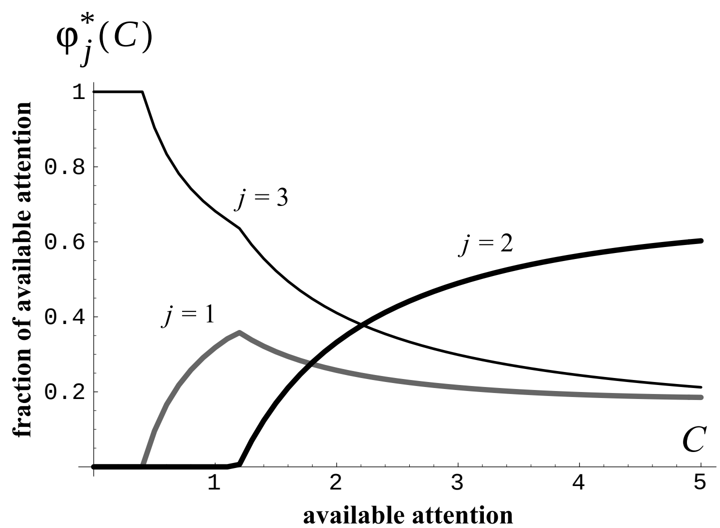


Figure 4.9. Fractional optimal allocations (portfolio example).

At low values of C, most of the consultant resources are allocated to funds 1 and 3. As C increases toward 5 and beyond, it is optimal to allocate roughly 60 percent of the consultants' attention to fund 2, and 20 percent to each of funds 1 and 3.

4.5.3 Activation prices

The **activation price** $\lambda_j(0)$, a measure of the *priority* of process j, provides further insight into our problem. Activation prices are process-specific parameters that can be computed apart from any optimization procedure. For example, we can compute activation prices for the portfolio manager's problem as follows:

$$\lambda_1(0) = 1.0 \qquad \lambda_2(0) = 0.7 \qquad \lambda_3(0) = 1.5\,.$$

4.5.3.1 Classification of processes as active or latent

In discussing the role of activation prices, we often refer to processes as either being active or latent. Under a given resource constraint C, a process j is **active** if $x_j^*(C) > 0$; otherwise, it is **latent** (i.e., $x_j^*(C) = 0$).[9] Now let us return to the family of portfolio-management problems discussed in Section 4.5.2. Figure 4.10 depicts an alternative representation of the solutions from Figure 4.6; different shaded regions correspond to attention units allocated to different processes. Observe that, when C is close to 0, only process 3 is active. As C increases beyond roughly four-tenths of an attention unit, process 1 becomes active. Finally, process 2 enters the picture when C surpasses approximately 1.2 attention units.

4.5.3.2 Prioritization of processes under scarce resources

With active and latent processes defined, we can now increase our understanding of Figure 4.10. Listed in descending order, the activation prices are $\lambda_3(0) = 1.5$, $\lambda_1(0) = 1.0$, and $\lambda_2(0) = 0.7$—in the same order that the corresponding processes become active as C is increased. In general, processes with higher activation prices require smaller resource amounts to become active.

The activation prices can also help us to determine exactly where on the C axis a process becomes active. The key to computing these critical points is to note that a process j is active only when its activation price $\lambda_j(0)$ exceeds the equilibrium price $\lambda(C)$. For

9. I use the term *latent* instead of *inactive* to acknowledge that a process may become active following a state transition or some other transformation.

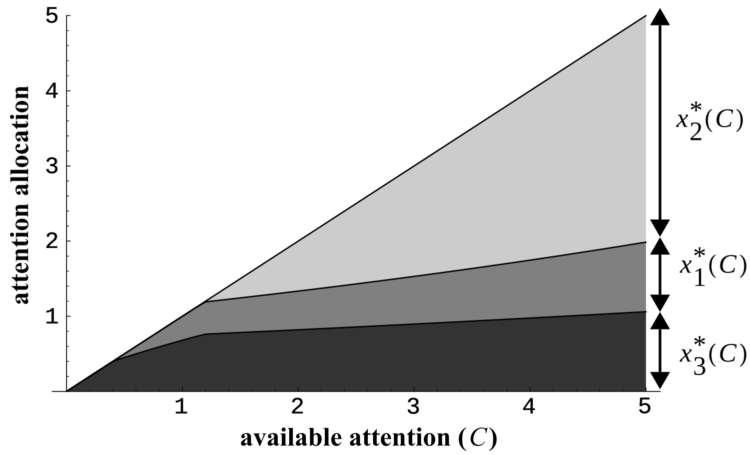


Figure 4.10. Distribution of attention under varying resource constraints. This figure displays optimal attention distributions, as a function of the available attention C, for the portfolio manager's problem. Note that, for low C, only process 3 is active; as C increases, first process 1 and then process 2 becomes active.

example, in the portfolio manager's problem, process 1 becomes active when $\lambda = \lambda_1(0) = 1.0$. Substituting this equilibrium price into Equation 4.48 yields

$$x_1(1.0) + x_2(1.0) + x_3(1.0) = 0 + 0 + 0.405465 = 0.405465 .$$

Thus, process 1 starts becoming active when $C = 0.405465$ attention units. A similar calculation reveals that process 2 becomes active when C surpasses 1.10971 attention units. Figure 4.11 depicts the fractional distribution of attention for various resource amounts C.

Note that the activation prices and the optimal allocations reveal different, related aspects of the solution to the optimization problem. For example, process 3 has the highest activation price, and therefore consumes most of the attention in resource-starved situations. However, it is process 2 that consumes an increasingly larger fraction of resources, as they become more available.

4.5.3.3 Analysis of the addition or removal of a process

In addition to their role in setting priorities on processes, the activation prices help us to predict whether adding or removing a process would disturb the presiding optimal attention distribution. As an illustration, let us return to the portfolio manager's problem with

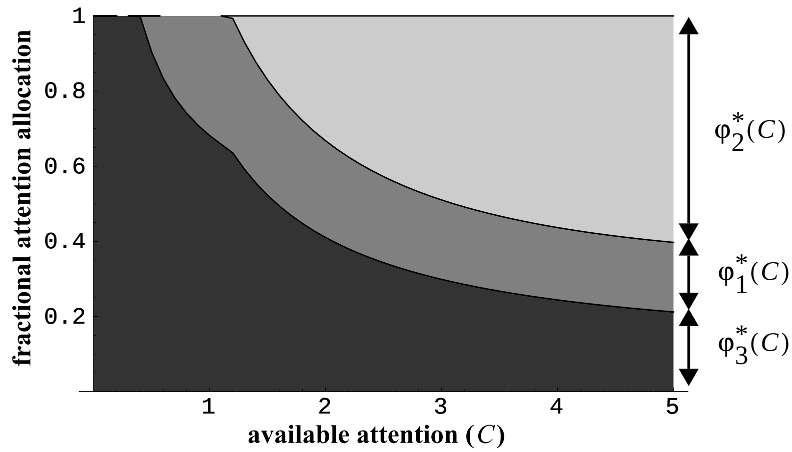


Figure 4.11. Fractional attention allocations as a function of resource. This figure displays varying fractional attention allocations for the portfolio manager's problem; the contents are derived from Figure 4.10 with $\varphi_j^*(C) \equiv x_j^*(C)/C$.

$C = 1$. We found that the optimal allocation was $x_1^*(C) = 0.318239$, $x_2^*(C) = 0$, $x_3^*(C) = 0.681761$, with an associated equilibrium price $\lambda = 0.758588$. We also determined, independent of the optimization procedure, that the activation prices were $\lambda_1(0) = 1.0$, $\lambda_2(0) = 0.7$, and $\lambda_3(0) = 1.5$.

Now consider what happens when we remove a process from this system. If we remove process 2, we should expect no change in the optimal attention allocation, because process 2 has no attention units assigned to it (i.e., $x_2^* = 0$). On the other hand, removing process 1 frees up 0.318239 attention units, which are then redistributed to the other processes. (Similarly, the removal of process 3 results in a redistribution of 0.681761 units of attention to processes 1 and 2.) In general, removing a process j disturbs the prevailing attention allocation only when $x_j^* > 0$.

We can recast this story in terms of activation prices. Recall that, during the optimization, a process receives attention only when its activation price exceeded the equilibrium price λ. Thus, for the removal of a process k to warrant a redistribution of attention, we must have $\lambda_k(0) > \lambda$. Note that, under these circumstances, the modified attention distribution has associated with it a new equilibrium price λ', where $\lambda' < \lambda$.

A similar analysis holds for adding a process. When we add a process k to a set of processes P with equilibrium price λ, we disturb the optimal attention allocation only if k's activation price $\lambda_k(0)$ exceeds λ. Such a disturbance, when it occurs, obliges each process in P to redistribute various amounts of attention to process k. Concomitantly, the equilibrium price increases from λ to λ', and inactivates any process $m \in P$ for which $\lambda' > \lambda_m(0) > \lambda$ (see Section 4.5.3.2).

4.6 Incorporation of partially attended processes

Now that we are equipped to analyze a set of processes given their utility functions $u_j(x_j)$, let us confront the challenge of producing the functions themselves. On first glance, there seems to be no challenge: The procedure of Section 4.3.2 specifies a series of steps for computing $u_j(x_j)$. Unfortunately, it computes u_j only for specific values of x_j; it does not, in general, produce $u_j(x_j)$ in closed, functional form. Thus, we cannot compute expressions for the first derivative $\lambda_j(x_j)$ and, more important, its inverse. We need such expressions to utilize the optimization strategies developed in Sections 4.4 and 4.5.

Fortunately, for a large class of processes, we can approximate $u_j(x_j)$ with a simple exponential form that satisfies the required convexity assumption. Let us explore this approximation by way of example to see how it can simplify our optimization strategy.

4.6.1 Example: Cows on a farm

Consider the following description of a particular cow on a farm. Whenever the cow is healthy (state 1), it produces milk at a rate equivalent to \$10 per day. Otherwise, the cow is sick (state 2), and its production rate slows to \$2 per day. Whether the cow is healthy depends on its previous health state and on whether or not the cow is supervised. Under no supervision, a healthy cow becomes sick with probability $1dt$ in a short time interval dt, and a sick cow heals at one-fifth that rate. Supervision enhances both a healthy cow's ability to resist illness (transition rate 0.5), and a sick cow's ability to become healthy (transition rate 1). Transition diagrams for this process are shown in Figure 4.12.

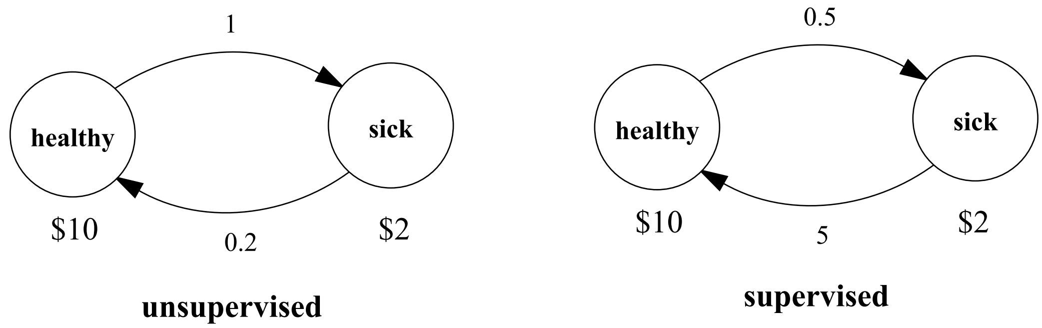


Figure 4.12. A Markov decision process (cow example).
At any given moment, the decision to supervise determines the cow's dynamic properties. For example, when the cow is *sick,* supervising the cow enables a transition rate of 5 to *healthy,* whereas leaving the cow unsupervised permits transitions to *healthy* at a rate of only 0.2. In this particular example, the earning rates are $10 for the healthy state and $2 for the sick state, regardless of whether the cow is supervised.

To maximize any aggregate earning measure, the cow should be kept healthy as often as possible; the optimal policy is to supervise the cow whether it is healthy or sick. We specify that the cow is by default not supervised; thus, to give attention to the cow would be to supervise the cow, and to give no attention would be to leave the cow unsupervised. Furthermore, let us assume that it takes the attention of 4 farm workers to supervise the cow (i.e., $\kappa = 4$), that we are considering an attention horizon of 1 week ($T = 7$), and that the discount rate for earnings is $\alpha = \frac{1}{8}$.

To determine accumulated profit as a function of attention, we employ the procedure summarized in Section 4.3.2. From the problem statement, we immediately extract

$$\mathbf{A}^0 = \begin{bmatrix} -1 & 1 \\ 0.2 & -0.2 \end{bmatrix} \qquad \mathbf{q}^0 = \begin{bmatrix} 10 \\ 2 \end{bmatrix}.$$

We use policy iteration to obtain

$$\mathbf{A}^1 = \begin{bmatrix} -0.5 & 0.5 \\ 5 & -5 \end{bmatrix} \qquad \mathbf{q}^1 = \begin{bmatrix} 10 \\ 2 \end{bmatrix};$$

$$\mathbf{v}^4 = \begin{bmatrix} 82.5990 \\ 80.9746 \end{bmatrix}.$$

We compute transition matrices for intermediate amounts of attention x (where $0 \le x \le 4$) according to Eqs. 4.10 and 4.11:

$$\mathbf{A}^x = \left(1 - \frac{x}{4}\right)\mathbf{A}^0 + \frac{x}{4}\mathbf{A}^4; \tag{4.49}$$

$$\mathbf{q}^x = \left(1 - \frac{x}{4}\right)\mathbf{q}^0 + \frac{x}{4}\mathbf{q}^4. \tag{4.50}$$

Application of Eqs. 4.3 and 4.4 then yields the accumulated discounted earnings:

$$\boldsymbol{v}^x(s) = \frac{1}{s}\left(\left(s + \frac{1}{8}\right)\mathbf{I} - \mathbf{A}^x\right)^{-1}\mathbf{q}^x + \left(\left(s + \frac{1}{8}\right)\mathbf{I} - \mathbf{A}^x\right)^{-1}\mathbf{v}^4; \tag{4.51}$$

$$\mathbf{v}^x(t) = \mathcal{L}^{-1}\{\boldsymbol{v}^x(s)\} \tag{4.52}$$

From $\mathbf{v}^x(t)$, we extract $u(x) = v_1^x(7)$ for a cow that is currently healthy, or $u(x) = v_2^x(7)$ for a presently sick cow. These relationships are plotted in Figure 4.13.

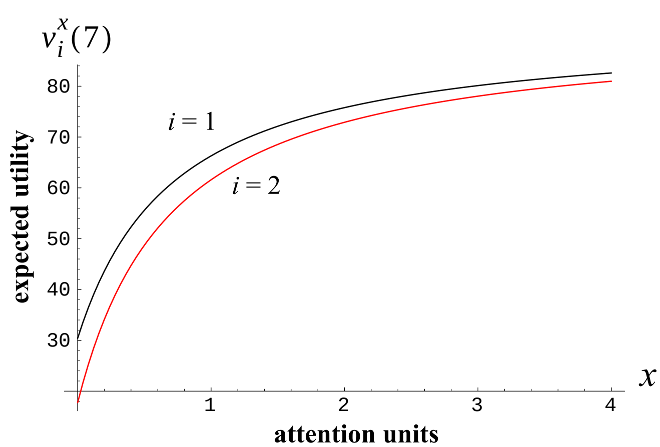


Figure 4.13. Utility-attention relationships (cow example).
Plots for total value as a function of attention units are shown for state 1 and state 2. These plots were generated for values of x between 0 and 4 in increments of 0.04. Separate iterations of Eqs. 4.49 through 4.52 were required for each distinct value of x.

Now suppose that we have several similar cows characterized by different numerical parameters, and a group of farm workers who supervise these cows. To find the optimal attention allocation, we need concave utility functions $u_j(x_j)$ for each cow j. Although the curves in Figure 4.13 do appear to be concave, they were generated from a numerical procedure, not a closed-form expression. Let us see how we can produce closed-form expressions that approximate utility-attention relationships typified by those in Figure 4.13.

4.6.2 Exponential utility-attention modeling

For a large class of processes, we can approximate the utility-attention relationship with the following concave exponential form:

$$\tilde{u}(x) = A - Be^{-\gamma x}, \tag{4.53}$$

where $A,\ B,$ and γ are positive constants, and $\tilde{u}(x)$ denotes an approximation of the actual utility-attention function $u(x)$.

We can solve for these three constants given three values of $u(x)$. One convenient strategy is to use the values $u(0)$, $u(\kappa/2)$, and $u(\kappa)$: the utilities of no attention, one half of the attention capacity, and the full attention capacity, respectively. Substituting these values into Equation 4.53 yields the system

$$\begin{aligned} u(0) &= A - B, \\ u(\kappa/2) &= A - Be^{-\gamma\kappa/2}, \\ u(\kappa) &= A - Be^{-\gamma\kappa}, \end{aligned}$$

or, defining $D = e^{-\gamma\kappa/2}$,

$$\begin{aligned} u(0) &= A - B, \\ u(\kappa/2) &= A - BD, \\ u(\kappa) &= A - BD^2. \end{aligned}$$

Solving this system yields the following cascaded formulae for $A,\ B,$ and γ:

$$D = \frac{u(\kappa) - u(\kappa/2)}{u(\kappa/2) - u(0)},$$

$$B = \frac{u(\kappa/2) - u(0)}{1 - D},$$

$$A = u(0) + B,$$

$$\gamma = -\frac{2}{\kappa}\ln D.$$

Let us apply this approximation to the cow example. For a healthy cow (state 1), we have $u(0) = 30.406$, $u(\kappa/2) = 75.783$, and $u(\kappa) = 82.599$, where $\kappa = 4$. Applying the preceding equations yields the approximation $\tilde{u}(x) = 83.8038 - 53.3976e^{-0.947868x}$.

Figure 4.14 illustrates the exact and approximate utility-attention functions, and their difference. Note how the approximate utility $\tilde{u}(x)$ underestimates the exact utility $u(x)$ for $x < 2$ and overestimates it for $x > 2$. We can reduce the magnitude of these errors by using more sophisticated curve-fitting approaches. For example, we could use additional or different values of the exact utility to select more strategic values of A, B, and γ. Alternatively, we could employ other concave forms of $\tilde{u}(x)$ altogether.

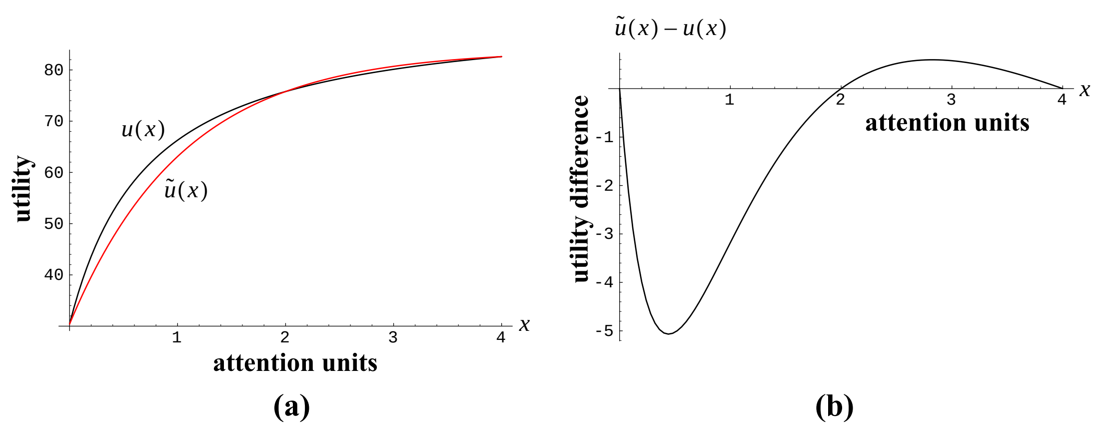


Figure 4.14. Comparison of approximate and exact utilities (cow example). (a) The approximate and exact utilities, $\tilde{u}$ and u respectively, both increase as the attention amount x increases, with diminishing returns. (b) The approximate utility underestimates the exact utility for $x < 2$ and overestimates it for $x > 2$.

4.6.3 Separable optimization

Given closed-form representations of the utility of a process for various amounts of attention, we can now utilize the optimization strategies developed in Sections 4.4 and 4.5. If each process j has an associated utility function $u_j(x_j) = A_j - B_j e^{-\gamma_j x_j}$, then the convex separable problem we want to solve assumes the specific form

$$\text{maximize} \sum_{j=1}^{N} [A_j - B_j e^{-\gamma_j x_j}]$$

$$\text{subject to} \sum_{j=1}^{N} x_j = C$$

$$\text{and} \quad 0 \le x_j \le \kappa_j \qquad \text{for } j = 1, \ldots, N,$$

where A_j, B_j, γ_j, and κ_j are process pararameters, x_1 through x_N are allocation variables, and C is the total amount of attention resource. Thus, to solve this optimization problem, we instantiate the procedure of Section 4.5.1.2, which produces the following steps:

- For each process j, construct the function

$$x_j(\lambda) = \begin{cases} \kappa_j & \lambda < B_j \gamma_j e^{-\gamma_j \kappa_j} \\ \dfrac{1}{\gamma_j} \ln\left[-\dfrac{B_j \gamma_j}{\lambda}\right] & B_j \gamma_j e^{-\gamma_j \kappa_j} \le \lambda \le B_j \gamma_j \\ 0 & \lambda > B_j \gamma_j \end{cases}.$$

- Find the equilibrium price λ^* that satisfies the equation $\sum_{j=1}^{N} x_j(\lambda) = C$.

- Compute the optimal allocations $x_j^* = x_j(\lambda^*)$ for each process j.

Let us apply these steps to the following example involving two cows. Suppose that we would like to allocate C attention units to two cows of the sort described in Section 4.6.1, with the first cow ($j = 1$) being healthy and the second cow ($j = 2$) being sick. If we apply the approximation procedure of Section 4.6.2 with an attention horizon $T = 7$, we obtain the following utility functions for the two cows:

$$u_1(x_1) = 83.8038 - 53.3976e^{-0.947868x_1},$$
$$u_2(x_2) = 82.3595 - 64.6127e^{-0.960705x_2}.$$

We can then use the steps on page 90 to determine the optimal attention allocation. Figure 4.15 displays the optimal attention allocation for values of C between 0 and 8. Compared to the healthy cow, the sick cow merits an additional amount of attention that remains constant for most values of C (Figure 4.15a); with increasing C, the proportion of attention allocated to the sick cow approaches that of the healthy cow (Figure 4.15b).

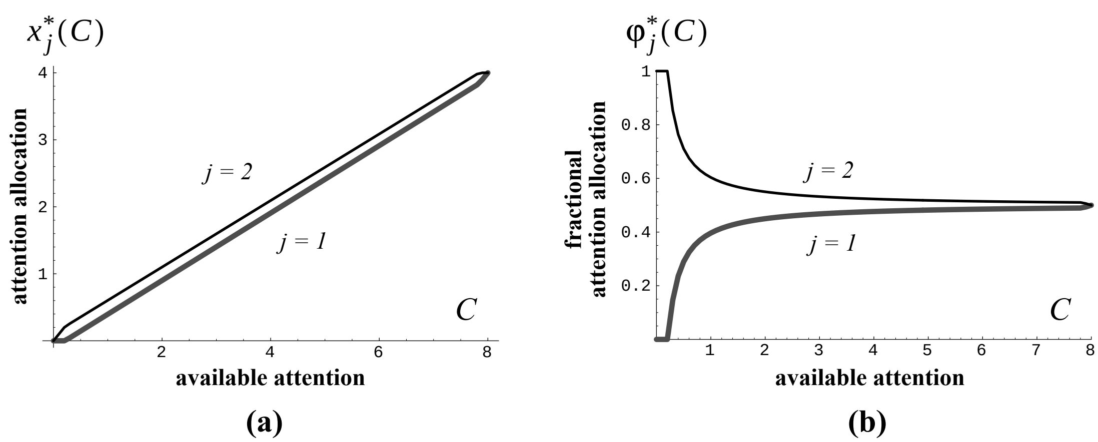


Figure 4.15. Optimal attention allocations (two-cow example).
(a) Optimal attention allocations $x_1^*(C)$ and $x_2^*(C)$ are plotted for various resource constraints C. (b) The same information is plotted with $\varphi_j^*(C) \equiv x_j^*(C)/C$.

4.6.4 Variation of the attention horizon

In addition to varying the total attention resource C, we can also vary the attention horizon T and observe how the solution behaves. Figure 4.16 displays the result of one such experiment ($C = 4$). Note that, when T is approximately less than 0.3, all the available attention is concentrated on the sick cow. For large values of T, the solution asymptotically approaches 1.951 attention units assigned to the healthy cow and 2.049 attention units assigned to the sick cow.

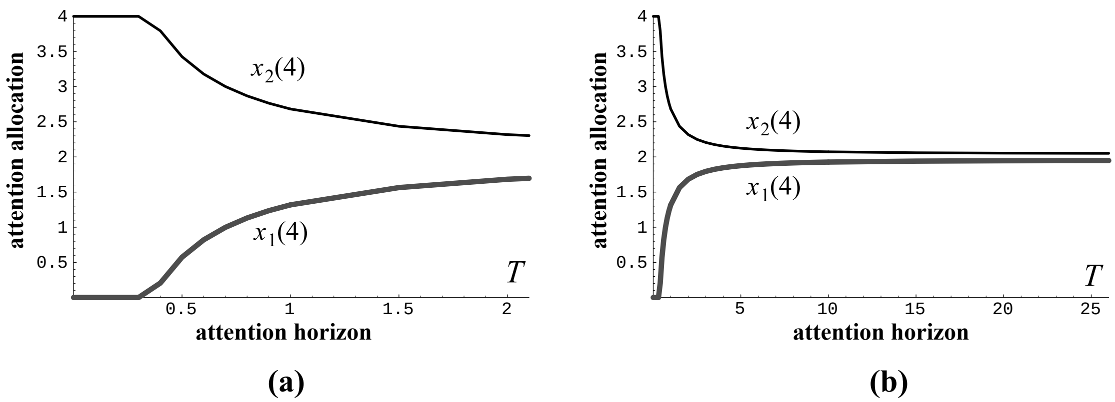


Figure 4.16. Effect of attention horizon on attention allocations (two-cow example). Illustrated are optimal allocations of 4 available attention units for different values of the attention horizon T. (a) For $T < 0.3$, all of the available attention is allocated to process 2. (b) As T increases, the attention distribution converges asymptotically toward 1.951 attention units for process 1, and 2.049 attention units for process 2.

Suppose that we are given a set of N processes, numbered such that $\lambda_1(0) > \ldots > \lambda_N(0)$. Then, in general, for $T \approx 0$, we can expect the available attention to concentrate on process 1, with any remaining attention concentrated on process 2, and so on.

4.7 Discussion

Decision-theoretic methods for managing concurrent processes under resource constraints have been investigated in the recent past. Let us see how my method for distributing continuous amounts of attention advances the state of the art.

4.7.1 Expected cost of delayed action

When we are managing concurrent processes with limited attention resources, we must temporarily forgo the optimal treatment of certain processes in favor of those that need attention more urgently. To address tradeoffs inherent in prioritizing one process over another, Horvitz developed a decision-theoretic measure of the **expected cost of delayed action (ECDA)** of a process. He defined ECDA as the difference in expected value of taking immediate ideal action and delaying ideal action until a time T from now. For a general process, to delay action means to control the process in a prespecified default manner; to take ideal action means to impose an optimal state of control from that time forward.

Although it can be unwieldy to compute ECDA for a general process, Horvitz characterized a class of processes, **pathologic processes,** for which we can assess directly the cost of various delays *T,* given specific current states. For such processes, computing ECDA reduces to computing a probability distribution over the current state, and then using this probability distribution to compute the expected cost of delay. Horvitz demonstrated this formulation of ECDA in various applications, including a system for setting priorities for the display of propulsion-system information at the NASA Mission Control Center, and a prototype system for triaging and transporting victims of a catastrophic trauma scene [Horvitz & Barry, 1995; Horvitz & Seiver, 1997].

We can think of ECDA as a special case of my framework. For a continuous-time Markov decision process with specified default alternatives, the ECDA for a delay T is simply $v_i^{\kappa}(0) - v_i^{\kappa}(T)$, where i is the current state, κ is the attention capacity, and each $v_i^{\kappa}(t)$ is computed according to Eqs. 4.13 and 4.14. Stated otherwise, the ECDA is the expected difference in value from applying full attention immediately, versus applying no attention for a time period T; in both cases, full attention is conferred after T.

4.7.2 From binary to continuous states of attention

My framework for distributing attention among concurrent processes generalizes the concept of ECDA by allowing the allocation of continuously divided amounts of attention. For a process j, we can consider allocating $0 \le x_j \le \kappa_j$ attention units over a horizon of interest T. The ECDA concept, on the other hand, considers only the possibilities of applying no attention ($x_j = 0$) or full attention ($x_j = \kappa_j$) during T.

We can appreciate this difference by considering the application of each methodology to the following multiple-process monitoring scenario. Consider N processes with attention capacities $\kappa_1 = \ldots \kappa_N = 1$, and an attention amount $C = 1$ to be allocated over a horizon T.

In the ECDA framework, the problem is to decide *which process* should receive this unit of attention. Assuming additive costs, the ECDA framework selects the process j that minimizes the sum total ECDA of all processes $k \neq j$, that is, the process with the highest ECDA for a delay T (Figure 4.17a).

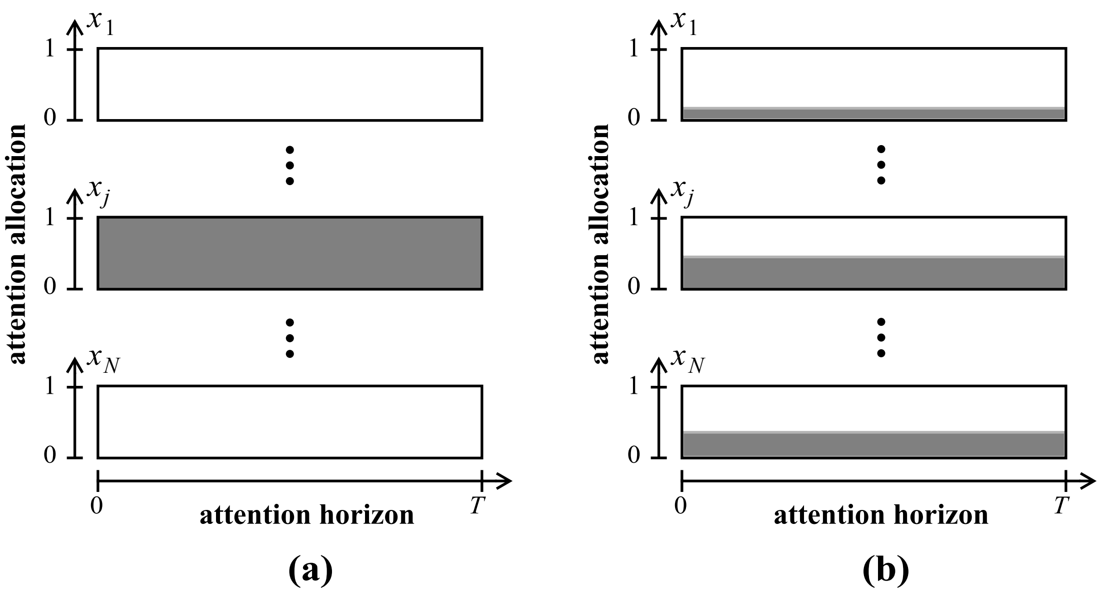


Figure 4.17. Strategies for allocating attention among multiple processes. Illustrated are N processes with attention capacities $\kappa_1 = \ldots = \kappa_N = 1$ over an attention horizon T. One attention unit is to be allocated among these processes. (a) The ECDA methodology decides which process should receive this attention. (b) My attention-distribution methodology, in contrast, considers how this attention can be divided and distributed among the processes.

My framework casts the attention-allocation problem as one of deciding *how much attention* to allocate to each process. In contrast with ECDA, my framework can allocate portions of the available attention to multiple processes; such a plan entails multitasking among processes that can operate while receiving less than their full attention capacities (Figure 4.17b).[10]

10. The ability to operate processes at partial levels of attention also allows us to model scenarios where the attention capacities κ_j differ from one another.

There are certain catastrophic situations (e.g., $T \to 0$ or $C \to 0$) for which the best plan may be to concentrate all the available attention on a few urgent processes. However, the ability to multitask usually produces greater overall utility than we can attain by assigning all the available attention to one process.

4.7.3 From one-dimensional to two-dimensional urgency measures

Although ECDA can serve as a priority measure for pathologic processes, my research demonstrates that this one-dimensional notion of urgency does not scale up to the general, attention-allocation problem. Given an attention amount C and an attention horizon T, my alarm framework produces, for each process j, a priority measure $\lambda_j(0)$ *and* an attention allocation $x_j^*(C)$. Each type of measure captures a different aspect of the allocation of attention over the finite duration T; both the portfolio-management example from Sections 4.4 and 4.5 and the example of the large intensive-care unit from Section 6.3 illustrate how processes that have higher priority do not necessarily need more attention. Although the ECDA framework also embodies a notion of a temporal horizon T, it considers only the effects of actions that are applied at the present instant. This assumption enables the one-dimensional notion of urgency that is captured by the ECDA metric.

4.8 Summary

In this chapter, we confronted the challenge of apportioning limited short-term attention resources among a collection of concurrent processes. We saw how this problem can be formulated as a convex separable problem, whose components represent processes that can operate at partial levels of short-term attention, in the sense described in Chapter 3. As in Chapter 3, we recognized two aspects of the attention assigned to a process j: (1) the attention amount x_j in the context of the process's attention capacity κ_j, and (2) the temporal horizon T over which this attention is applied.

In analyzing the convex separable problem, we identified metrics that provided insight into different aspects of the optimal management of these processes. Given an attention amount C, the optimal attention allocations $x_j^*(C)$ indicate how this attention should be shared among the processes. We also investigated the activation prices $\lambda_j(0)$ for each process j, and showed how they provide a basis for prioritizing processes when resources are scarce. These alarm signals can help to focus the attention of a group of agents who are managing several processes to which they cannot attend all at once.

We used relatively simple examples to illustrate various concepts presented in this chapter. In Chapter 5, we see how these concepts can scale up to a more complex problem of real-world interest.

Chapter 5

Simulated Patients and Patient Wards

Thus far, I have motivated and presented a methodology for generating quantitative, decision-based alarm signals for a set of concurrent, continuing processes. In this chapter, I describe the SIMON[1] alert system, a program that applies this methodology to the important problem of monitoring multiple patients in a busy intensive-care unit (ICU).

Using the methods of Chapter 4, SIMON produces alarm signals for a collection of simulated ICU patients who are being ventilated mechanically. Each patient, in turn, is modeled as a process (of the type developed in Chapter 3) whose parameters are derived in part from VentPlan [Rutledge et al., 1993].[2] I used the alarm signals produced by SIMON to support the validation phase of this research, described in Chapter 6.

I present SIMON's overall architecture in Section 5.1. Then in Sections 5.2 and 5.3, I develop a schema, or template, for the ICU-patient models used in my research. I illustrate the use of this schema by way of example in Section 5.4.

---

1. SIMON stands for System for Intelligent MONitoring. I developed SIMON using Mathematica 3.0 on a 180 MHz, PowerPC 604e machine running MacOS 8.1.

2. VentPlan is a validated prototype ventilator-management advisor that computes the utilities of alternative ventilator settings in a patient-specific manner. I use the utilities of VentPlan to construct models that represent realistic ICU patients.

5.1 Architecture

Let us begin by reviewing the architecture of the SIMON alert system. To produce alert signals for a collection of patients, SIMON requires up-to-date information on each patient, as well as the values of user-controlled parameters that specify the planning horizon and the total available attention resources.

5.1.1 Single patient unit

Consider a single, critically ill patient. The patient receives ongoing life support, which gives his body a chance to heal itself. He is monitored through devices that provide information on his current physiologic status. Observed changes in telemetry indicate possible underlying physiologic changes that warrant a change in treatment protocol. However, any change in treatment requires the attention of a capable clinician who physically attends to the patient at the bedside.

In determining how to allocate attention, SIMON interacts not directly with the physical patient, but rather with a computational representation, or a model, of that patient. The model represents key aspects of the patient's static and dynamic characteristics. SIMON uses different models to simulate different types of patients, and interprets observed data from each patient in light of specific characteristics that are captured in the corresponding model.

Figure 5.1 illustrates a patient-monitoring unit, or **patient unit,** consisting of a physical patient and his corresponding patient model. The model processes specific queries concerning the patient, taking into account measurements obtained from the patient, actions applied to the patient, and fundamental patient characteristics, as encoded in the model. Thus, a "patient," from SIMON's point of view, is a patient unit in the sense just described. I often use the term *patient* to refer to the physical patient in conjunction with his corresponding model.

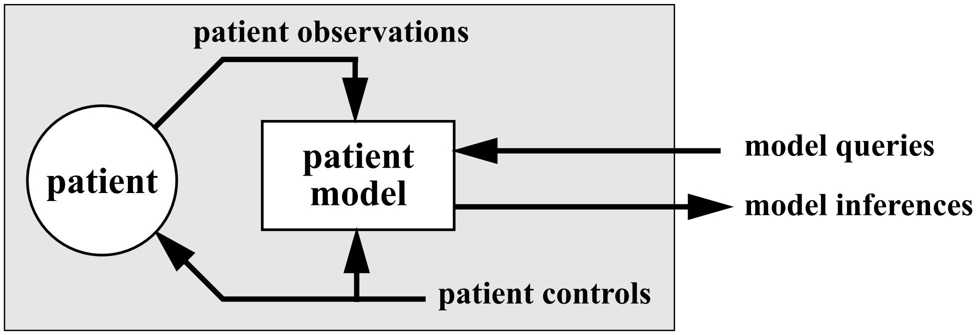


Figure 5.1. A patient unit.
A patient unit consists of a physical patient and a patient model. The physical patient is observed and treated over time, as indicated by the information paths *patient observations* and *patient controls,* respectively. The patient model is a computational representation of the patient. The model produces inferences in response to incoming queries by considering prior knowledge about the patient and ongoing information obtained from monitors attached to the patient.

5.1.2 Patient-unit interface

Figure 5.2 illustrates the two types of interactions that occur between SIMON and a typical patient unit j. The quantity x_j denotes the amount of attention applied to patient j, and the quantity λ_j denotes the marginal benefit per unit of additional attention. The attention is applied for a positive duration T starting from the present time.[3] Given a specified value of T, there is a correspondence between the quantities x_j and λ_j: The model produces λ_j given x_j and T, or x_j given λ_j and T. SIMON relies on both types of interactions in computing its alarm signals.

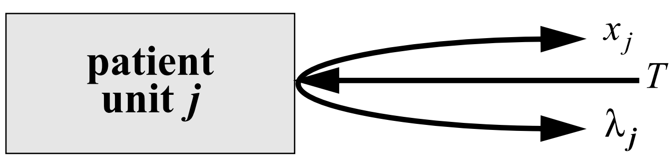


Figure 5.2. Patient-unit interface.
A patient unit can process two types of queries. Given an attention horizon T, it can infer the marginal price λ_j of an attention amount x_j, or the attention amount x_j that corresponds to a given marginal price λ_j.

3. These quantities are discussed in detail in Chapter 4.

5.1.3 Multiple patients

A typical ICU ward contains several patients who are being managed by a limited clinician-resource pool. SIMON produces alarm signals to assist these clinicians in utilizing the limited attention resources intelligently. Figure 5.3 illustrates the interactions involved in producing such signals. SIMON receives a query that comprises an available attention amount and an attention horizon of interest. SIMON, in turn, computes alarm signals by coordinating a series of computations with each patient unit.

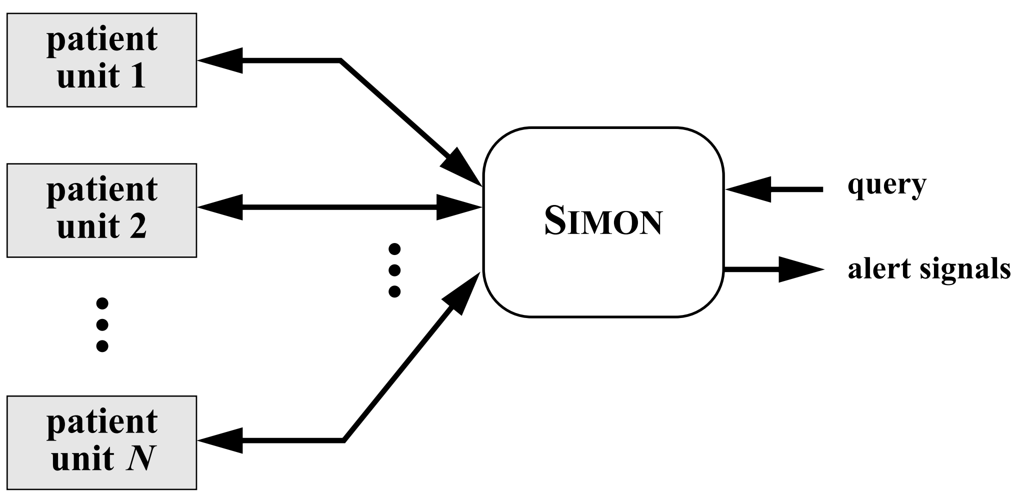


Figure 5.3. SIMON alert system.
SIMON computes alarm signals by integrating relevant information from each patient unit. The alarm signals are generated in response to specific queries. The information exchange that occurs along each bidirectional arrow conforms to the protocol shown in Figure 5.2.

In an ICU setting, SIMON would be used to generate alarm signals on a continuing basis. To accomplish this behavior, SIMON must receive a continuous stream of queries and up-to-date patient information. The alarm signals change in response to changes in the query, the patient composition, or the state of one or more patients.

5.1.4 SIMON interface

Figure 5.4 depicts a generic user interface for SIMON. For each patient j, the activation price $\lambda_j(0)$ and the attention allocation $x_j^*(C)$ are displayed, along with an appropriate patient identifier, in descending order of $\lambda_j(0)$. Given $\lambda_1(0) > \lambda_2(0) > \ldots > \lambda_N(0)$, SIMON recommends that an attention amount C, over an attention horizon T, should be utilized by first allocating $x_1^*(C)$ attention units to patient 1, then allocating $x_2^*(C)$ attention units to patient 2, and so on.

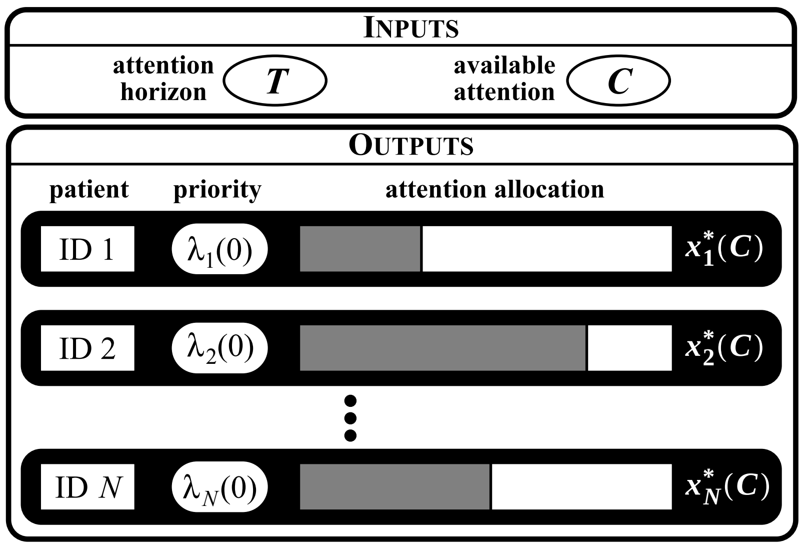


Figure 5.4. SIMON interface.

SIMON computes alert signals based on information received from the patient units, and the user-specified parameters T (the attention horizon) and C (the available attention amount). The alert signals $\lambda_j(0)$ and $x_j^*(C)$ for each patient j are displayed alongside a corresponding patient identifier. The patients are prioritized and listed in descending order of $\lambda_j(0)$ (i.e., $\lambda_1(0) > \lambda_2(0) > \ldots > \lambda_N(0)$).

In the specific example shown in the figure, $x_2^*(C) > x_N^*(C) > x_1^*(C)$, as indicated by the relative lengths of the horizontal bars. Note that the highest-priority patient (ID 1) is not necessarily the patient who warrants the most attention (ID 2). The two alarm concepts are related, but convey separate notions. In general, we should recognize the difference between a process's priority and the amount of attention that is assigned to that process.

5.2 Patient-model structure

Having reviewed the architecture of SIMON, let us turn our attention to the patient models themselves. I build these models using continuous-time Markov-process components and the concepts developed in Chapter 3. The models share a common process structure, but differ from one another in the way that this structure is parameterized. I develop the structure in this section; a strategy for quantifying it is described in Section 5.3.

5.2.1 Physiologic states

While he is in the ICU, a patient can assume one of two physiologic states: an **unstable state** and a **critical state.** From the unstable state, the patient can deteriorate to the critical state, whereas a critical-state patient can improve to the unstable state. In addition, there are two ways for the patient to leave the ICU. An unstable-state patient can improve to a **stable state** and be discharged to the floor, whereas a patient in the critical state may devolve to the **dead state.** These state transitions are shown in Figure 5.5a.

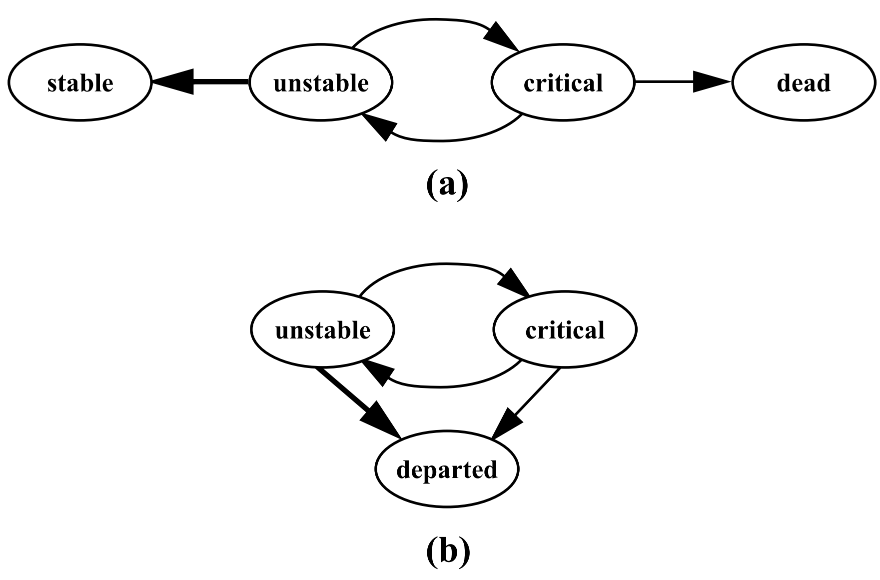


Figure 5.5. Physiologic states.
These diagrams depict continuous-time Markov-process representations of an ICU patient. In (a), the patient transitions between the *unstable* and *critical* states, eventually terminating in either the *stable* state or the *dead* state. In (b), the *stable* and *dead* states are aggregated into a *departed* state. The thick arrows indicate transitions that yield a reward of 1. The remaining, thin arrows denote transitions that yield no reward.

The desired outcome is for the patient to make the transition to the stable state. I endow this transition with a reward of 1; it is distinguished in the figure by a thick arrow. Thin arrows denote the other transitions, which yield no reward. A patient who enters the dead state loses all possibility of making a rewarding transition.

Figure 5.5b depicts an alternative representation of the scenario modeled by Figure 5.5a. The stable and dead states are aggregated into a single **departed state.** A transition from unstable to departed means that the patient has left the ICU alive; one from critical to departed indicates that he has died. A reward is associated with only the former transition, which is distinguished from the other transitions by a thick arrow.

For the purpose of computing expected rewards, the two models shown in Figure 5.5 are equivalent. However, the preceding structural transformation is necessary for computational purposes, because the process machinery of Chapter 3 applies to only ergodic processes. To maintain clinical relevance, I continue my exposition by augmenting the model of Figure 5.5a, recognizing that I can always convert this working model to an equivalent, ergodic form suitable for computation.

5.2.2 Alternative treatments

I now incorporate treatments to our working model. At any given moment, our patient is receiving one of two types of treatments: a conservative **maintenance treatment,** or an **aggressive treatment.** These treatments influence patient dynamics differently; different transition rates prevail depending on the selected treatment. The maintenance regimen is preferred whenever the patient is unstable; a critical patient necessitates the more aggressive therapy. A state diagram for this Markov decision process (MDP) is shown in Figure 5.6.

5.2.3 Transitory and persistent attention effects

Recall that a process that is receiving attention operates according to its optimal policy, whereas the same process that is receiving no attention operates according to its default

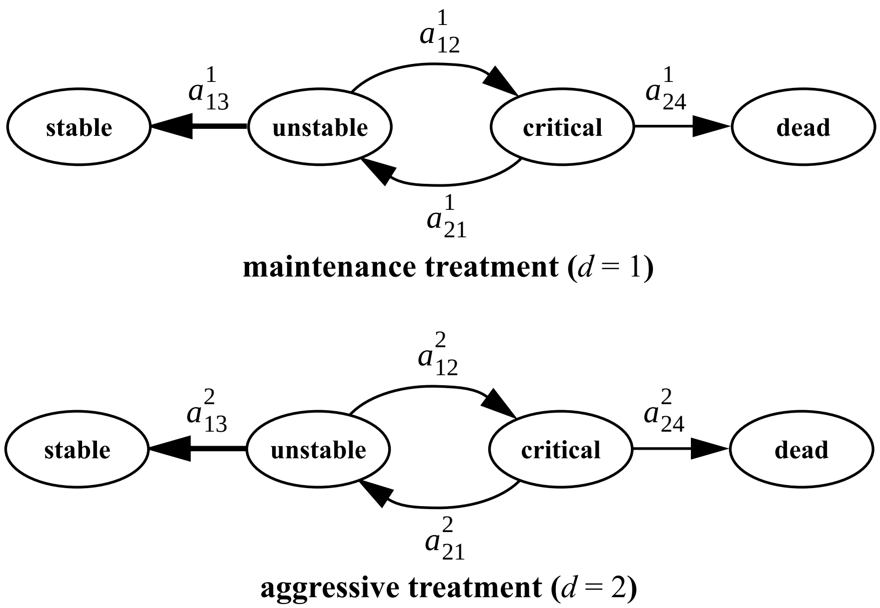


Figure 5.6. Physiologic states in the context of different treatments. Illustrated is a continuous-time MDP, representing an ICU patient who can receive one of two treatments at any given time. The transition rates between states depend on which treatment the patient is receiving. The transition from *unstable* to *stable* garners a reward of 1, as indicated by the thick arrow. (The states are indexed as follows: state 1 is *unstable*, state 2 is *critical*, state 3 is *stable*, and state 4 is *dead.*)

policy (see Section 3.3). We have seen that the optimal policy is to treat an unstable patient conservatively and a critical patient aggressively. To model the favorable effects of attention, though, we need a default policy, which stipulates how the patient is treated when he is unattended.

There are several such default policies, depending on which treatment options require that the patient be afforded attention. For example, we could specify that the aggressive-treatment option is available only under attention, or, equivalently, that the patient undergoes maintenance treatment when he is receiving no attention. In this scenario, the effects of attention are **transitory,** in that nondefault treatments are terminated when the attention is withdrawn. Such attention effects are analogous to a push-and-hold button, which can remain pushed only when it is receiving human attention (Figure 5.7a).

Not all treatments in medicine are transitory. For example, when we adjust an ICU patient's ventilator settings, the effects of our attention are **persistent:** Treatments

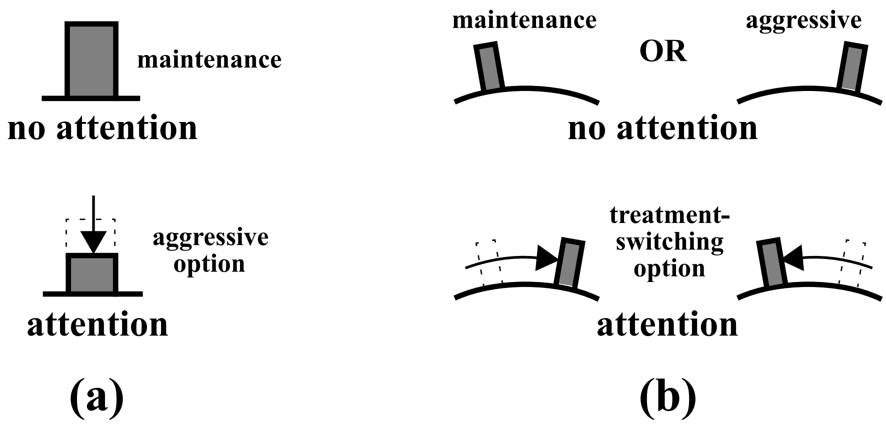


Figure 5.7. Transitory and persistent attention effects.
The distinction between transitory and persistent attention effects is exemplified by the difference in the response of a push-and-hold button (a) and a toggle switch (b). In (a), the effects of attention are transitory: Attention enables the option of providing aggressive treatment, whereas the withdrawal of attention necessitates the termination of any such treatment. In (b), the effects of attention are persistent: Attention enables the option of switching treatments, whereas the withdrawal of attention results in continued administration of only the most recent treatment.

introduced under attention remain in place after the attention is withdrawn. The persistence of treatment settings is exemplified by a control knob, which can be adjusted only when it is receiving human attention. For our patient models, I consider two possible ventilator settings: a maintenance setting and an aggressive setting. Such a ventilator-setting mechanism is analogous to a toggle switch, shown in Figure 5.7b.

5.2.4 Modeling persistence

The persistent effects of adjusting a ventilator require additional modeling, because the model must remember the treatment that is being administered whenever attention is withdrawn. We can encode such memory effects by introducing an additional state variable to our working model.

To apply this modeling strategy to our patient, I consider the possible physiologic conditions (unstable and critical states) in conjunction with the possible treatments (maintenance and aggressive). There are a total of four physiology–treatment combinations, each combination a possible state in our revised patient model. In addition, I retain the two absorbing states (stable and dead), which represent the two ways that the patient can leave the ICU. Thus, there are a total of six patient states to consider.

We must also revise our concept of alternatives. Although I continue to provide either maintenance or aggressive treatment to our patient, I now distinguish our alternatives by whether the ventilator setting can be adjusted. In the first alternative, **continue prevailing treatment,** the ventilator settings cannot be changed; the second alternative, **ensure optimal treatment,** adjusts the ventilator settings, as necessary, to provide maintenance treatment to an unstable patient and aggressive treatment to a critical patient. As in the previous model, an unstable patient can change to stable or critical, whereas a critical patient can make a transition to unstable or to dead. Figure 5.8 depicts our revised patient model.

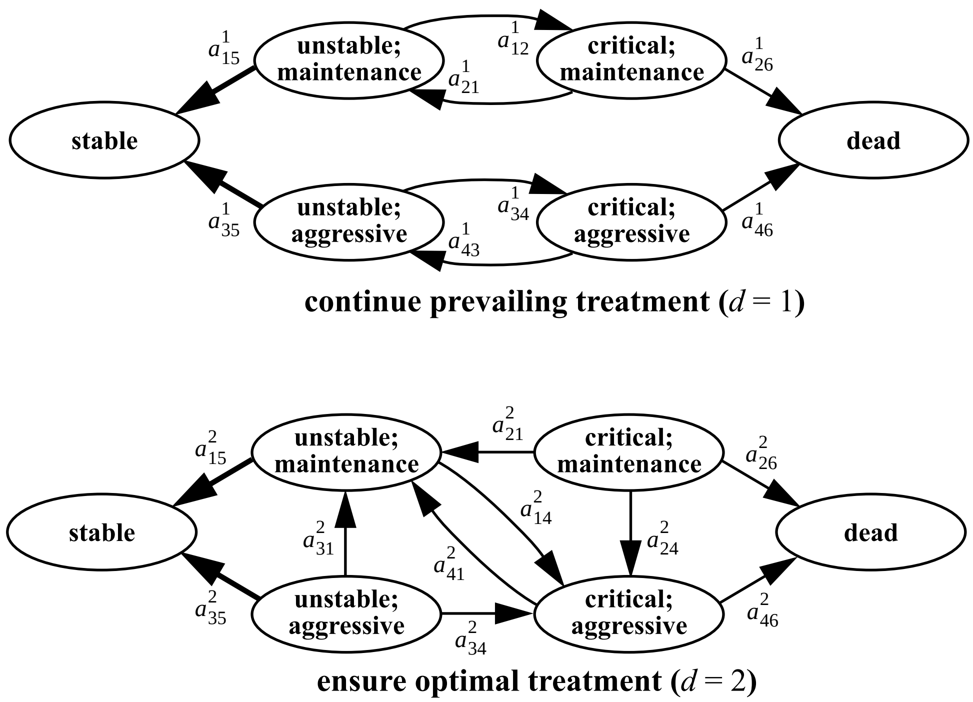


Figure 5.8. Physiologic states with persistent treatment effects.
This continuous-time MDP represents an ICU patient whose ventilator settings can be left unchanged (alternative 1) or adjusted to the optimal setting (alternative 2). Transitions from an unstable to a stable state yield a reward of 1, as indicated by the thick arrows. Note that the states are indexed: (1) *unstable-maintenance;* (2) *critical-maintenance;* (3) *unstable-aggressive;* (4) *critical-aggressive;* (5) *stable;* and (6) *dead.*

5.3 Patient parameters

The preceding model structure must be quantified for it to be a complete patient model. To construct models that represent realistic ICU patients, I employ a parameterization strategy that incorporates outputs from VentPlan, a validated prototype ventilator-management advisor [Rutledge et al., 1993]. Sections 5.3.1 through 5.3.5 describe the specification of transition rates; the remaining parameters are discussed in Section 5.3.6.

5.3.1 VentPlan physiologies and treatments

VentPlan computes recommended ventilator settings of ICU patients by interpreting the patients' physiologic measurements. The physiologic and treatment variables used by VentPlan are shown in Table 5.1. For a given set of physiologic values, VentPlan computes the predicted steady-state effects of alternative ventilator settings. The predicted effects, in turn, are evaluated according to a multiattribute utility function [Farr, 1991]. The recommended treatment maximizes the expected value of this utility.

PHYSIOLOGIC STATE VARIABLES		TREATMENT VARIABLES	
SYMBOL	PHYSIOLOGIC QUANTITY	SYMBOL	PHYSIOLOGIC QUANTITY
Q	cardiac output	RR	respiratory rate
V_{ds}	dead space	V_{Tset}	set tidal volume
$\dot{V}_{\mathrm{O}_2}$	oxygen consumption rate	FIO_2	fraction of inspired oxygen
f_{S}	shunt fraction	$PEEP$	positive end-expiratory pressure
RQ	respiratory quotient		

Table 5.1. VentPlan variables.
Shown here are the principal quantities that are processed by VentPlan's mathematical model. The physiologic variables encode the patient's health status, and the treatment variables specify the ventilator settings. (Not shown are other parameters that VentPlan uses to infer the patient's physiology, and to predict the effects of alternative treatments.)

The utility computed by VentPlan is inversely related to the probability that the patient will transition to a less desirable state during a time interval of interest. I exploit this interpretation of VentPlan's utilities in assessing transition rates for our patient model.

5.3.2 Patient states

Our patient models assume that the physiologic state can be observed directly through appropriate telemetry. I associate a set of physiologic values Θ_u with the unstable state, and a "worse" set of physiologic values Θ_c with the critical state. Using VentPlan, I compute optimal ventilator settings X_u and X_c for each of these respective physiologies; I designate X_u as the maintenance treatment and X_c as the aggressive treatment. Each physiology–treatment combination defines a patient state in our model. The patient states and their corresponding VentPlan descriptions are shown in Table 5.2.

State	Description (physiology; treatment)	VentPlan values
1	*unstable; maintenance*	Θ_u, X_u
2	*critical; maintenance*	Θ_c, X_u
3	*unstable; aggressive*	Θ_u, X_c
4	*critical; aggressive*	Θ_c, X_c

Table 5.2. Patient states.
The states of the patient model from Figure 5.8 are defined in terms of VentPlan input and output variables. The symbols Θ_u and Θ_c denote the *unstable* and *critical* physiologic states, respectively; X_u and X_c denote the recommended respective *maintenance* and *aggressive* ventilator treatments. Not shown are the absorbing states *stable* and *dead*.

The resulting assignment of patient states should be consistent with the patient's clinical scenario. An unstable state should always be more desirable than a critical state, and, given a physiologic state, the optimal treatment should always be more desirable than non-optimal treatments. Let $u(\Theta, X)$ denote the utility (computed by VentPlan) of applying treatment X to a patient with physiologic state Θ. Our clinical scenario, then, requires that $u(\Theta_u, X_u) > u(\Theta_u, X_c) > u(\Theta_c, X_c) > u(\Theta_c, X_u)$. The first and third inequalities are satisfied automatically by VentPlan's utility-maximization routine; the second inequality holds for valid physiologic-state specifications Θ_u and Θ_c.

5.3.3 Deterioration and death rates

I now convert the utilities of our patient states into appropriate transition rates. I begin by assessing the probability of an adverse change in physiology over a 24-hour period, p_{24}, as a function of VentPlan's utility, u, for a patient suffering from congestive heart failure (CHF).[4] Figure 5.9 illustrates the assessed quantity p_{24} as a function of u.

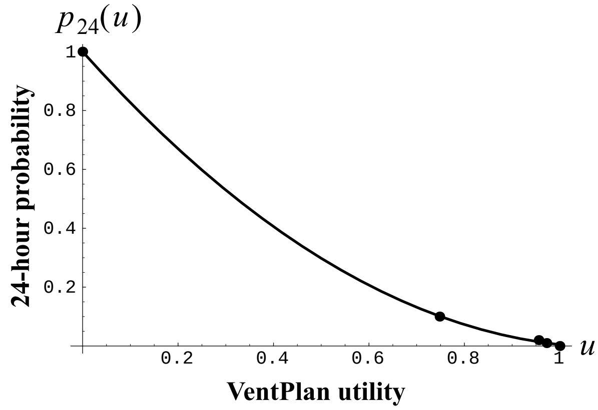


Figure 5.9. Assessment of 24-hour probabilities of adverse physiologic changes. The quantity p_{24}, the probability of an adverse change in physiology over a 24-hour period, is plotted as a function of u, the utility computed by VentPlan, for simulated patients with varying degrees of CHF. The assessed points are satisfied by the quadratic equation $p_{24}(u) = 0.999928 - 1.81227u + 0.816943u^2$. (Note: This mapping does not cover extremely ill patients for whom the utility is less than or equal to 0.)

Given a value of p_{24}, I can now obtain a continuous-time transition rate k by noting that the probability $y(t)$ of remaining in the current physiologic state after an elapsed time t is governed by the differential equation

$$\frac{d}{dt}y(t) = -kt, \tag{5.1}$$

where t is measured in hours, and by the conditions $y(0) = 1$ and $y(24) = 1 - p_{24}$. Solving Equation 5.1 in conjunction with these conditions yields the transition rate k as a function of p_{24}:

4. The probabilities were assessed by Geoffrey Rutledge, M.D., Ph.D., an emergency-medicine specialist with extensive experience in the management of critically ill patients.

$$k(p_{24}) = -\frac{1}{24}\ln(1 - p_{24}). \qquad (5.2)$$

For each patient state, we can use this derivation procedure to calculate the instantaneous transition rate to the next less desirable state. Recall that adverse physiologic changes proceed from *unstable* to *critical* to *dead*. I describe the transition rates from unstable to critical as **deterioration rates,** and the transition rates from critical to dead as **death rates.** Table 5.3 summarizes the assignment of these parameters.

CURRENT STATE	TRANSITION RATES	
	PARAMETERS	VALUES
1	a_{12}^{1}, a_{14}^{2}	$k(p_{24}(u(\Theta_u, X_u)))$
2	a_{26}^{1}, a_{26}^{2}	$k(p_{24}(u(\Theta_c, X_u)))$
3	a_{34}^{1}, a_{34}^{2}	$k(p_{24}(u(\Theta_u, X_c)))$
4	a_{46}^{1}, a_{46}^{2}	$k(p_{24}(u(\Theta_c, X_c)))$

Table 5.3. Deterioration and death rates.
The deterioration and death rates for the model of Figure 5.8 are summarized here. Each pair (Θ, X) denotes a physiology–treatment combination for a patient state. The quantity $u(\Theta, X)$ is the utility from VentPlan, $p_{24}(u)$ is the 24-hour probability of adverse physiologic changes (assessed according to Figure 5.9), and $k(p_{24})$ is the transition rate obtained from Equation 5.2.

5.3.4 Recovery and discharge rates

I now specify the various rates of transitions to more desirable patient states. Unfortunately, VentPlan does not compute quantities that reflect such healing rates, because its utility model was designed for the sole purpose of minimizing patient hazards [Farr 1991; Rutledge et al., 1993]. Therefore, I assess these healing rates separately.

One strategy for specifying clinically plausible healing rates involves the modulation of assessed base healing rates. For a given physiologic state, the **base healing rate** is the rate of transitions to a more desirable physiologic state when no adverse forces are acting on the patient.[5] Since illness does impede the healing process, I compute, for each patient state, an **effective healing rate** by attenuating the base healing rate h according to the adverse-transition rate k confronted by the patient. Figure 5.10 depicts a linear implementation of this strategy. When k assumes its minimum value, the effective healing rate is equal to h; at the other extreme, when $k = k_{max}$, the effective healing rate is zero.

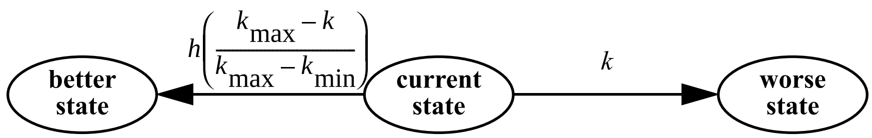


Figure 5.10. Modulation of base healing rate.
This diagram illustrates the computation of an effective healing rate for a generic, current physiologic state. The quantity h denotes the base healing rate, and k denotes the adverse transition rate. We compute the effective healing rate by using k to attenuate h. (Note that k_{max} and k_{min} are the adverse-transition rates for the VentPlan utilities 0 and 1, respectively.)

I use the preceding parameterization strategy to specify healing rates for the model shown in Figure 5.8. I begin by assessing h_u and h_c, the base healing rates for the unstable and critical physiologies, respectively. Then, for each state i, I compute the effective healing rate by using the adverse-transition rate k_i to modulate the appropriate base healing rate. I describe the transition rates from unstable to stable as **discharge rates,** and the transition rates from critical to unstable as **recovery rates.** The assignment of values to these rate parameters are summarized in Table 5.4.

I often refer to a transition rate a_{ij} in terms of its multiplicative inverse, $(a_{ij})^{-1}$. I describe $(a_{ij})^{-1}$ as the **transition time** from state i to state j. Disregarding all transitions to states other than j, a process starting in state i will remain in state i with probability $\frac{1}{e}$ (and therefore will be found in state j with probability $1 - \frac{1}{e}$) after an amount of time $(a_{ij})^{-1}$ elapses.[6]

---

5. Technically, our simulated patients do not heal in *response* to treatments; rather, they heal on their own, at various stochastic rates that are *influenced* by the choice of treatment. The goal of treatment is to help patients get better before they get worse.

CURRENT STATE	TRANSITION RATES	
	PARAMETERS	VALUES
1	a_{15}^{1}, a_{15}^{2}	$h_u\left(\frac{k_{max}-k_1}{k_{max}-k_{min}}\right)$
2	a_{21}^{1}, a_{21}^{2}	$h_u\left(\frac{k_{max}-k_2}{k_{max}-k_{min}}\right)$
3	a_{35}^{1}, a_{35}^{2}	$h_u\left(\frac{k_{max}-k_3}{k_{max}-k_{min}}\right)$
4	a_{43}^{1}, a_{41}^{2}	$h_u\left(\frac{k_{max}-k_4}{k_{max}-k_{min}}\right)$

Table 5.4. Discharge and recovery rates.

Formulae for computing the discharge and recovery rates for the model of Figure 5.8 are shown here. The quantities h_u and h_c are base healing rates assessed for the unstable and critical states, respectively. For each state i, k_i is the adverse-transition rate computed via the formulae displayed in Table 5.3.

5.3.5 Treatment rates

The remaining unspecified transition rates, a_{31}^{2} and a_{24}^{2}, relate to the correction of inappropriate treatment, which can happen only under attention. I assume that these corrections take effect rapidly, at rates of (1 minute)$^{-1}$.

5.3.6 Other model parameters

We now have a process structure complete with transition rates. The remaining parameters specify rewards associated with the process, and the number of attention units required to operate the process optimally. I allow only the latter to vary between patients.

5.3.6.1 Reward parameters

As noted in Section 5.2.1, I associate a reward of 1 with transitions from the unstable to the stable condition; transition rewards are otherwise 0. In other words, I specify

6. Note: $\frac{1}{e} \approx 0.368$; $1 - \frac{1}{e} \approx 0.632$.

$$r_{15}^{1} = r_{15}^{2} = r_{35}^{1} = r_{35}^{2} = 1,$$

and $r_{ij}^{k} = 0$ for the other combinations of i, j, and k. I also employ, for each patient, a discount rate $\alpha = 0.02$ (see Section 3.2.3). This reward scheme appraises the intrinsic value of each patient's life equally, but favors patients who are discharged sooner.

5.3.6.2 Total attention capacity

For each patient j, I specify the attention capacity κ_j, the number of clinicians needed to treat this patient optimally on a continuing basis. This value is typically 1 for a patient supported by mechanical ventilation; however, the optimal treatment of patients with complex illnesses or other concomitant challenges may require the attention of more than one clinician. (The parameter κ_j is discussed in Chapter 4.)

5.4 Example: Congestive heart failure

Let us construct a model for Mr. Dickinson, a 61-year-old man who is recovering from a heart attack. Mr. Dickinson is being monitored in the ICU for signs of CHF. In CHF, the heart loses its normal ability to pump blood to the systemic circulation. There is a decrease in cardiac output and a concomitant increase in the pulmonary wedge pressure. In severe CHF, this pressure causes fluid to fill the lungs; a physiologic shunt results from the impaired oxygenation of blood flowing through the pulmonary capillaries. Severe CHF exacerbates itself when untreated, and is therefore life threatening.

5.4.1 Patient specification

Table 5.5 specifies a patient model for Mr. Dickinson. The unstable and critical physiologic states differ in their values for cardiac output Q and shunt fraction f_S.[7] The discharge and recovery times are 2 days and 1 day, respectively. Mr. Dickinson has no special conditions that require his attention capacity to be greater than 1. The values in

Table 5.5 describe Mr. Dickinson's particular condition; other patients who are suffering from nonidentical illnesses would be described by a different set of numerical values.

<table>
<tr><th rowspan="2">PHYSIOLOGIC STATE</th><th colspan="5">PHYSIOLOGIC PARAMETERS</th></tr>
<tr><th>Q</th><th>V_{ds}</th><th>$\dot{V}_{O_2}$</th><th>f_S</th><th>RQ</th></tr>
<tr><td>Unstable</td><td>3.0 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.110</td><td>0.9</td></tr>
<tr><td>Critical</td><td>2.5 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.260</td><td>0.9</td></tr>
<tr><th colspan="2">BASE DISCHARGE RATE
h_u</th><th colspan="2">BASE RECOVERY RATE
h_c</th><th colspan="2">ATTENTION CAPACITY
κ</th></tr>
<tr><td colspan="2">$(48\ \text{hr})^{-1}$</td><td colspan="2">$(24\ \text{hr})^{-1}$</td><td colspan="2">1</td></tr>
</table>

Table 5.5. Patient-model specification for Mr. Dickinson.
We specify a patient model using these thirteen parameters. The parameters are subsequently compiled into a form suitable for processing SIMON's queries (see Section 5.4.2).

5.4.2 Computational issues

The parameters that describe Mr. Dickinson—or another ICU patient—must be converted into a form suitable for processing SIMON's queries (see Figure 5.2). This transformation involves a series of automated steps. The first step is performed once: The patient description is compiled into an MDP of the type shown in Figure 5.8 (see Section 5.3). The remaining steps occur in response to each query from SIMON:

- *MDP compilation.* The MDP, along with the patient parameters κ and α (see Section 5.3.6) and the query parameter T, is compiled via the methods presented in Sections 4.6.1 and 4.6.2. The result is a set of utility functions $\{u_1(x), u_2(x), u_3(x), u_4(x)\}$, one for each nonterminating patient state (see Table 5.2).[8]

---

7. I note, in passing, that VentPlan recommends treating the unstable patient with 25-percent oxygen, and the critical patient with 50-percent oxygen, with the other treatment parameters virtually unchanged. An increased inspired-oxygen fraction helps the lungs to compensate for the loss of blood oxygenation, at the expense of introducing new toxicities associated with abnormally high oxygen concentrations.

8. We do not keep *stable* or *dead* patients in the ICU; therefore, we have no use for utility functions corresponding to these patient states.

- *Utility-function selection.* The current state i is ascertained, and the appropriate utility function $u_i(x)$ is selected.
- *Query processing.* The function $u_i(x)$ is differentiated, such that the function $\lambda_i(x)$ is obtained. SIMON then uses $\lambda_i(x)$ and κ to process the query, as follows (see Section 4.5):
    1. A query (x, T) is answered with $\lambda_i(x)$ for $0 \leq x < \kappa$, or with 0 for $x \geq \kappa$.
    2. A query (λ, T) is answered with $\lambda_i^{-1}(\lambda)$ for $\lambda_i(\kappa) < \lambda < \lambda_i(0)$, with 0 for $\lambda \geq \lambda_i(0)$, or with κ for $\lambda \leq \lambda_i(\kappa)$.

The MDP-compilation step is by far the most expensive computationally. We can achieve significant computational savings by specifying, in advance, values of T that can be allowed in queries. The utility functions can then be precomputed for each possible value of T, prior to the real-time processing of queries.

5.5 Summary

I have built a prototype application, SIMON, that implements the alarm methodology of Chapters 3 and 4. SIMON was designed to meet the important challenge of managing a busy ICU by allocating scarce attention resources intelligently. Currently, SIMON is equipped to produce alarm signals for collections of simulated ICU patients.

To produce patient models that represent realistic clinical scenarios, I designed an abstract patient model that was based, in large part, on the outputs of a validated prototype ventilator-management advisor. I demonstrated that I can create a simulated patient by instantiating this abstract model with a small set of clinically meaningful parameters.

Given several such patients, SIMON produces alarm signals that allocate limited attention resources to where they are most needed. To characterize the behavior of SIMON, I conducted a series of controlled experiments, which I describe in Chapter 6.

Chapter 6

Alarm Signals for a Busy Intensive-Care Unit

To meet the need for sensible alerts, I developed a decision-theoretic alarm methodology that embodies the related concepts of process priority and the amount of attention that a process merits. I created a prototype, SIMON, that applies this methodology to the problem of monitoring patients in a busy intensive-care unit (ICU), and I evaluated the methodology by running SIMON on a number of simulated ICU scenarios. I report on the evaluation process and results in this chapter.

My evaluation proceeded along the dimensions of validity and computability. **Validity** is concerned with correctness; the relevant question is whether SIMON produces clinically sensible outputs, given inputs that are clinically sensible. In Sections 6.1 and 6.2, I show how I answered this question by investigating how SIMON's alarm signals respond to controlled changes in the patient population used as input.

Computability is concerned with scalability; it asks how well a methodology adapts when it moves from small to large problems. We saw, in Section 2.5.2, that the overall state space grows exponentially with the number of patients in a given collection. In Section 6.3, I report how I demonstrated the scalability of my approach by applying SIMON to a simulated ICU that contains numerous patients.

6.1 Case 1: Comparing different states

Our first case study covers four patients in an ICU who have the same illness as Mr. Dickinson (Table 6.1), our heart-attack patient from Chapter 5. Each patient occupies a different state, but the patients are otherwise identical. I refer to these patients using numbers that describe their current state (Table 6.2).

<table>
<tr><th rowspan="2">PHYSIOLOGIC STATE</th><th colspan="5">PHYSIOLOGIC PARAMETERS</th></tr>
<tr><th>Q</th><th>V_{ds}</th><th>$\dot{V}_{O_2}$</th><th>f_S</th><th>RQ</th></tr>
<tr><td>Unstable</td><td>3.0 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.110</td><td>0.9</td></tr>
<tr><td>Critical</td><td>2.5 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.260</td><td>0.9</td></tr>
<tr><td colspan="2">BASE DISCHARGE RATE h_u</td><td colspan="2">BASE RECOVERY RATE h_c</td><td colspan="2">ATTENTION CAPACITY κ</td></tr>
<tr><td colspan="2">$(48\ \text{hr})^{-1}$</td><td colspan="2">$(24\ \text{hr})^{-1}$</td><td colspan="2">1</td></tr>
</table>

Table 6.1. Patient-model specification.
Shown are parameters that describe the four patients used in my first set of experiments. I used variations of this patient-model specification in the later case studies.

PATIENT NUMBER	CURRENT STATE (PHYSIOLOGY; TREATMENT)
1	*unstable; maintenance*
2	*critical; maintenance*
3	*unstable; aggressive*
4	*critical; aggressive*

Table 6.2. Patient-numbering scheme (case 1).
The four patients in this case study are numbered according their current state; the numbering scheme parallels that of the patient model in Figure 6.1.

Let us see how SIMON allocates limited attention resources to these patients. We shall find it particularly useful to visualize trends in SIMON's outputs, trends that occur when the query values T (the attention horizon) and C (the total available attention) are varied.

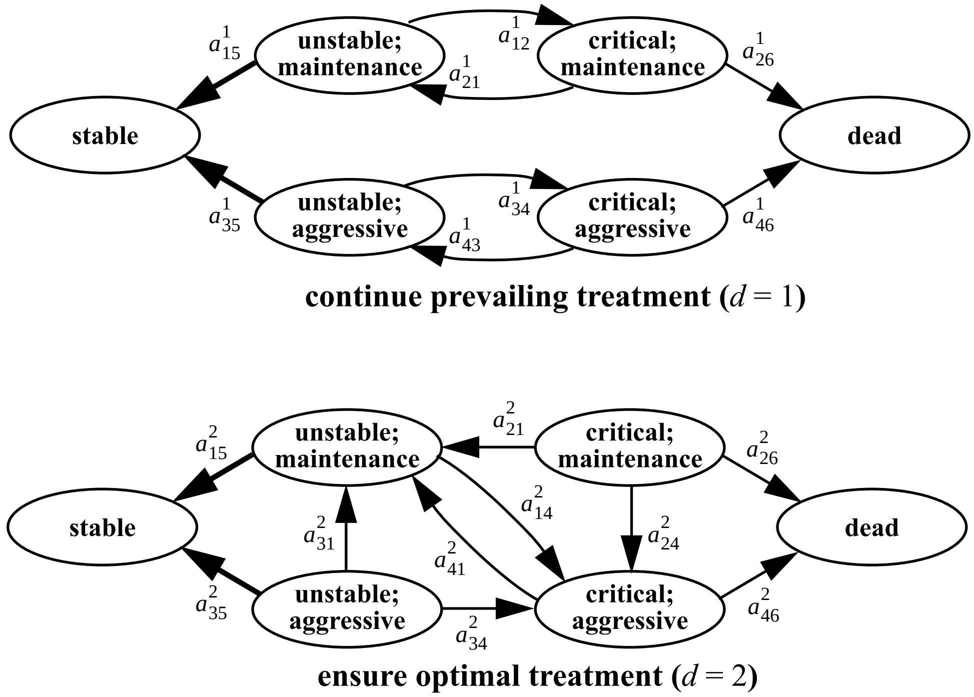


Figure 6.1. Physiologic states with persistent treatment effects. The abstract patient model of Figure 5.8 is reproduced here for the reader's convenience. The states are indexed: (1) *unstable-maintenance;* (2) *critical-maintenance;* (3) *unstable-aggressive;* (4) *critical-aggressive;* (5) *stable;* and (6) *dead.*

6.1.1 Activation prices

Given an attention horizon T, there is an activation price $\lambda_j(0)$ associated with each patient j. I computed the activation prices $\lambda_j(0)$ as function of T for our four patients, and plotted the results in Figure 6.2. For virtually all values of T, we observe that the activation price of patient 2 greatly exceeds those of the other patients. For patients 1, 3, and 4, we observe that $\lambda_3(0) > \lambda_4(0) > \lambda_1(0)$.

The activation price of a process determines the priority that that process commands in accessing scarce attention resources. In our case scenario, it is patient 2, the improperly treated critical patient, who receives top priority in this regard. As additional attention resources become available, the other patients—in the order 3, 4, and 1—become eligible to receive them. In general, improperly treated patients have priorities higher than those

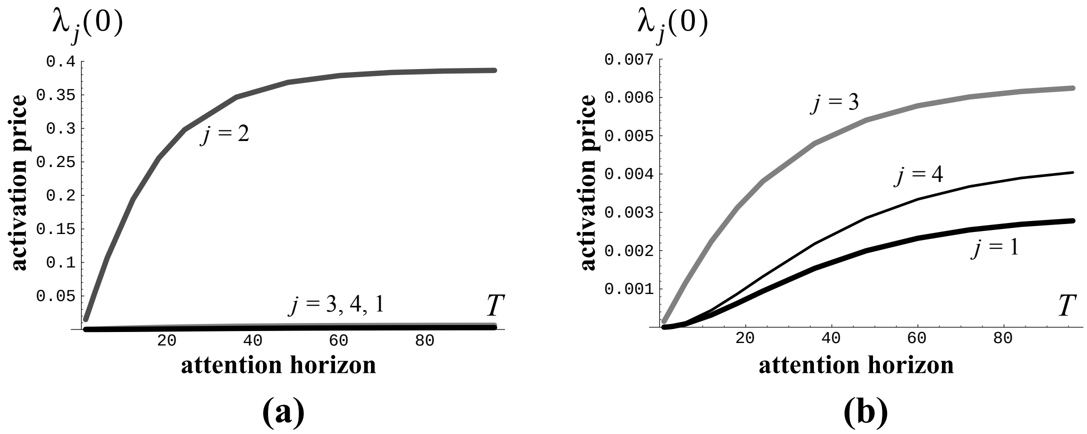


Figure 6.2. Activation prices versus *T* (case 1).

In (a), we see that $\lambda_2(0)$ greatly exceeds $\lambda_j(0)$ of the other patients, for virtually all values of T. The activation prices of the other patients are more easily compared in (b). Note that, in general, $\lambda_j(0)$ increases with T, confirming the intuitive idea that the effects of attention increase with the duration for which that attention is applied.

of properly treated patients; within each category, the critical patient has priority higher than that of the unstable patient.

6.1.2 Attention allocations

In contrast to the activation prices, the attention allocations depend not only on the attention horizon T, but also on the available attention C. Given a horizon T and attention amounts $C_1 > C_2$, we expect that, for each patient j, $x_j^*(C_1) \geq x_j^*(C_2)$. Let us witness this trend, and investigate other trends in the attention allocations for our case scenario.

6.1.2.1 Holding *T* and varying *C*

Plots of the attention allocations, as a function of the total available attention C, are shown for various values of T in Figure 6.3. Two common trends prevail in these plots. First, note that with increasing C, the patients are activated in the order 2, 3, 4, and 1—as predicted by their activation prices.[1] Second, for all values of C, it appears that $x_2^*(C) \geq x_3^*(C) \geq x_4^*(C) \geq x_1^*(C)$. This latter trend does not hold in not general: Section 6.3 describes a case that exhibits attention allocations that do not parallel the activation prices.

1. *Activated* patients command nonzero amounts of attention.

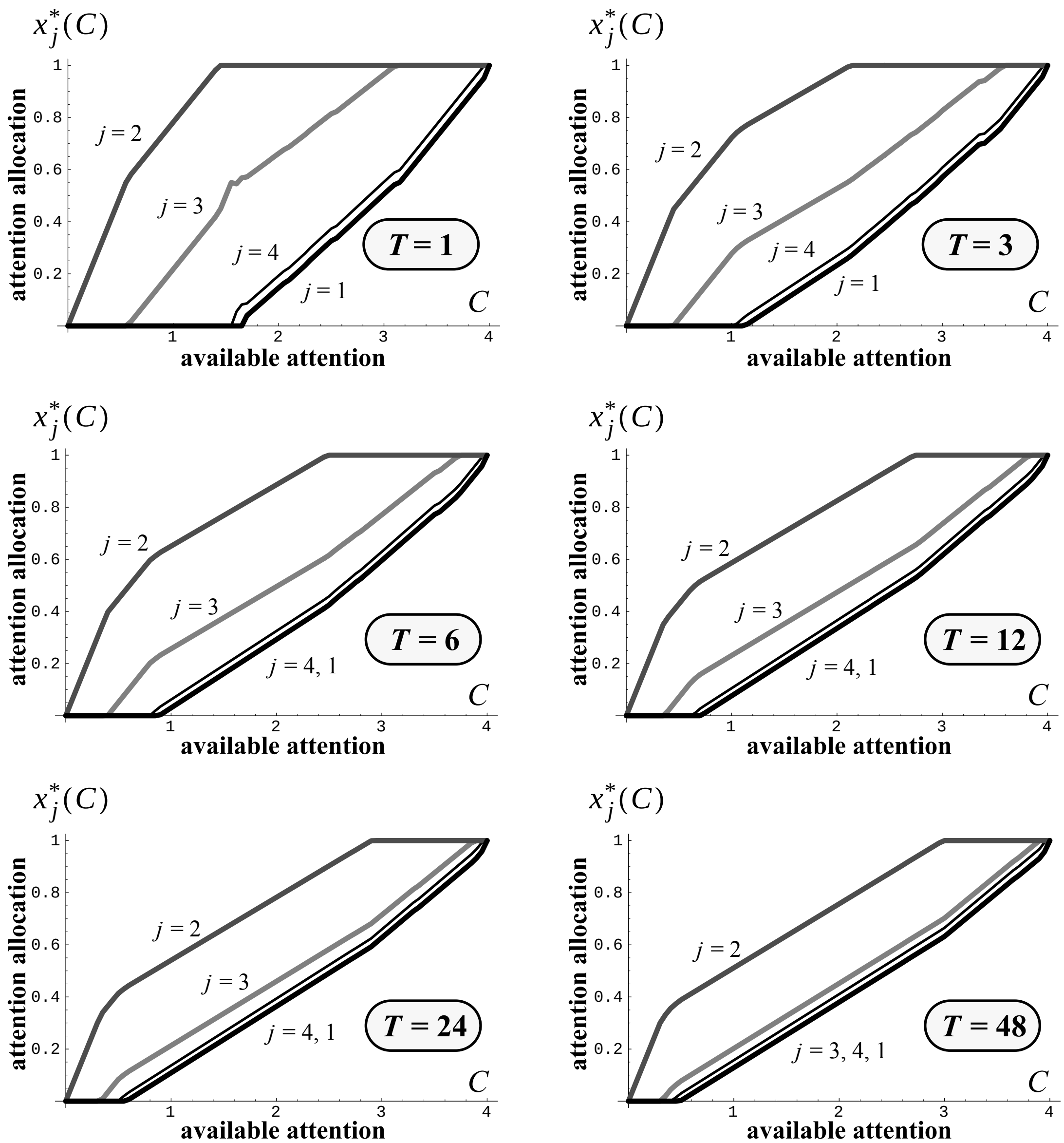


Figure 6.3. Attention allocations versus C for fixed T (case 1).
The attention allocations $x_j^*(C)$ for case 1 are plotted for $T = 1, 3, 6, 12, 24$, and 48 hours.

Trends also occur *between* the plots in Figure 6.3. To highlight one of them, we focus on the attention allocations for $C = 2$ in each plot. We observe that, when the attention horizon is short (i.e. 1 hour), $x_2^* = 1$, $x_3^* \approx 0.65$, $x_4^* \approx 0.20$, and $x_1^* \approx 0.15$. That is, patients 2 and 3 merit about one-half and one-third of the available attention, respectively, with the remaining one-sixth divided between patients 4 and 1. As T increases, however, this preferential assignment of attention becomes less extreme. At $T = 48$ hours, for example, patient 2 commands about 37 percent of the available attention, while the remaining patients command about 21 percent each.

6.1.2.2 Holding *C* and varying *T*

We can appreciate the smoothing of attention allocations that occurs with increasing attention horizons by examining Figure 6.4, which shows plots of attention allocations as a function of T, for various values of C. As $T \to \infty$, the attention allocations appear to converge toward a stable set of values. Going in the other direction, as T approaches 0, the attention distribution polarizes, and the available attention becomes concentrated on the processes that have the highest activation prices (see Section 4.6.4).

6.1.3 Global measures

We can gain insight into a collection of processes, taken as a whole, by examining various global measures that result when we compute alarm signals for the individual processes. Let us see what we can learn about the collection of patients from case 1.

6.1.3.1 Equilibrium prices

The equilibrium price $\lambda(C)$ is the sensitivity of total utility to small changes in available attention in the neighborhood of C (see Section 4.4.5). In Figure 6.5, I plot $\lambda(C)$ for various values of the attention horizon T. At $T = 1$ hour, $\lambda(C)$ bottoms out at approximately one-half of an attention unit; when T is increased to 48 hours, the point of negligible incremental returns appears to occur at one-third of an attention unit.

Practical consequences arise from the shape of the $\lambda(C)$ curves. Suppose that there is an attention amount $C = C'$, such that the benefit of any extra attention beyond C' is negligible. Then any extra attention beyond C' may be expended more wisely on other tasks—for example, on taking care of another collection of patients. Excluding the possibility of other tasks, we can say that the specific distribution of any additional attention beyond C' is practically irrelevant. In other words, once each patient j receives $x_j^*(C')$ attention units, any extra attention units may be distributed arbitrarily with little consequence.

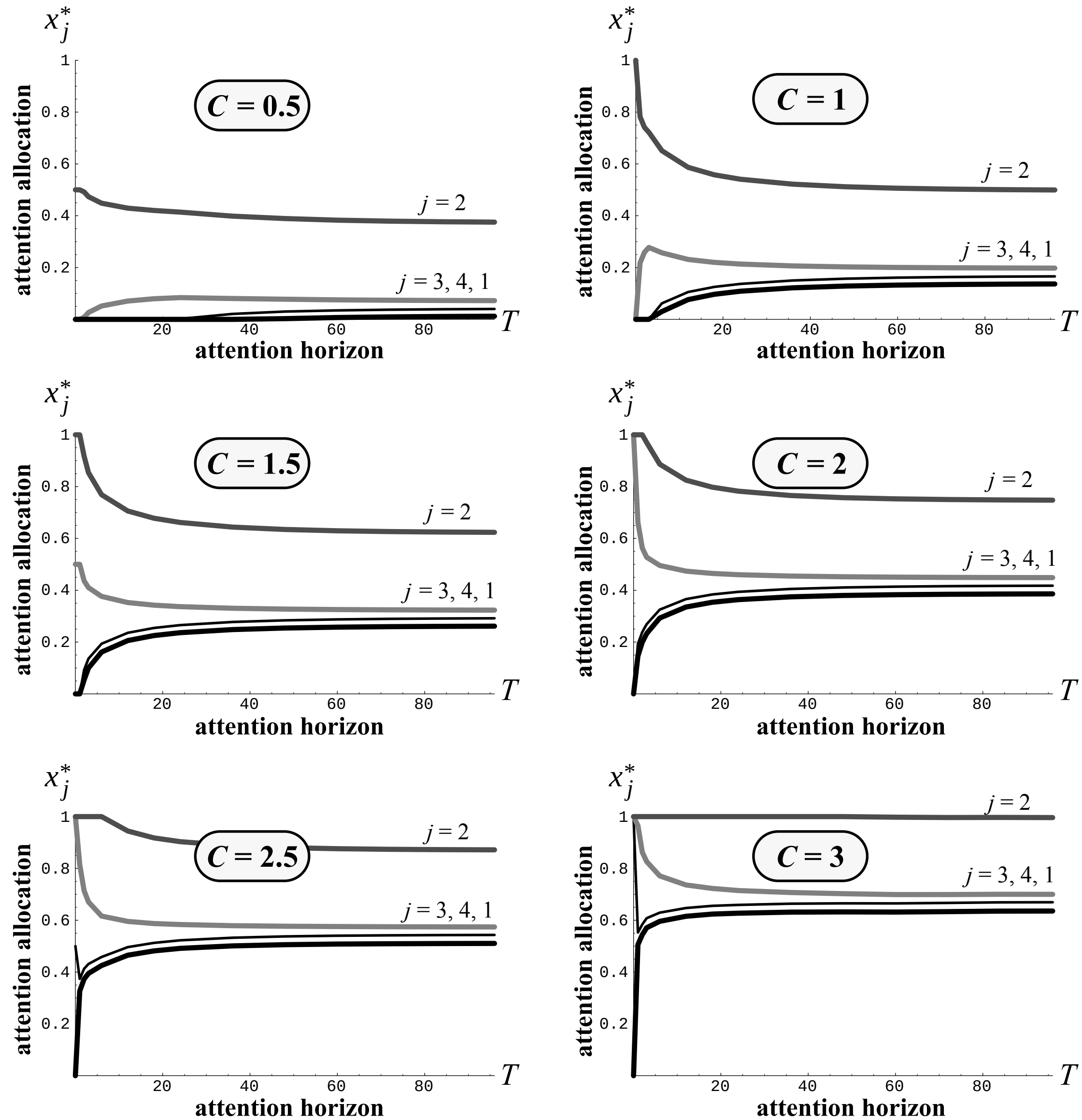


Figure 6.4. Attention allocations versus *T* for fixed *C* (case 1).

The attention allocations $x_j^*(C)$ for case 1, as a function of T, are plotted for $C = 0.5$, 1, 1.5, 2, 2.5, and 3 attention units.

6.1.3.2 Total utilities

I plot the total utility $U(C)$ for various values of T in Figure 6.6. In each plot, the value of U appears to plateau at approximately 1.701 for large values of C. However, the minimum total utility $U(0)$ descends lower for larger T. This trend is consistent with the increase in λ values and other benefit measures as the attention horizon increases.

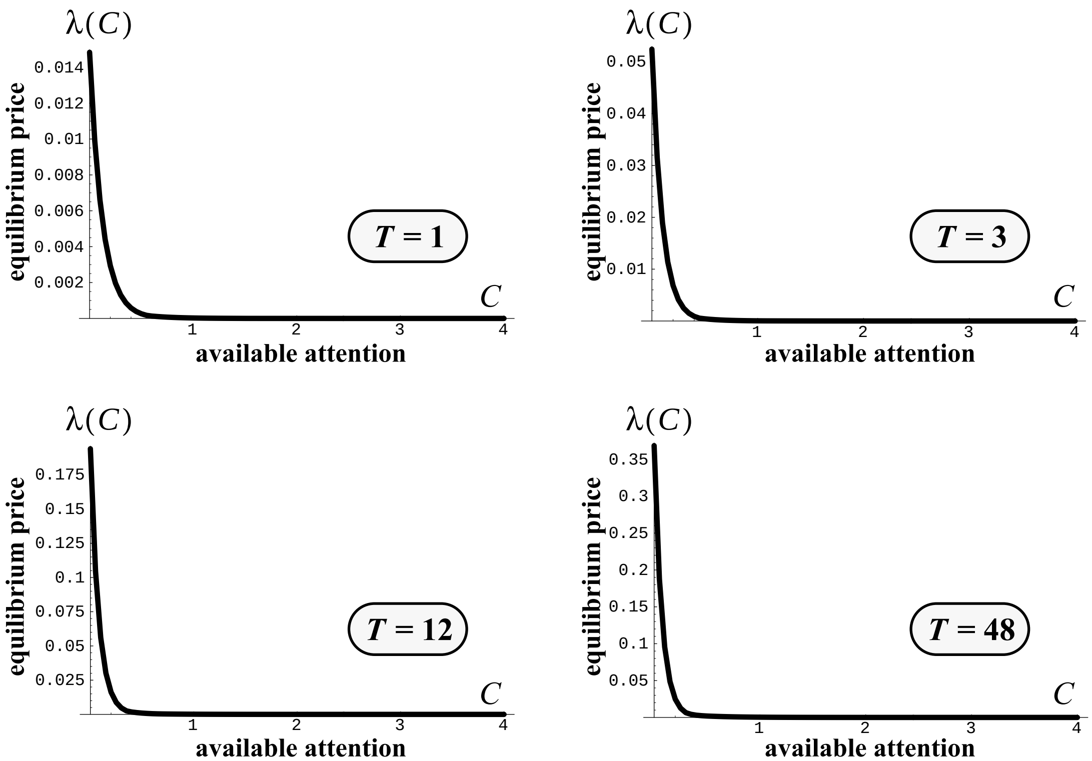


Figure 6.5. Equilibrium price versus C for fixed T (case 1).

The equilibrium prices $\lambda(C)$ for case 1 are plotted for $T = 1$, 3, 12, and 48 hours. The point of negligible incremental returns occurs at a fraction of an attention unit—approximately $1/2$ for $T = 1$ hour and $1/3$ for $T = 48$ hours. As expected, the λ values increase with T, because the effects of attention increase with the duration of attention.

6.2 Varying individual patient parameters

In the next several case studies, we examine how changes in individual patient parameters affect the patient's ability to command scarce attention resources.

6.2.1 Experimental protocol

Each of the case studies occurs in an ICU that contains only two patients. One patient (patient 1) is identical to Mr. Dickinson (Table 6.1); the other patient (patient 2) differs from the first by a single parameter value. The two patients are in the same state (see Table 6.2). The available attention—and thus the total attention allocated—is one-half of the sum of both patients' attention capacities. The attention horizon is $T = 12$ hours.

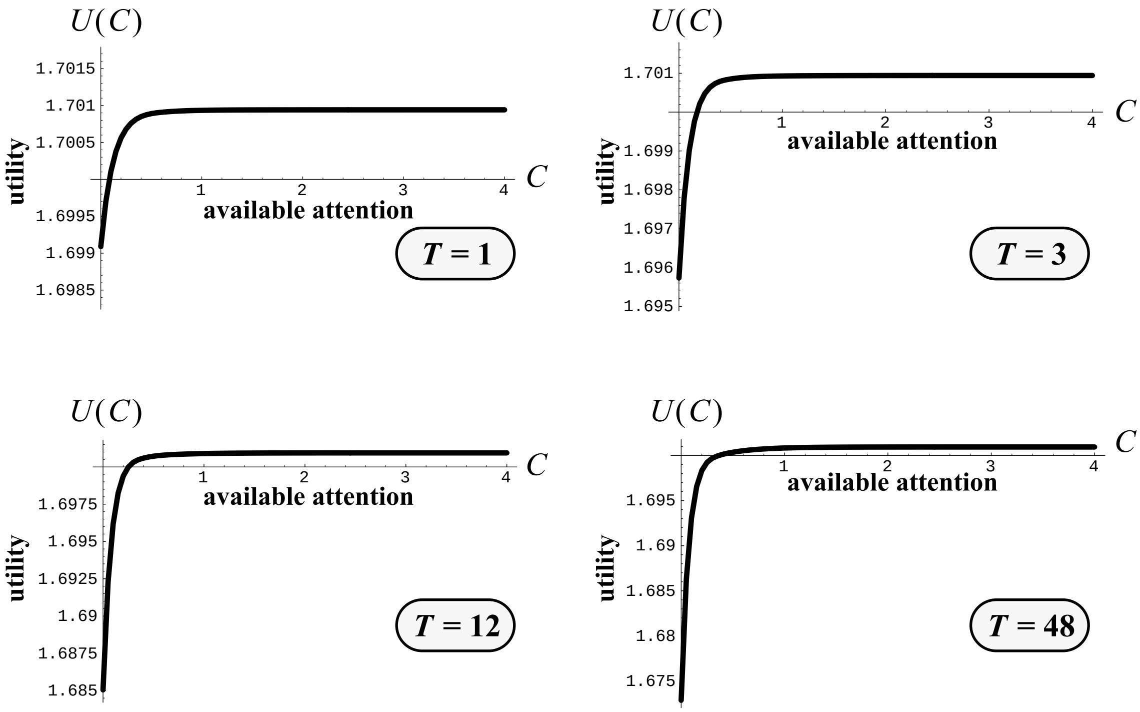


Figure 6.6. Total utility versus C for fixed T (case 1).
The total utilities $U(C)$ for case 1 are plotted for $T = 1$, 3, 12, and 48 hours. The utility plateaus occur at the values of C where the equilibrium prices of Figure 6.5 bottom out.

We can think of patient 1 as a control subject and patient 2 as an experimental subject. Instead of noting separate attention allocations x_1^* and x_2^*—as we did in Section 6.1—we observe the **attention ratio** of the two patients, the ratio of x_2^* to x_1^*. We use ζ_i to denote the attention ratio x_2^*/x_1^* when both patients are in state i. When the two patients are identical, $\zeta_i = 1$. Our goal is to see how ζ_i is affected by specific changes in the characteristics of patient 2.

6.2.2 Case 2: Varying the attention capacity

I begin by varying the attention capacity κ of the experimental patient. I fix the attention capacity of the control patient at 1. Thus, for a given κ, the experimental patient requires κ times the amount of attention that the control patient needs, to be treated in the same optimal manner.

Figure 6.7 plots, for each state i, the attention ratio $\zeta_i = x_2^*/x_1^*$ as a function of κ. In each plot, we observe that ζ_i increases with κ in an approximate, linear fashion. We are not surprised to witness that $\zeta_i(1) = 1$, although it is interesting that $\zeta_i(\kappa) \neq \kappa$ in general. In any event, these results, taken together, suggest that a patient's ability to command attention is directly proportional to the effort required for that patient to be treated optimally.

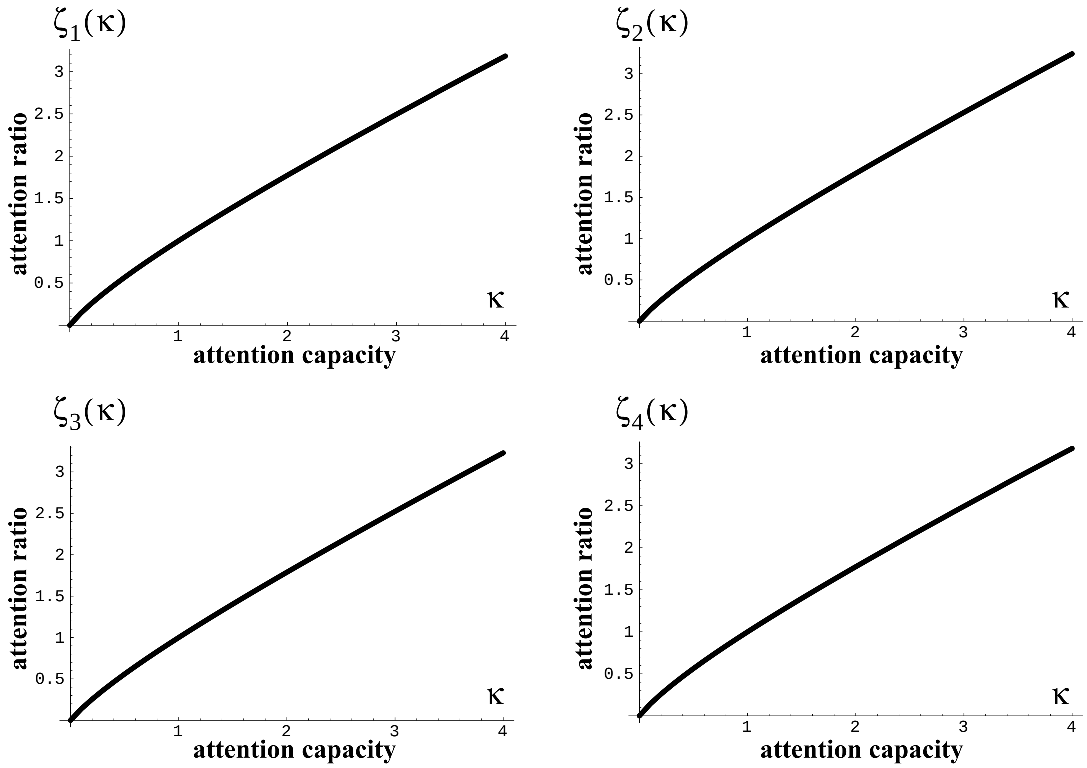


Figure 6.7. Attention ratios for experimental κ values (case 2).
The attention ratio ζ_i is plotted as a function of κ, the experimental patient's attention capacity, for each state i. The experimental patient's ability to command attention, compared to that of the control patient, increases approximately linearly with κ.

6.2.3 Case 3: Varying healing-rate parameters

In the next series of experiments, I manipulate the healing-rate parameters h_{u} and h_{c}. Recall that h_{u} is the base discharge rate, the maximum rate at which the patient can improve from the unstable state, whereas h_{c} is the base recovery rate, the maximum rate at which the patient can improve from the critical state (see Section 5.3.4).

6.2.3.1 Varying the base discharge rate

I begin by varying the base discharge rate h_u of the experimental patient. I often describe h_u in terms of the **base discharge time** $t_u = (h_u)^{-1}$. The base discharge time of the control patient is held constant at 48 hours. Figure 6.8 plots, for each state i, the attention ratio ζ_i as a function of t_u.

In all these plots, ζ_i falls below 1 as t_u increases above 48 hours. In other words, when a patient's ability to improve from the unstable state is compromised, his *pull* on the available attention is weakened. We might be tempted to object to this result on the notion that we should be more concerned about patients who are more sick. However, this result does not tell us to shift our concern away from sick patients; it merely suggests that patients who stand to benefit most from additional attention should get that additional attention.

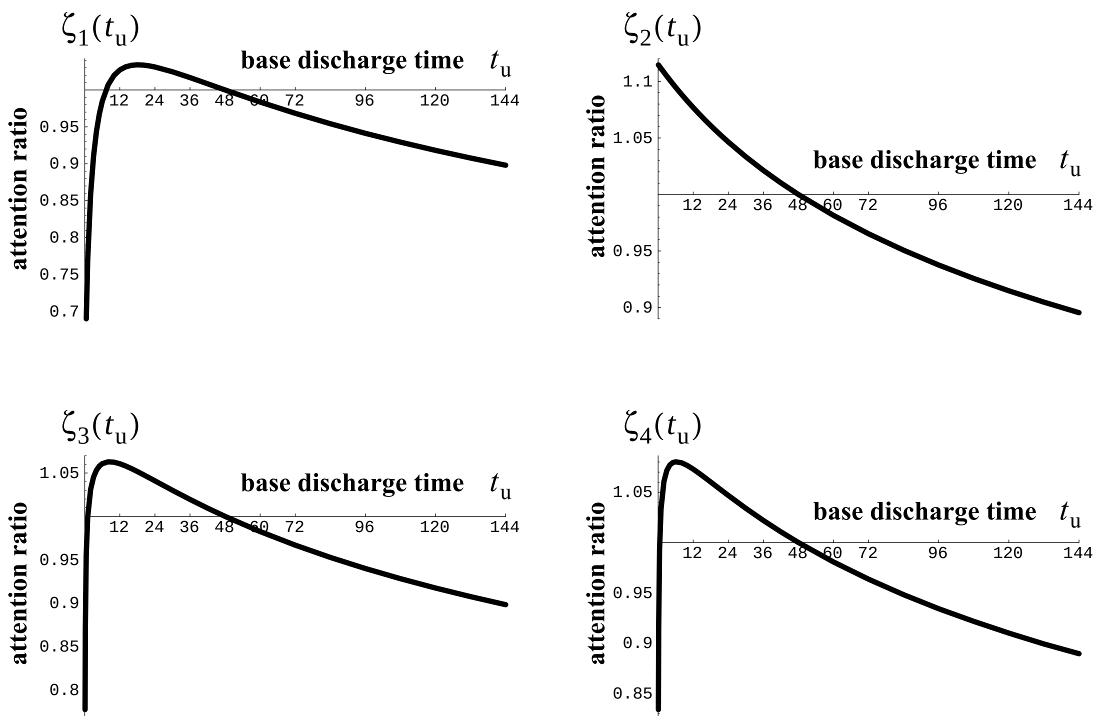


Figure 6.8. Attention ratios for experimental t_u values (case 3).
For each state i, the attention ratio ζ_i is plotted as a function of t_u, the base discharge time of patient 2. For $t_u \geq 48$ hours, ζ_i decreases below 1 as t_u increases. This trend continues for $t_u < 48$ hours, but the trend is reversed at low values of t_u for $i = 1$, 3, and 4.

Interesting phenomena occur when I decrease the base discharge time of patient 2 below 48 hours. For each state i, ζ_i increases as t_u is decreased from 48 hours to 24 hours; we expect this behavior because a decreased healing time increases the effectiveness of attention, and therefore the patient's pull on the available attention. However, for $i = 1$, 3, and 4, ζ_i stops increasing at a certain point (roughly $t_u = 15$ hours for $i = 1$ and $t_u = 6$ hours for $i = 2$, 3), and then *decreases* as we continue to lower t_u. To understand this behavior, we again invoke the normative idea that patients who benefit the most from additional attention should get that additional attention. At low values of t_u, where ζ_i decreases with decreasing t_u, small amounts of attention have an extraordinarily positive effect on the patient's ability to improve from the unstable state to the stable state. But once he is in the stable state, the patient needs no further attention.

It is interesting that the reversal of ζ_i does not occur for $i = 2$—the improperly treated critical state. To gain insight into this exception, we note that the risk of death from state 2 is generally much higher than that from the other states. We can surmise that the condition of patients in state 2 is so severe that any attention conferred serves mainly to prevent death, rather than to facilitate discharge. Lower values of t_u increase the pull on available attention because patients who are discharged faster are in less danger of dying.

6.2.3.2 Varying the recovery rate

We now study the effects of varying the experimental patient's base recovery rate h_c. I often refer to the **base recovery time** $t_c = (h_c)^{-1}$. We hold the control patient's base recovery time constant at 24 hours. Figure 6.9 plots, for each state i, the attention ratio ζ_i as a function of t_c.

For $i = 1$, 2, and 4, we note that, as t_c decreases below 48 hours, ζ_i increases, but reverses direction for low values of t_c. The initial increase in ζ_i can be explained by the notion that a critically ill patient who responds more quickly to proper treatment commands more attention. At low values of t_c, the trend reverses because small amounts of attention have an extraordinarily positive effect on the patient's ability to improve from a critical state to an unstable state, from which the patient can most readily leave the ICU alive.

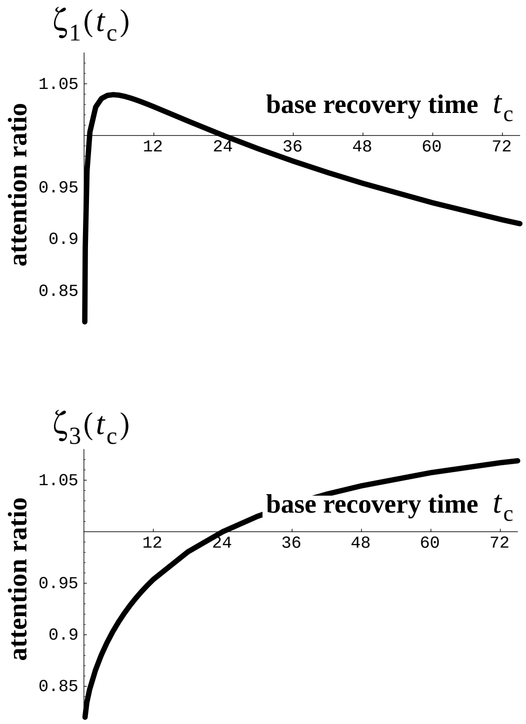


Figure 6.9. Attention ratios for experimental t_c values (case 3).
For each state *i,* the attention ratio ζ_i is plotted as a function of t_c, the base recovery time of patient 2. For $i = 1$, 2, and 4, ζ_i increases as t_c decreases below 48 hours, but the trend is reversed at low values of t_c. For $i = 3$, ζ_i increases with increasing t_c.

The results for $i = 3$ are mystifying at first glance. Here, we see ζ_i *increasing* with t_c, in contrast with what we saw for $i = 1$, 2, and 4. We can readily distinguish state 3 from states 2 and 4: Only the latter are directly affected by changes in the recovery rate. The question, then, is how state 3 differs from state 1. Given that both states are unstable, why does state 3 attract more attention with increasing t_c, whereas state 1 generally attracts less attention with increasing t_c?

The key to this mystery lies in the mechanisms that translate attention into benefit. To isolate the dominant mechanisms of attention for states 1 and 3, we imagine that t_c is infinitely large—that is, a patient who reaches a critical state has no hope for recovery. Then, the importance of discharging an unstable patient quickly becomes paramount. From state 3, more attention always helps, because it facilitates a transition into state 1, from which a successful discharge can most readily occur.[2] In contrast, no amount of attention can improve the situation in state 1, because state 1 is already the most desirable state for our patient.

We can now appreciate the essential difference between states 1 and 3. In state 1, the primary effect of attention is to faciliate recovery from a subsequent critical state. A patient who recovers more quickly commands more attention, although beyond a certain recovery rate, the required attention decreases in response to continued increases in the recovery rate. In state 3, attention serves mainly to facilitate a transition into state 1, where the effective discharge rate is highest. A patient who is placed in greater jeapordy by a transition to a critical state has a greater need for such attention.

6.2.4 Case 4: Varying physiologic parameters

We have one final set of parameters to investigate: parameters that define the physiologic states themselves. Although there are 10 such parameters, it is not helpful to study the individual role of each, as real physiologic changes typically affect several parameters in a correlated manner. Our goal, instead, is to explore how meaningful variations in the characterization of a physiologic state affect a patient's pull on the available attention.

In each of the following experiments, I continue comparing the attention allocations of an experimental and a control patient. Our patients are still being monitored for some type of congestive heart failure (CHF), in which a decrease in cardiac output causes a increase in the physiologic shunt fraction. However, I use other control patients in place of Mr. Dickinson.

6.2.4.1 Varying the severity of the critical state

I begin by manipulating the critical state. Table 6.3 specifies our control patient for this series of experiments. I produced experimental patients by modifying the critical state as

2. Remember that the effective discharge rates for states 1 and 3 differ because the deterioration rate for state 3 exceeds that of state 1 (see Sections 5.3.3–5.3.4).

follows: I introduced various deviations ΔQ in the cardiac output Q, each in conjunction with an increase in shunt fraction $\Delta f_S = -0.28\ \Delta Q$. I computed, for each state i, the attention ratio ζ_i as a function of Q; the plots of the results are shown in Figure 6.10.

<table>
<tr><th rowspan="2">PHYSIOLOGIC STATE</th><th colspan="5">PHYSIOLOGIC PARAMETERS</th></tr>
<tr><th>Q</th><th>V_{ds}</th><th>$\dot{V}_{O_2}$</th><th>f_S</th><th>RQ</th></tr>
<tr><td>Unstable</td><td>3.00 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.110</td><td>0.9</td></tr>
<tr><td>Critical</td><td>2.25 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.190</td><td>0.9</td></tr>
<tr><th colspan="2">BASE DISCHARGE RATE
h_u</th><th colspan="2">BASE RECOVERY RATE
h_c</th><th colspan="2">ATTENTION CAPACITY
κ</th></tr>
<tr><td colspan="2">$(48\text{ hr})^{-1}$</td><td colspan="2">$(24\text{ hr})^{-1}$</td><td colspan="2">1</td></tr>
</table>

Table 6.3. Control patient for critical-state experiments (case 4).
This table specifies the control patient that I used to study the effects of varying the critical state. Experimental patients who have different severities of critical CHF are specified via a coordinated manipulation of the shaded parameters: A change in cardiac output ΔQ (measured in l/min) is accompanied by a change in shunt fraction $\Delta f_S = -0.28\ \Delta Q$.

The geometric irregularities in Figure 6.10 deserve explanation. I generated these plots, in contrast to the smooth plots of Figures 6.7 through 6.9, by varying parameters that affect patient dynamics in complex ways. The critical-state parameters determine the death rates from states 2 and 4, which in turn help to determine the effective recovery rates from these respective states. Thus, the definition of critical state affects four distinct transition rates in the process model of Figure 6.1. Furthermore, the discontinuities in these plots appear to stem from discontinuities inherent in VentPlan's quantitative models.[3]

Given these complications in the translation of physiologic-parameter effects, we are reassured by the gross trends that emerge from these experiments. In each of these plots, we note that ζ_i increases with decreasing Q. In other words, patients who are in more danger

3. VentPlan's utility measure does not respond smoothly to changes in the physiologic parameters. Sometimes, small parameter changes do not affect the utility. In other instances, a small change in input parameter can cause the utility to skip to a new value.

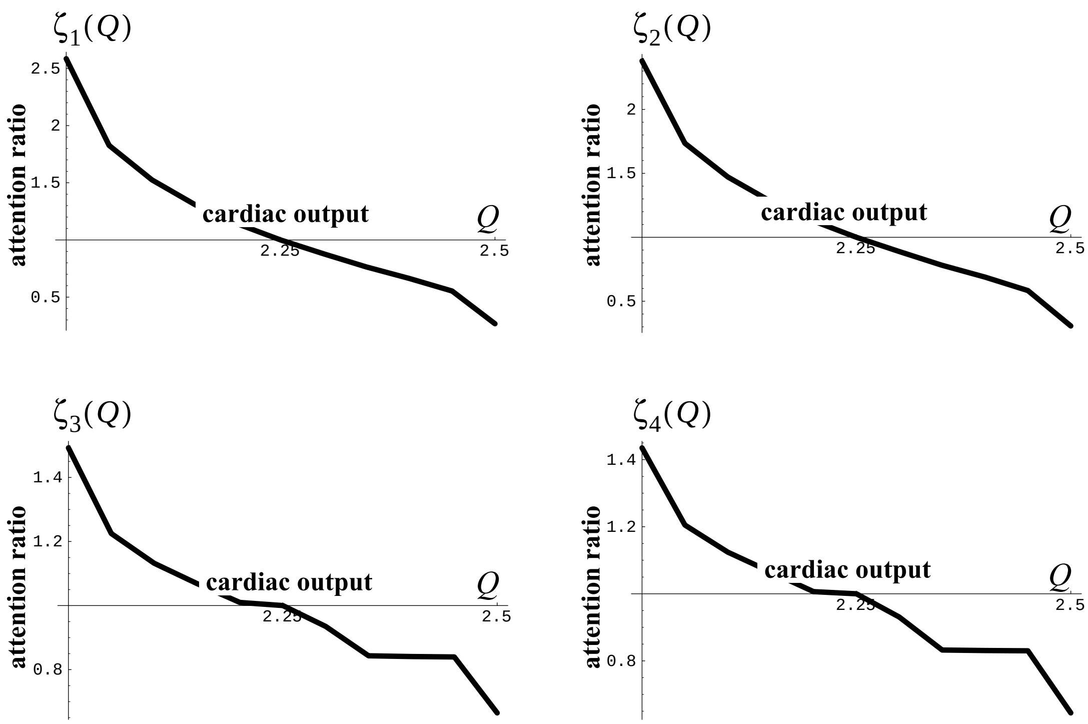


Figure 6.10. Attention ratios for experimental critical states (case 4). For each state i, the attention ratio ζ_i is plotted as a function of Q, the cardiac output of the critical state. Note that the shunt fraction f_S varies with Q; see Table 6.3. I generated the plots by computing ζ_i for Q varying in increments of 0.05 l/min.

of dying from a critical state command greater attention. This result confirms our intuition that more attention should be given to those patients who stand to lose more from neglect.

6.2.4.2 Varying the health of the unstable state

Let us now focus on the unstable state. Our control patient for this series of experiments is specified in Table 6.4. I produced experimental patients by modifying the unstable state: I introduced various deviations ΔQ in the cardiac output Q, each in conjunction with an increase in shunt fraction $\Delta f_S = -0.28\,\Delta Q$. I computed, for each state i, the attention ratio ζ_i as a function of Q; the results are plotted in Figure 6.11.

As in the previous set of experiments, these plots exhibit irregularities that arise from the complex relationship between physiologic-state parameters and various process-transition rates. Fortunately, we witness gross trends that, once again, make clinical sense. In each

Physiologic state	Physiologic parameters				
	Q	V_{ds}	$\dot{V}_{O_2}$	f_S	RQ
Unstable	**2.75 l/min**	150 ml	150 ml/min	**0.180**	0.9
Critical	2.40 l/min	150 ml	150 ml/min	0.260	0.9
Base discharge rate h_u		Base recovery rate h_c		Attention capacity κ	
$(48\text{ hr})^{-1}$		$(24\text{ hr})^{-1}$		1	

Table 6.4. Control patient for unstable-state experiments (case 4).
This table specifies the control patient that I used to study the effects of varying the unstable state. I generated experimental patients by adjusting the cardiac output Q and the shunt fraction f_S simultaneously: For a cardiac-output adjustment ΔQ (measured in liters per minute), I adjusted the shunt fraction by an amount $\Delta f_S = -0.28\ \Delta Q$.

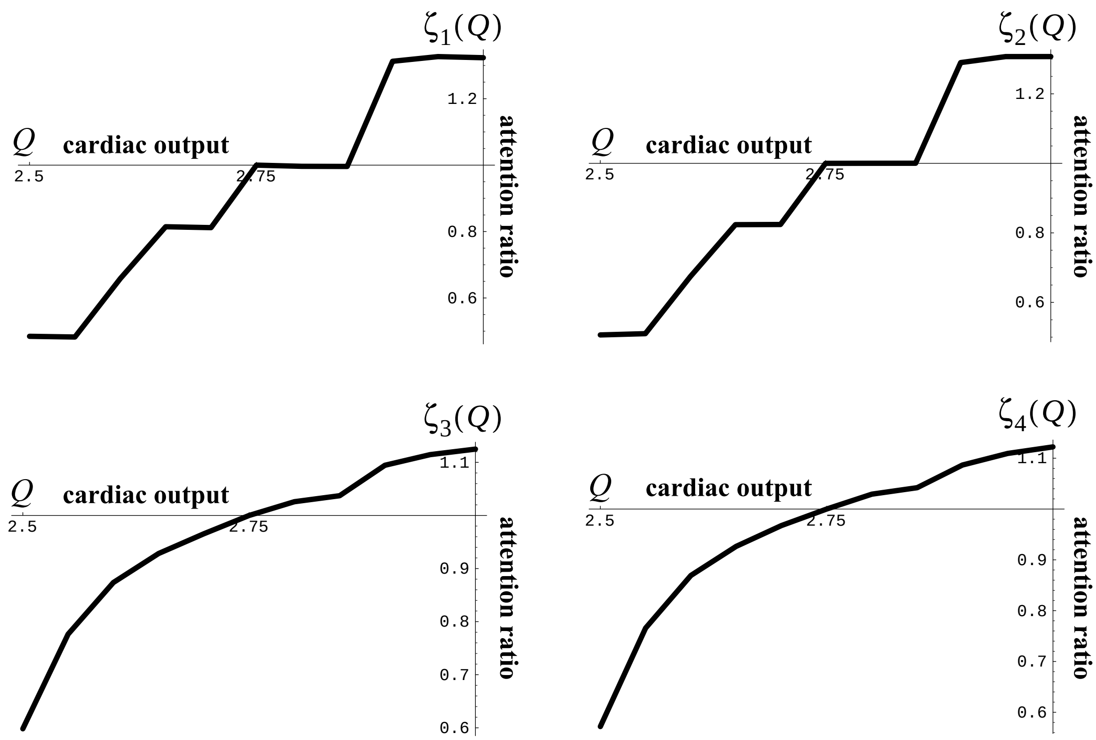


Figure 6.11. Attention ratios for experimental unstable states (case 4).
For each state i, the attention ratio ζ_i is plotted as a function of Q, the cardiac output of the unstable state. The shunt fraction f_S varies with Q, as noted in the caption of Table 6.4. I generated the plots by computing ζ_i for Q varying in increments of 0.05 liters per minute.

of these plots, ζ_i increases with increasing Q. In other words, patients who recover to a more healthy state—that is, an unstable state with a higher effective discharge rate—command greater attention. (Recall that healthier patients get discharged faster.) This result confirms the idea that available attention should be awarded preferentially to patients who stand to benefit the most from it.

6.3 Case 5: Simulating a large patient ward

After I tested SIMON's validity with a barrage of sensitivity experiments, I evaluated how well SIMON's alarm methodology scaled up to large-sized problems. In my next series of experiments, I used SIMON to generate alarm signals for a simulated ICU that contains 144 patients.

The computational features that are built into the theory of separable optimization give us reason to be optimistic about the outcome of this investigation (see Section 4.4). When we allocate attention among N processes, we are really performing a search in N-dimensional Euclidean space. Separable optimization reduces this problem to a search in one-dimensional Euclidean space, with a constant factor proportional to the magnitude of N. Thus, we should expect SIMON's time complexity to grow linearly with the problem size N.

6.3.1 Simulating 144 heterogenous patients

To perform a simulation that produces interesting output, we need a heterogenous collection of patients. I produced 144 different patients by permuting possible values of various patient parameters (Tables 6.5 and 6.6); the situation faced by each patient is a unique, but minor, variation of the CHF scenario faced by Mr. Dickinson. The total attention capacity is 288 attention units; two-thirds of the patients have an attention capacity greater than 1.

I spawned separate patient models for $T = 1$, 6, 12, and 24 hours, using the compilation techniques of Section 5.4.2. The entire patient-creation process took about 90 minutes on

Patient parameters		Permutable values
Description	Symbol	
Physiologic states	see Table 6.6	
Discharge rate	h_u	$(24\ \text{hr})^{-1}$, $(48\ \text{hr})^{-1}$
Recovery rate	h_c	$(24\ \text{hr})^{-1}$, $(36\ \text{hr})^{-1}$
Attention capacity	κ	1, 2, 3
Current state	i	1, 2, 3, 4

Table 6.5. Permutable patient parameters (case 5).
I generated 144 unique patients by permuting the parameter values shown in the table. The three physiologic-state definitions are shown separately in Table 6.6.

Parameter set	Unstable state		Critical state	
	Q	f_S	Q	f_S
1	3.0	0.110	2.5	0.260
2	2.7	0.210	2.2	0.260
3	3.0	0.110	2.7	0.210

Table 6.6. Permutable physiologic-state definitions (case 5).
This table specifies values for the three permutable physiologic-state definitions referred to in Table 6.5. Each of these definitions is crafted such that the critical condition is less desirable than the unstable condition. Not shown are the remaining physiologic-parameter values, common to both conditions: $V_{ds} = 150$ ml, $\dot{V}_{O_2} = 150$ ml/min, and $RQ = 0.9$.

a 180-MHz PowerPC 604e machine running MacOS 8.1. I then used SIMON to generate alarm signals for attention amounts C ranging from 0 to 288, in increments of 4. I repeated this procedure for each of the T values. The entire computation took under 10 minutes, or under 2.1 seconds for each set of alarm signals.

6.3.2 Attention allocations

To highlight different aspects of the attention allocations, I plot these allocations for different ranges of C. Figure 6.12 illustrates the results for $C \leq 36$. The patient groups labeled **A** and **B** exemplify an interesting and important phenomenon. Note that group **B** commands more attention than group **A** when attention resources are scarce ($C < 5$); that

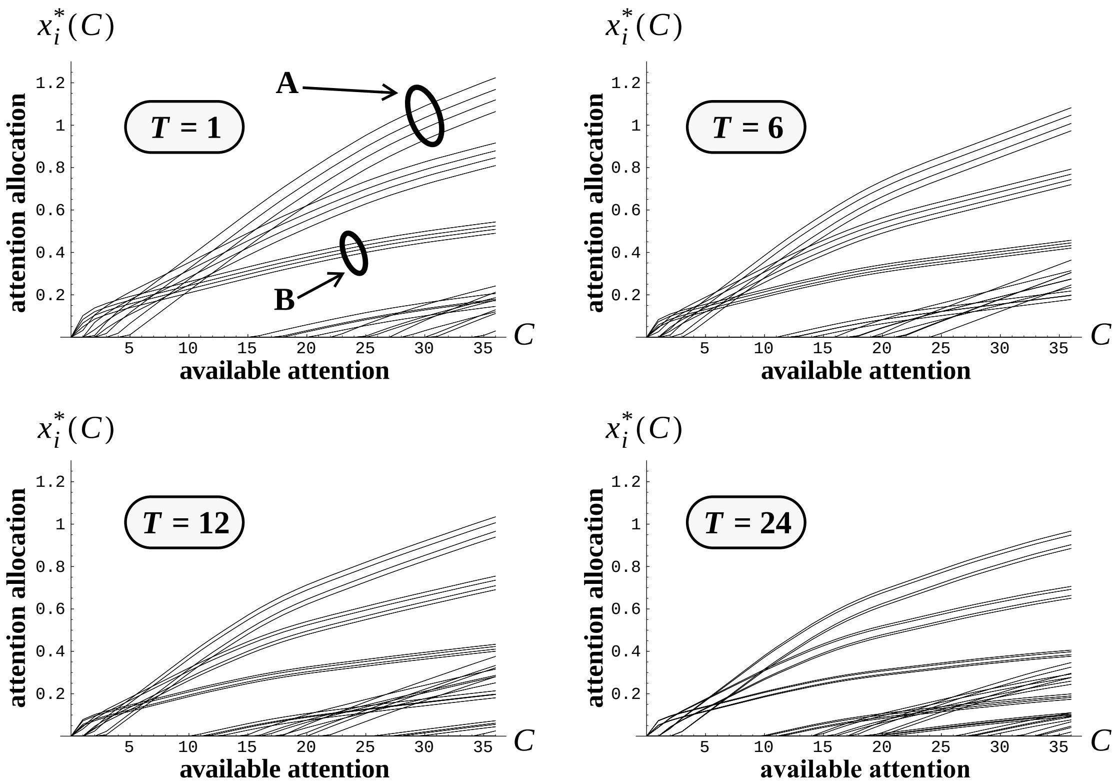


Figure 6.12. Attention allocations versus $C \le 36$ for fixed T (case 5). These plots summarize attention allocations for the 144-patient scenario for $C \le 36$. The attention allocations $x_i^*(C)$ are plotted for T = 1, 6, 12, and 24 hours. The patients labeled **A** and **B** are discussed in the text.

is, group **B** has priority higher than that of group **A.** However, for $C \ge 15$, group **A** commands a greater amount of attention than group **B**. This example demonstrates that higher-priority processes do not always need more attention than their lower-priority counterparts.

If we are observant, we may have legitimate concerns about the paucity of lines at $C = 36$: Only 24 lines can be discerned for the 144 patients. It turns out that this graphical observation is explained in part by the results, in part by artifact. At $C = 36$, only 72 of the 144 patients are activated; the remaining patients receive 0 attention units. Of these 72 patients, Mathematica's plotting facilities can recognize only 24 as unique. We shall encounter similar plotting artifacts in subsequent result displays.

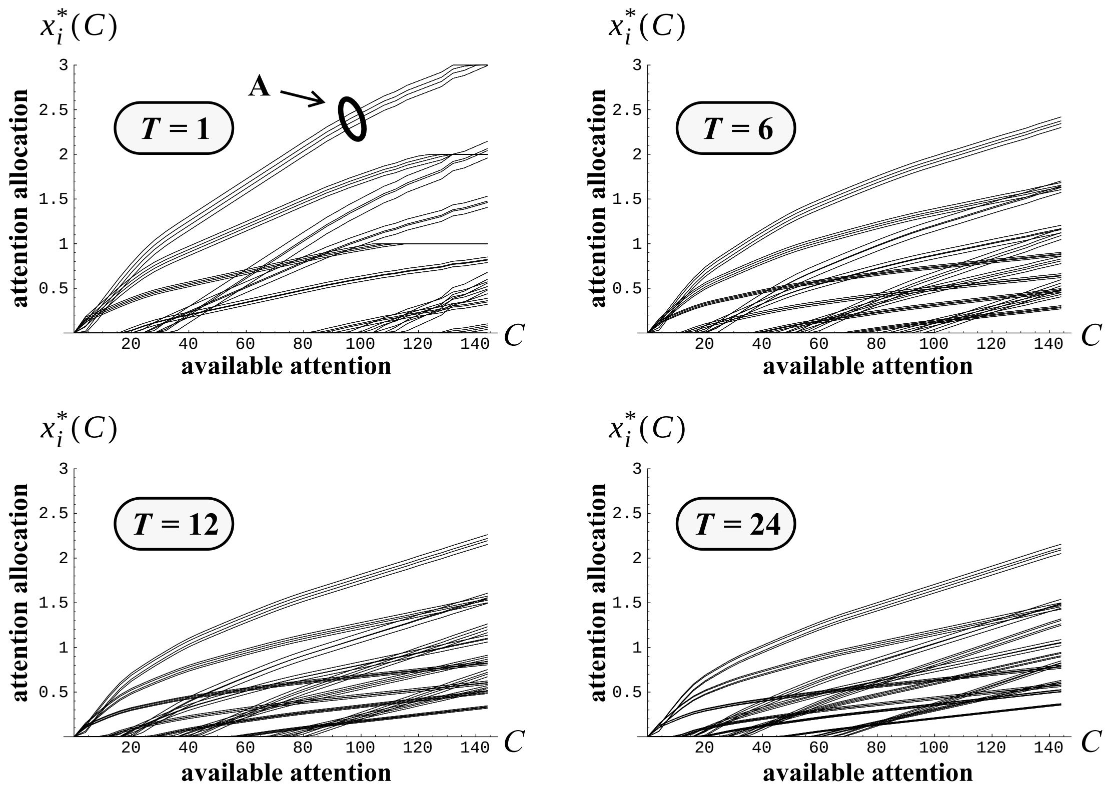


Figure 6.13. Attention allocations versus $C \leq 144$ for fixed T (case 5).
Attention allocations $x_i^*(C)$ for the 144-patient scenario for $C \leq 144$ are plotted for $T = 1$, 6, 12, and 24 hours. The patients labeled **A** are discussed in the text.

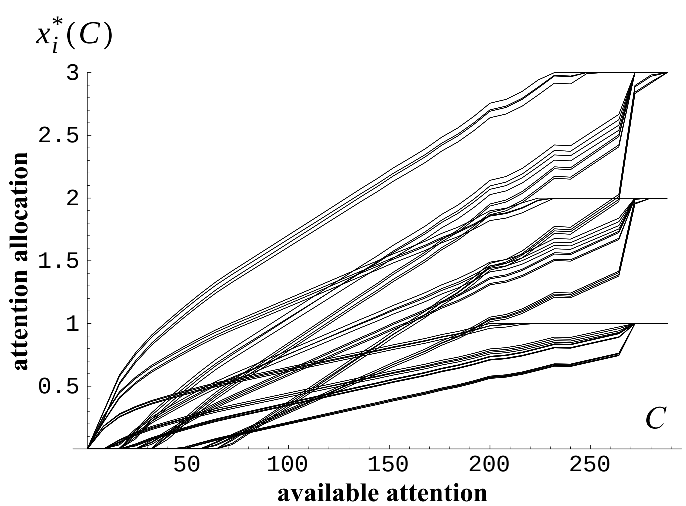


Figure 6.14. Attention allocations versus $C \leq 288$ for $T = 24$ (case 5).
Displayed are the results of the 144-patient scenario for $C \leq 288$, with T set at 24 hours.

The results for $C \leq 144$ are shown in Figure 6.13. Not surprisingly, the patients who receive no attention at $C = 36$ become activated at higher values of C. Note that group **A**'s attention allocations continue to climb above those of the other patients. The explanation of this behavior is easiest to discern in the $T = 1$ plot: The patients in group **A** have an attention capacity $\kappa = 3$, in contrast with patients whose attention capacities are 1 or 2. Although group **A**'s greater attention capacity does not automatically earn a higher priority (see Figure 6.12), these patients have an increasing pull on the available attention, as more attention becomes available.

To demonstrate how the attention allocation for each patient approaches his attention capacity with increasing C, I produced a plot of attention allocations for the entire range of C (Figure 6.14).

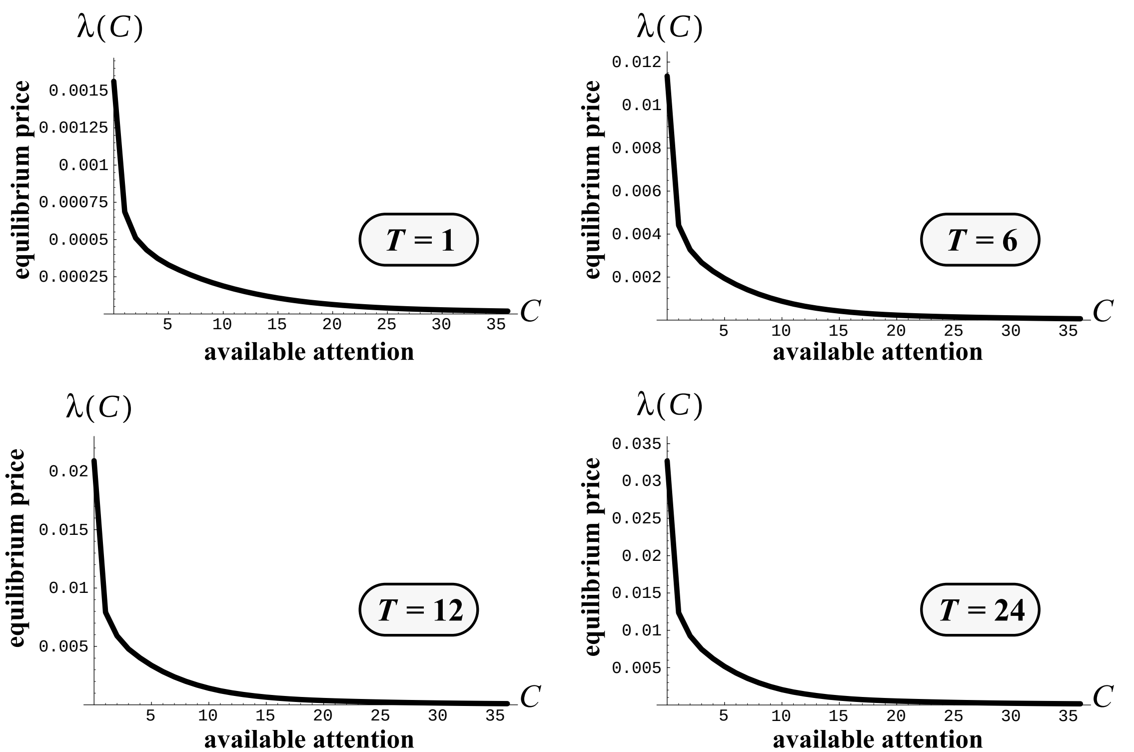


Figure 6.15. Equilibrium prices versus C for fixed T (case 5).
Equilibrium prices $\lambda(C)$ for case 5 are plotted for $T = 1$, 6, 12, and 24 hours.

6.3.3 Equilibrium prices

Associated with the aforementioned attention allocations are equilibrium prices that tell us how much we can expect our entire 144-patient ICU to benefit from small attention increments. In Figure 6.15, I plot these equilibrium prices $\lambda(C)$ for various values of the attention horizon T. In each of these plots, the magnitude of $\lambda(C)$ appears negligible for $C > 30$: The first 30 attention units are worth much more to our ICU than are any additional units. If additional attention units were made available, their specific distribution would be practically inconsequential.

6.4 Case 6: Simulating other disease categories

Thus far, we have been dealing with simulated patients who each suffer from some type of CHF. I next simulated other disease categories through an appropriate manipulation of VentPlan's physiologic parameters. In my final series of experiments, I tested SIMON on a simulated ICU that contained three patients, each of whom was recovering from one disorder: congestive heart failure, pulmonary embolus, or pneumonia.

6.4.1 Patient descriptions

Table 6.7 displays pertinent identifying information for our three patients. The pathophysiology of Mr. Dickinson (patient 1) has been discussed in Section 5.4.1. Brief descriptions of the physiologic challenges faced by patients 2 and 3 follow.

ID #	NAME	CONDITION	DESCRIPTION
1	Mr. Dickinson	congestive heart failure	Table 6.1
2	Ms. Di'Anno	pulmonary embolism	Table 6.8
3	Mr. Bayley	pneumonia	Table 6.9

Table 6.7. Description of three patients (case 6).

6.4.1.1 Pulmonary embolism

Ms. Di'Anno is a 76-year-old woman who has been bedridden over the past several years for a variety of ailments. Thirty-six hours ago, she was admitted to the emergency room for sudden-onset dyspnea and pleuritic chest pain. She was subsequently admitted to the ICU after being diagnosed as having a pulmonary embolism: A blood clot from her systemic venous circulation (most likely the deep leg veins) had lodged in a pulmonary vessel, occluding blood flow to a region of her lung. Such functional obliteration of pulmonary vasculature impairs gas exchange through ill-defined mechanisms that increase both the physiologic shunt and the physiologic dead space. Hypoxemia occurs. Pulmonary embolism can be rapidly fatal if not corrected promptly; patients who receive treatment need to be monitored for recurrence.

Parameters that encode Ms. Di'Anno's pulmonary embolism are shown in Table 6.7. The critical state is marked by an increase in the dead space V_{ds} and the physiologic shunt fraction f_S. Furthermore, the occlusion of overall blood flow decreases the cardiac output Q.

<table>
<tr><th rowspan="2">PHYSIOLOGIC STATE</th><th colspan="5">PHYSIOLOGIC PARAMETERS</th></tr>
<tr><th>Q</th><th>V_{ds}</th><th>$\dot{V}_{O_2}$</th><th>f_S</th><th>RQ</th></tr>
<tr><td>Unstable</td><td>3.0 l/min</td><td>150 ml</td><td>150 ml/min</td><td>0.110</td><td>0.9</td></tr>
<tr><td>Critical</td><td>2.5 l/min</td><td>200 ml</td><td>150 ml/min</td><td>0.260</td><td>0.9</td></tr>
<tr><td colspan="2">BASE DISCHARGE RATE
h_u</td><td colspan="2">BASE RECOVERY RATE
h_c</td><td colspan="2">ATTENTION CAPACITY
κ</td></tr>
<tr><td colspan="2">$(48\ \mathrm{hr})^{-1}$</td><td colspan="2">$(24\ \mathrm{hr})^{-1}$</td><td colspan="2">1</td></tr>
</table>

Table 6.8. Pulmonary-embolism patient (case 6).

From this patient specification, I compiled a patient model suitable for SIMON by modifying slightly the procedure outlined in Section 5.4.2. For a given physiology recognized by VentPlan, a pulmonary-embolism patient deterioriates to a more seriously ill state twice as quickly as does a CHF patient.[4] Therefore, I doubled the deterioration rates and death rates of Section 5.3.3.

6.4.1.2 Pneumonia

Mr. Bayley, a 46-year old man who has a history of smoking, was admitted to the ICU four days ago following a routine cholecystectomy. During the course of his stay, he developed a fever that persisted for three days, despite the use of intravenous antibiotics that covered gram-positive bacteria. These signs and the results of radiologic studies caused his physicians to suspect a nosocomial (hospital-acquired) pneumonia. Compared to their community-acquired counterparts, nosocomial pneumonias are more aggressive and more resistant to conventional antibiotic therapy; they are associated with a significantly higher mortality rate.

Table 6.9 displays a quantitative description of Mr. Bayley's medical condition. From a cardiopulmonary standpoint, pneumonia manifests itself as a hypermetabolic state, in which increased oxgyen consumption $\dot{V}_{\mathrm{O}_2}$ is supported by increased cardiac output Q. In addition, a concomitant disturbance in the ventilation–perfusion ratio increases the physiologic shunt fraction f_{S}. Severe pneumonia, such as those that often result from nosocomial infection, can cause hypoxia and death.

PHYSIOLOGIC STATE	PHYSIOLOGIC PARAMETERS				
	Q	V_{ds}	$\dot{V}_{\mathrm{O}_2}$	f_{S}	RQ
Unstable	3.0 l/min	150 ml	150 ml/min	0.110	0.9
Critical	4.5 l/min	150 ml	250 ml/min	0.210	0.9

BASE DISCHARGE RATE h_{u}	BASE RECOVERY RATE h_{c}	ATTENTION CAPACITY κ
$(48\ \mathrm{hr})^{-1}$	$(24\ \mathrm{hr})^{-1}$	1

Table 6.9. Pneumonia patient (case 6).

4. This assessement was made by Dr. Rutledge, the creator of VentPlan.

6.4.2 Attention allocations

It would be ideal to provide optimal treatment for all three of these patients on a continuing basis. But when attention resources are limited, the best we can do is to allocate them wisely. I computed, for various values of T, the recommended attention allocations for the three patients in the unstable, properly treated state, as a function of the total available attention C; the results are plotted in Figure 6.16. What is apparent immediately in all of these plots is the order in which the patients receive priority: As C is increased from 0, patient 2 is activated first, followed by patients 1 and 3. Furthermore, we witness $x_2^*(C) \geq x_1^*(C) \geq x_3^*(C)$ for all C.

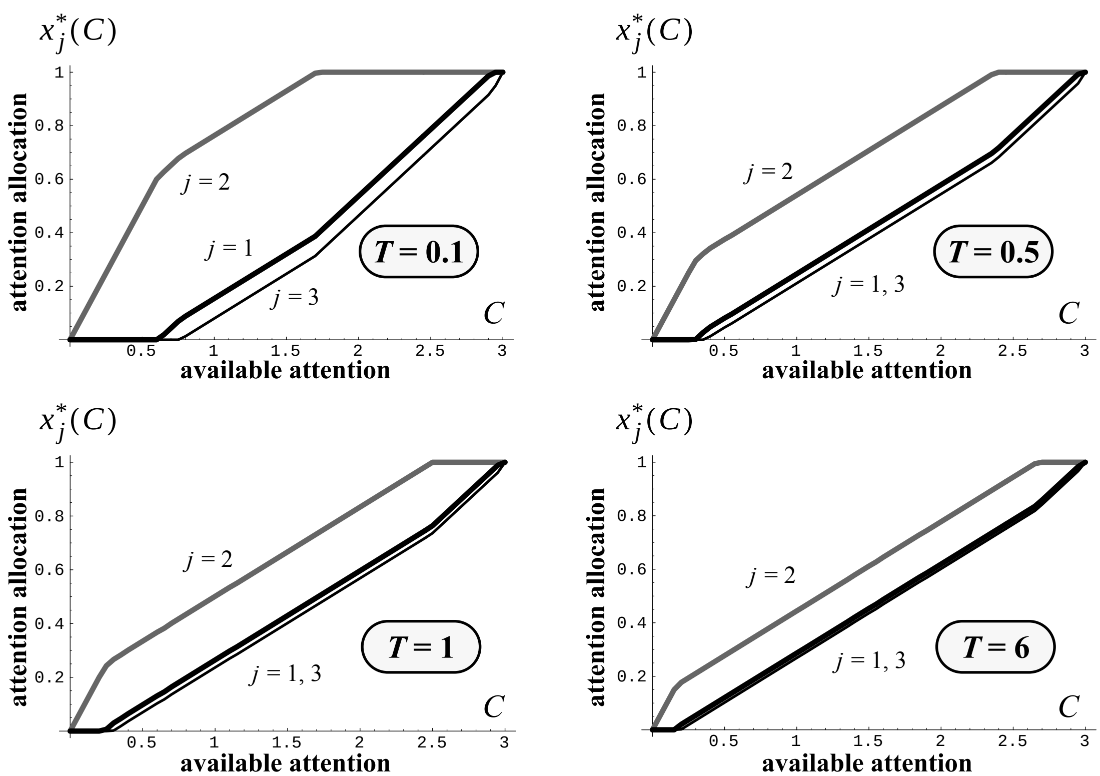


Figure 6.16. Attention allocations versus C for fixed T (case 6).
The attention allocations $x_j^*(C)$ for case 6 are plotted for $T = 0.1$, 0.5, 1, and 6 hours.

We conclude broadly from these results that Ms. Di'Anno (patient 2) commands measurably more attention than does either Mr. Dickinson (patient 1) or Mr. Bayley (patient 3).

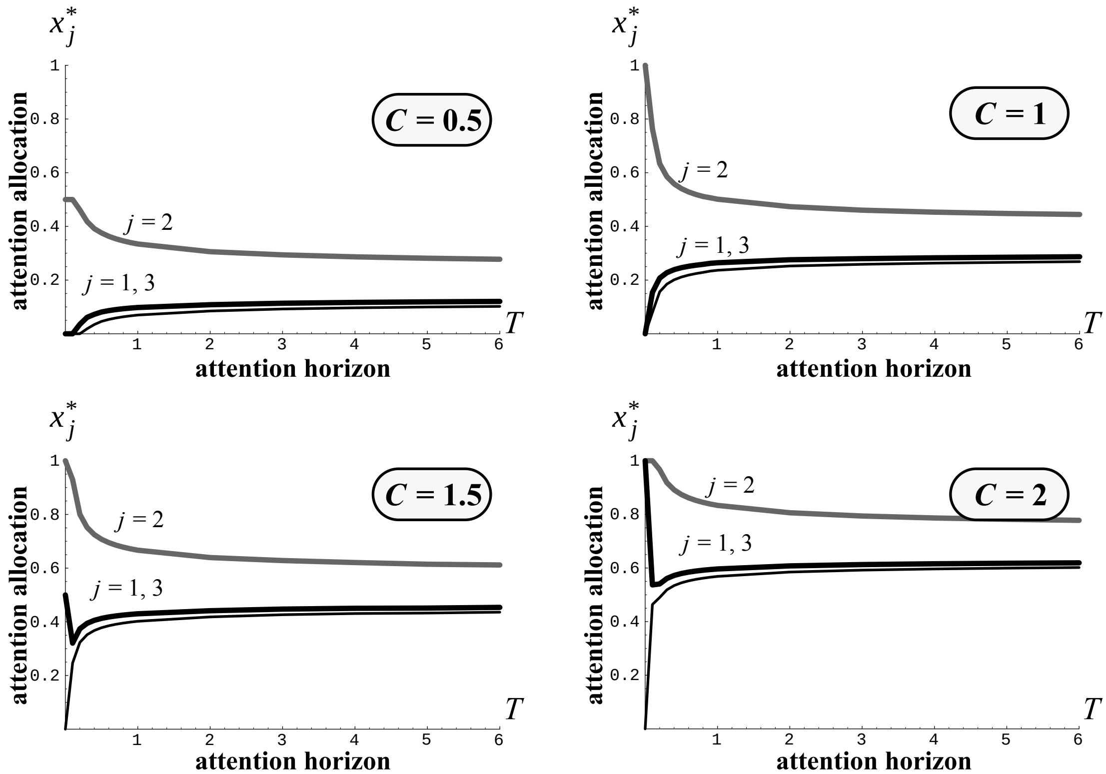


Figure 6.17. Attention allocations versus *T* for fixed *C* (case 6).

The attention allocations $x_j^*(C)$ for case 6, as a function of T, are plotted for $C = 0.5$, 1, 1.5, and 2 attention units.

The exact amounts are worth looking at. Suppose that $C = 1$—that one clinician is available to care for the three patients. If the clinician is considering the next 6 minutes of her time ($T = 0.1$ hours), she should devote about 75 percent of her attention to Ms. Di'Anno, 15 percent to Mr. Dickinson, and 10 percent to Mr. Bayley.[5] Over a longer time horizon, however, these attention allocations become less centered on Ms. Di'Anno. For example, at $T = 6$ hours, Ms. Di'Anno commands about 40 percent of the total attention, and the other two patients each command about 30 percent.

5. Note that when the attention horizon T is sufficiently small, the continuous-multitasking assumption (discussed in Chapter 3) becomes unrealistic. I use $T = 0.1$ hours to demonstrate an attention allocation focused mostly on Ms. Di'Anno.

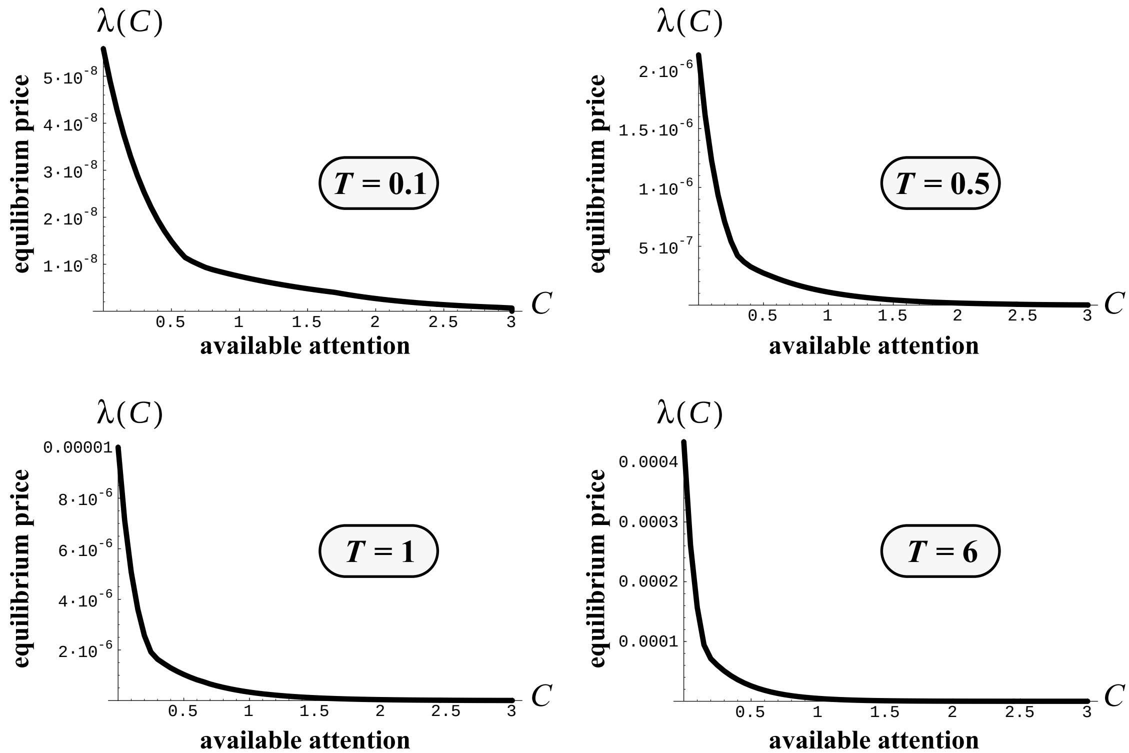


Figure 6.18. Equilibrium price versus *C* for fixed *T* (case 6).
The equilibrium prices $\lambda(C)$ for case 6 are plotted for $T = 0.1$, 0.5, 1, and 6 hours.

I plotted the attention allocations as a function of the horizon T, for various values of C, in Figure 6.17. Again, we see that patient 2 is allocated notably more attention than either patients 1 or 3. The available attention concentrates on the patients that have the highest priority as T approaches 0.

Remember that these attention allocations were computed under the assumption that the three patients are unstable and are being properly treated. Naturally, the attention allocations would change if any patient were to devolve to a critical state.

6.4.3 Equilibrium prices

The equilibrium prices $\lambda(C)$ are plotted for various values of T in Figure 6.18. Compared with the λ plots that we encountered previously, in these it is difficult to judge, without controversy, how much attention is sufficient. In particular, for $T = 0.1$, graphically discernible gains are predicted until C is close to its maximum value of 3.

The preceding observation reinforces an important point stressed by this dissertation, and by decision theory in general. The question is *not* how much attention is sufficient for any given patient. Rather, the question is how we should allocate limited attention resources to all the patients. It does no good to our patients to pronounce arbitrarily whether we have helped them enough, when we are given only a limited amount of attention to allocate. We can only do our best. If our best efforts are insufficient, there may be a justification for mobilizing attention resources from outside the ICU.

6.5 Summary

In this chapter, I reported on my evaluation of the alarm methodology that I developed in Chapters 3 and 4. To demonstrate validity, I investigated how SIMON's outputs responded to controlled changes in input. The results corresponded with clinical intuition and, in some cases, elucidated clinical insights that were not immediately apparent from the patient models. To demonstrate computational scalability, I applied SIMON to a simulated ICU that contained 144 patients. For these patients, SIMON produced a complete set of attention allocations in under 2.1 seconds on a machine that had only moderate processing capabilities. These results, and their timeliness, show that my methodology can scale up gracefully to problems that involve many concurrent processes. I thus conclude that my alarm methodology is both valid and computable.

Chapter 7

Conclusions

In this chapter, I summarize my work on developing a computable, normatively sound methodology for intelligent alarms. I reflect on the contributions made to various disciplines, and I delineate various possibilities for future research.

7.1 Summary of dissertation

In Chapter 1, I reflected on the monitoring tasks that alarms are intended to support, and I established that an intelligent alarm produces alerts that allocate the attention of busy agents among a set of concurrent processes they are managing. Unfortunately, current alarms fall short of this ideal; I discussed, in particular, how current intensive-care unit (ICU) alarms, which are typically based on thresholds for individual measured parameters, often encourage nonoptimal attention allocations.

In Chapter 2, I surveyed various methods that researchers have developed to improve the current state of alarms. Although these methods perform inferences of varying sophistication, most of them do not explicitly consider the problem of attention allocation. Indeed, we saw that there are difficult, fundamental challenges in even *formulating* an attention-allocation problem: Any formulation that manipulates quantitative attention amounts must incorporate simplifying assumptions about the future evolution of processes. Fortunately, these assumptions can be made transparent under the decision-theoretic basis.

I developed my alarm approach in Chapters 3 and 4. In Chapter 3, I used Markov decision processes (MDPs) to model the value of applying a partial amount of attention over a limited **attention horizon** T into the future. In Chapter 4, I presented a framework that used separable-optimization techniques to allocate an **available attention** amount C among a set of concurrent, continuing processes. Given C and T, my framework computes *two* alert measures for each process:

- an **activation price,** which reflects the *priority,* or the order by which the process merits attention
- an **attention allocation,** which indicates *how much* of the available attention the process should receive over T

In Chapter 5, I described SIMON, a program that applies my alarm approach to the problem of monitoring multiple patients in a busy ICU. I elaborated, in particular, a schema that I developed for creating realistic ICU-patient models. Then, in Chapter 6, I evaluated my alarm approach. To assess validity, I asked whether SIMON produces sensible output given sensible input. The results showed that SIMON produces alarm signals that can be explained from the normative basis, and that are consistent with sound clinical judgment. To assess computability, I used SIMON to generate alarm signals for an ICU that contained 144 simulated patients; the entire computation took about 2.1 seconds on a machine with only moderate processing capabilities. I thus concluded that my alarm framework is valid and computable, and therefore is potentially useful in a real-world ICU setting.

We noted, in passing, that processes with higher priority measures do not necessarily have higher attention allocations. This observation confirmed the complementary roles of the two alert measures.

7.2 Contributions

My dissertation research makes contributions in the fields of critical-care medicine, decision analysis (DA), artificial intelligence (AI), and medical informatics.

7.2.1 Critical-care medicine

My work contributes to the field of critical-care medicine by addressing the inadequacies of current alarms in the critical-care setting.

7.2.1.1 Cost-effective ICU management

Critical-care medicine is a multidisciplinary specialty that encompasses the care of patients who are being treated in an ICU [Seiver, 1992]. The cost of ICU care is significant, accounting for 1 percent of the gross national product [Raffin et al., 1989]. Virtually every medical subspecialty has been affected by recent pressures to produce cost-effective health care. Critical-care medicine, in particular, has undergone intense scrutiny: Disproportionate portions of health-care resources are expended on ICU patients, many of whom do not survive the hospitalization [Esserman et al., 1995; Oye & Bellamy, 1989].

Alarms can improve the cost-effectiveness of critical care by issuing alerts according to the attention needs of individual patients. They can help ICU clinicians to maintain multiple patients concurrently, thereby increasing the effective utilization of scarce ICU-clinician resources. Also, an attention-based alarm can reduce the costs (monetary and patient risk) of unnecessary monitoring by utilizing only those patient measurements that affect the decision to confer attention. These benefits, however, can be realized fully only if the alarms function as intended. Current ICU alarms do not optimally allocate the attention of clinicians among the patients they are managing.

7.2.1.2 The promise of intelligent alarms

To address the inadequacies of current ICU alarms, I considered the domain-independent alarm task. Current alarms are inadequate because their underlying methodologies, however sophisticated, do not address explicitly the problem of attention allocation. I developed a normative framework for producing alarm signals that optimally allocate the attention of busy agents among a set of concurrent processes.

To assess the potential applicability of my alarm framework in the critical-care setting, I developed a prototype, SIMON, that allocates scarce clinician resources for a simulated ICU. SIMON's patient models are based, in large part, on the outputs of a validated prototype ventilator-management advisor [Rutledge et al., 1993]. SIMON's outputs were consistent with clinical intuition and, in some cases, elucidated clinical insights that were not immediately apparent from the patient models. SIMON was able to generate alarm signals in real time, even for a large, simulated ICU on a platform with only moderate processing capabilities. Thus, my alarm approach proved to be valid and computable in a realistic, simulated ICU setting. Alarm signals that allocate the attention of ICU clinicians (or of busy agents in general) did not exist prior to my dissertation research.

7.2.2 Decision analysis

My work on intelligent alarms contributes to DA by defining and implementing a normative concept of attention allocation among concurrent, continuing processes. Traditionally, DA has focused on building and solving normative models of single-shot decisions: their alternatives, their preferences, and their uncertainties. This simple model was the basis of all subsequent research that sought to expand the scope and applicability of DA.

7.2.2.1 From normative foundations to practical applications

Recently, DA researchers have been interested in modeling decisions that occur over time. Such interest has spurred the development of sequential decision models [Gorry & Barnett, 1968] and multistage decision models, in particular the MDP [Howard, 1960; Howard, 1971; Ross, 1983; Puterman, 1994]. Although the notion of time opened up broad, new avenues of research—such as the problem of decision-theoretic control—the problem of allocating attention among multiple, concurrent processes under constraints of time and attention resource remained largely unchartered.

Horvitz [1995] conducted pioneering, decision-theoretic research on the attention-allocation problem by developing the notion of ECDA for a particular class of time-critical processes. He used the ECDA measure to assign priorities to multiple processes that are competing for the immediate attention of a limited supply of agents [Horvitz & Barry, 1995; Horvitz & Seiver, 1997]. However, ECDA was built on numerous assumptions that limited its application scope (Sections 2.5.3 and 4.7.1).

7.2.2.2 New possibilities in attention allocation

From a DA standpoint, my dissertation research extends the boundaries of decision-theoretic attention allocation in the following manners:

- *Problem definition.* I showed that the definition of an attention-allocation problem posed difficult challenges. Specifically, I had to make assumptions about the processes being modeled, to formulate an attention-allocation problem that is tractable and is meaningful to the end user. I used my meta-analysis of the attention-allocation problem to guide my choice of assumptions.
- *Partial attention.* Binary notions of attention were modeled in the research that led to the ECDA measure. My alarm approach generalized this all-or-none notion of attention by defining and modeling *fractions of attention* applied over a given attention horizon (Sections 2.5 and 4.7).
- *Expanded process scope.* The ECDA measure was developed for a class of processes that embody two assumptions: (1) the effects of immediate action dominate over the effects of subsequent observations and actions; (2) the ECDA function (utility versus time) is assessed by a domain expert, for each possible process state. My alarm framework explicitly considers the effects of interleaved observations and actions over the attention horizon. Furthermore, my utilities of partial attention are computed from MDP parameters, rather than assessed arbitrarily by an expert. Thus, my work extends the range of processes that can be managed with the help of an attention allocation, or alarm system.

- *New alarm concepts.* ECDA is a one-dimensional measure of urgency that does not scale up to the general attention-allocation problem. My alarm framework generates *two* alarm signals for each process: its activation price and its attention allocation. These measures constitute a nondeterministic guide for how busy agents should allocate their future attention resources.

7.2.3 Artificial intelligence

AI researchers develop methodologies that simulate intelligent behavior. My work contributes to AI by providing a normative, computable framework for intelligent alarms that can help a group of busy agents to manage a broad class of time-varying, stochastic processes that respond to externally applied actions.

7.2.3.1 A mission in evolution

Classic AI emphasized symbolic manipulation as a means of achieving intelligent inferences [Simon, 1969]. Early AI researchers developed expert systems that were based on the chaining of rules [Buchanan & Shortliffe, 1984]; rule-based expert systems still enjoy a broad application scope today. During the mid-1970s, however, AI researchers, faced with uncertainties that are inherent in complex domains, acknowledged the need to incorporate uncertainty into their symbolic-reasoning frameworks [Shortliffe & Buchanan, 1975]. In general, a systematic approach to uncertain reasoning required a quantitative (or pseudoquantitative) measure of an intelligent agent's degree of belief. Shafer and Pearl [1990] reviewed various approaches to modeling uncertainty in artificial intelligence.

During the early 1980s, AI researchers also began to incorporate the notion of time into their frameworks, to achieve intelligent inferences in complex, dynamic domains [Fagan, 1980; Shahar, 1994; Haimowitz, 1994]. Uckun [1994] reviewed various AI approaches to the difficult problem of monitoring ICU patients. Researchers developed ICU-monitoring prototypes that embodied different reasoning approaches, to take advantage of the unique advantages of each approach [Hayes-Roth et al., 1992; Rutledge et al., 1993].

Recently, AI researchers have been concerned wtih how to monitor dynamic environments in ways that respect the scarcity of real-world resources. Because many problems in AI are $\mathcal{NP}$-hard, several researchers looked specifically at the problem of allocating scarce *computational* resources in time-critical monitoring applications [Horvitz, 1990; Ash et al., 1993; Rutledge, 1995]. Horvitz [1995], in particular, built on his earlier work by investigating methods for allocating scarce *attention* resources for a particular class of time-critical processes.

7.2.3.2 My contributions

From an AI perspective, my aim was to develop and to build a prototype for an intelligent solution to the alarm problem. I contributed to AI in the following manners:

- *Attention-allocation framework and prototype.* I argued that an intelligent alarm allocates the attention of busy agents among a set of concurrent processes they are managing. I developed a normatively sound approach for performing this attention allocation. I built a prototype, SIMON, that generates decision-theoretic alarm signals for a simulated ICU, and used it to verify the validity and computability of my alarm approach.
- *New applications for decision theory.* Since the mid-1980s, several researchers paved the way for decision theory's eventual acceptance as a sound and computable approach to automated reasoning (Section 2.3.3). In developing and evaluating a decision-theoretic framework for alarms, I further strengthened the argument for adopting decision-theoretic approaches to intelligent reasoning.
- *Computational developments.* Because AI problems are typically $\mathcal{NP}$-hard, their solutions often embody computational techniques that are designed to overcome this intractability. My alarm approach performs inference on a state space whose size increases exponentially with the number of processes N. By designing a model of partial attention that reasons about attention amounts, and by employing techniques from the theory of separable optimization, I enabled my alarm approach to perform this inference in time proportional to N.

7.2.4 Medical informatics

My work contributes to medical informatics by addressing a real-world, medical information-management problem, and by developing and prototyping a novel, interdisciplinary approach to solving that problem.

7.2.4.1 An interdisciplinary field

Medical informatics is the study of methodologies for representing, transmitting, processing, and displaying medical information [Greenes & Shortliffe, 1990; Shortliffe & Perreault, 1990]. It is inherently interdisciplinary, drawing from diverse fields such as biomedicine, computer science, decision science, information science, cognitive science, information science, management science, and several other component disciplines. Medical informatics evolved in response to the increasing need for intelligent, automated management of voluminous, complex medical information.

7.2.4.2 My interdisciplinary contributions

My research contributes to medical informatics in several dimensions:

- *Critical-care medicine.* My work contributes to research on intelligent alarms for the ICU setting. In analyzing the inadequacies of current ICU alarms, I identified the root problem as an informatics problem. I developed an original, domain-independent solution to the alarm problem, and then validated its output in a simulated ICU environment.
- *DA and AI.* In addressing the informatics problem of producing intelligent alerts in the critical-care setting, I developed methodologies that, in and of themselves, can be considered contributions to the fields of DA and AI. I elaborated on these domain-independent contributions in Sections 7.2.2 and 7.2.3.

- *Interdisciplinary developments.* Medical-informatics research has a unique history of combining techniques from different disciplines, in the greater effort to attack real-world, medical-information management challenges. My alarm framework embodies techniques from DA, stochastic dynamic process theory, and optimization theory.

7.3 Future research

I propose several avenues of research to develop improved methods for allocating scarce attention resources among a set of concurrent processes.

7.3.1 Problem definition

I have established that an intelligent alarm assists busy agents in the allocation of scarce attention resources, and that framing an attention-allocation problem requires making assumptions about how such attention would be utilized in the future. My framework for intelligent alarms assumes that a decision maker can allocate partial amounts of attention to each of a set of concurrent process. Furthermore, it assumes that her attention is distributed uniformly over a finite duration, which means that she multitasks continuously and randomly among the processes (Section 3.5).

Although my framework for attention allocation advances the state of the art (Sections 2.5 and 4.7), its built-in assumptions limit its application scope. The notion that decision makers multitask continuously and randomly among processes is a mathematical idealization that approximates certain real-world situations more closely than it does others. For example, the continuous-multitasking model may describe reasonably the activities of clinicians in an ICU where interventions are performed quickly and patients are located in close proximity to one another; however, lengthy interventions or long travel times between patients would make the approximation less accurate.[1] One possibility for future work is to characterize, more precisely, the class of processes that can be described accurately by my alarm framework.

Another opportunity for future investigation is to develop alternative frameworks for allocating attention. There may be other ways to approximate the exact, intractable scenario of Section 2.5, such that the computational outputs can be construed meaningfully as alarm signals. These alarm signals would assume forms, and have meanings, different from those of the numerical measures of my framework. Once developed, such approximations could describe the allocation of attention in certain domains that are not as well-described by my framework.

7.3.2 Model of partial attention

In Chapter 3, I described a model of partial attention that is based on the continuous-time MDP. The MDP is the most general model of stochastic processes for which practical solutions have been developed [Howard, 1971; Puterman, 1994]. I developed my model in continuous time because the required mathematics is clean, and the semantics of applying partial attention over a continuous-time interval is straightforward.

Nonetheless, we can extend the application scope of my intelligent-alarm framework by developing models of partial attention for other classes of stochastic processes. The most straightforward possibility is to develop a discrete-time–MDP model of partial attention. A discrete-time model may be useful in applications for which there are a limited number of decisions to be made over the attention horizon. We can develop a discrete-time version of the model in Chapter 3 by using z-transforms to manipulate transition probabilities, in a manner analogous to the use of Laplace transforms to manipulate transition rates [Howard, 1960; Howard, 1971]. Although using z-transforms presents no new analytical barriers, the discrete-time model introduces semantic challenges that did not exist in the continuous-time realm. Namely, if we have an attention horizon of N discrete periods, then the notion of continuously divided amounts of attention becomes suspect: We cannot talk about continuous, random multitasking, when there are only N opportunities to switch

1. *Lengthy* and *long* refer to amounts of time that are significant, in comparison with the total duration of interest.

tasks during the attention horizon. The form, meaning, and applicability of alarm metrics in the discrete-time realm need to be elucidated through further research.

Another opportunity for future work is to develop a model of partial attention that is based on the partially observable Markov decision process (POMDP). The POMDP generalizes the MDP: In a POMDP, states of the world at time t influence states of the world at time $t+1$, but the states can be observed only indirectly, through proxy quantities that provide probabilistic information about the states. Algorithms for solving POMDPs were first developed by Sondik [1971; 1978]; the POMDP is now finding increasing use as a modeling formalism in current artificial-intelligence research [Cassandra et al., 1997; Monahan, 1982]. The POMDP presents significant computational challenges because of the need to maintain a *probability distribution* over the set of possible current states [Sondik, 1971]; in constrast, the MDP requires merely that we observe and remember the current state. Thus, to allocate scarce attention resources among multiple, concurrent POMDPs in a tractable manner will be even harder. However, if we meet this challenge, we can greatly broaden the application scope of decision-theoretic models for attention allocation.

We can also explore the possibility of assessing the utility function directly. Horvitz and associates characterized a class of processes for which the utility as a function of the attention horizon can be assessed directly [Horvitz & Seiver, 1997; Horvitz & Rutledge, 1991]. Future work that builds on mine may similarly involve identifying domains for which we can directly assess the utility as a function of attention horizon and attention amount.

7.3.3 Optimization

In Chapter 4, I modeled the allocation of scarce attention resources as a convex separable optimization problem. My adoption of optimization techniques was facilitated by my modeling of attention as a scalar quantity (Sections 2.5.4 and 3.5). The independent operation of multiple processes enabled me to introduce the separability assumption gracefully. I imposed the convexity assumption, thereby guaranteeing that the locally optimal attention allocation is globally optimal (Sections 4.4 and 4.5). Although the utility of partial attention is an algorithmically generated function, I used closed-form exponential expressions to model the utility function. This approximation enabled me to convert the convex separable problem into a one-dimensional line search, the computational complexity of which increases linearly with the number of processes (Section 4.6).

The convexity assumption merits further analysis. For the MDP models used in this dissertation, the expected utility as a function of the amount of attention allocated increased with diminishing returns. However, it is not clear which MDP models exhibit this convexity property. We must do additional work to delineate the class of MDP models for which the convexity assumption holds.

The accuracy of the optimization can be improved through additional engineering. We can improve on the curve-fitting technique of Section 4.6.2 by using additional points, or by fitting a more complex parameterized expression. Alternatively, we can eschew the use of exact expressions and explore instead the possibility of developing and using numerical approximations of the utility of partial attention.

The attention allocations as a function of the attention horizon T also can be better understood. The meaning of T, and the examples of Section 4.6 and Chapter 6, suggest that the available attention concentrates on the highest-priority processes as T approaches 0. Future work may uncover a theoretical basis for this behavior.

7.4 Concluding remarks

Many challenges remain in our quest to realize an intelligent alarm. There remain unresolved issues in the framing, solving, and prototyping of an attention-allocation approach. We should, however, be encouraged by the results of my work, which suggest that a computable, normatively sound approach to the alarm problem can be achieved. It is my hope that my contributions will inspire future researchers who are interested in developing decision-theoretic solutions to the alarm problem. By exercising care in framing the alarm problem, and by adopting fundamentally sound approaches in our solutions, we can best position ourselves to build alarms that serve their intended purposes.

Bibliography

Ash D, Gold G, Seiver A, Hayes-Roth B (1993). Guaranteeing real-time response with limited resources. *Artificial Intelligence in Medicine,* 5:49–66.

Avent RK, Charlton JD (1990). A critical review of trend-detection methodologies for biomedical monitoring systems. *CRC Critical Reviews in Biomedical Engineering,* 17:621–659.

Bayes T (1958). An essay towards solving a problem in the doctrine of chances. *Biometrika,* 46:293–298. Reprint of original work of 1763.

Becker K, Thull B, Kasmacher-Leidinger H, Stemmer J, Rau G, Kalff G, Zimmermann HJ (1997). Design and validation of an intelligent patient monitoring and alarm system based on a fuzzy logic process model. *Artificial Intelligence in Medicine,* 11:33–53.

Beinlich IA, Gaba DM (1989). The ALARM monitoring system: Intelligent decision making under uncertainty. *Anesthesiology,* 71:A336.

Beneken JEW, van der Aa JJ (1989). Alarms and their limits in monitoring. *Journal of Clinical Monitoring,* 5:205–210.

Buchanan BG, Shortliffe EH (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA.

Cassandra A, Littman ML, Zhang NL (1997). Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence,* pp 54–61, Morgan Kaufmann, San Francisco.

Connelly DP, Willard KE, Hallgren JH, Sielaff BH (1996). Closing the clinical laboratory testing loop with information technology. *American Journal of Clinical Pathology,* 105:S40–47.

Dagum P, Galper A, Horvitz E, Seiver A (1995). Uncertain reasoning and forecasting. *International Journal of Forecasting,* 11:73–87.

Dagum P, Luby M (1997). An optimal approximation algorithm for Bayesian inference. *Artificial Intelligence,* 93:1–27.

de Dombal FT, Leaper DJ, Staniland JR, McCann AP, Horrocks JC (1972). Computer-aided diagnosis of acute abdominal pain. *British Medical Journal,* 2:9–13.

Dean T, Kaelbling L, Kirman J, Nicholson A (1995). Planning under time constraints in stochastic domains. *Artificial Intelligence,* 76:35–74.

Deller A, Konrad R, Kilian J, Schühle B (1992). Alarms in an operative intensive care unit: Response of the nursing staff. In [Hedley-Whyte, 1992], pp 19–26.

Esserman L, Belkora J, Lenert L (1995). Potentially ineffective care: A new outcome to assess the limits of critical care. *Journal of the American Medical Association,* 274:1544–1551.

Fagan LM (1980). *VM: Representing Time-dependent Relations in a Medical Setting.* Doctoral dissertation. Department of Computer Science, Stanford University, Stanford, CA, June 1980.

Farr BR (1991). *Assessment of Preferences Through Simulated Decision Scenarios.* Doctoral dissertation. Section on Medical Informatics, Stanford University, Stanford CA, June 1991.

Gleick J (1987). *Chaos: Making a New Science.* Penguin, New York.

Gorry G, Barnett G (1968). Experience with a model of sequential diagnosis. *Computers and Biomedical Research,* 1:490–507.

Gravenstein JS, Newbower RS, Ream AK, Smith NT (1987), editors. *The Automated Anesthesia Record and Alarm Systems.* Butterworths, Boston.

Gravenstein JS, Newbower RS, Ream AK, Smith NT (1983), editors. *An Integrated Approach to Patient Monitoring.* Butterworths, Boston.

Greenes RA, Shortliffe EH (1990). Medical informatics: An emerging discipline with academic and institutional perspectives. *Journal of the American Medical Association,* 263:1114–1120.

Haimowitz IJ (1994). *Knowledge-Based Trend Detection and Diagnosis.* Doctoral dissertation. Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, June 1994.

Haug PJ, Gardner RM, Tate KE, Evans RS, East TD, Kuperman G, Pryor TA, Huff SM, Warner HR (1994). Decision support in medicine: Examples from the HELP system. *Computers and Biomedical Research,* 27:396–418.

Hayes-Roth B, Washington R, Ash D, Hewett R, Collinot A, Vina A, Seiver A (1992). Guardian: A prototype intelligent agent for intensive-care monitoring. *Artificial Intelligence in Medicine,* 4:165–185.

Heckerman DE (1990). Appendix A: Background. In *Probability Similarity Networks.* Doctoral dissertation. Section on Medical Informatics, Stanford University, Stanford, CA, June 1990.

Heckerman DE (1986). Probabilistic interpretations for MYCIN's certainty factors. In Kanal L and Lemmer J, editors, *Uncertainty in Artificial Intelligence,* pp 167–196. North-Holland, New York.

Heckerman DE, Shortliffe EH (1992). From certainty factors to belief networks. *Artificial Intelligence in Medicine,* 4:35–52.

Hedley-Whyte J (1992), editor. *Operating Room and Intensive Care Alarms and Information Transfer, ASTM STP 1152.* American Society for Testing and Materials, Philadelphia, PA.

Horvitz E (1995). Transmission and display of information for time-critical decisions. Technical report MSR-TR-95-13, Decision Theory Group, Microsoft Research, Redmond, WA.

Horvitz EJ (1990). *Computation and Action Under Bounded Resources.* Doctoral dissertation. Section on Medical Informatics, Stanford University, Stanford, CA, December 1990.

Horvitz E, Barry M (1995). Display of information for time-critical decision making. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence,* pp 296–305, Morgan Kaufmann, San Francisco.

Horvitz EJ, Breese JS, Henrion M (1988). Decision theory in expert systems and artificial intelligence. *Journal of Approximate Reasoning, Special Issue on Uncertain Reasoning,* 2:247–302.

Horvitz EJ, Heckerman DE (1986). The inconsistent use of measures of certainty in artificial intelligence research. In Kanal LN and Lemmer JF, editors, *Uncertainty in Artificial Intelligence,* pp 137–151. North-Holland, Amsterdam.

Horvitz E, Rutledge G (1991). Time-dependent utility and action under uncertainty. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence,* pp 151–158, Morgan Kaufmann, San Mateo, CA.

Horvitz E, Seiver A (1997). Time-critical action: Representations and application. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence,* pp 250–257, Morgan Kaufmann, San Francisco.

Howard RA (1988). Decision analysis: Practice and promise. *Management Science,* 34:679–695.

Howard RA (1971). *Dynamic Probabilistic Systems,* Vol. 1 & 2. Wiley, New York.

Howard RA (1960). *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, MA.

Howard R, Matheson J (1981). Influence diagrams. In Howard R and Matheson J, editors, *Readings on the Principles and Applications of Decision Analysis,* Volume 2, pp 721–762. Strategic Decisions Group, Menlo Park, CA.

Huang C, Darwiche A (1996). Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning,* 15:225–263.

Jensen FV (1996). *An Introduction to Bayesian Networks.* Springer, New York.

Kestin IG, Miller BR, Lockhart CH (1988). Auditory alarms during anesthesia monitoring. *Anesthesiology,* 69:106–109.

Knaus WA, Draper EA, Wagner DP, Zimmerman JE, Bergner M, Bastos PG, Sirio CA, Murphy DJ, Lotring T, Damiano A, Harrel FE (1991). The APACHE III prognostic system: Risk prediction of hospital mortality for critically ill hospitalized adults. *Chest,* 100:1619–1636.

Laursen P (1994). Event detection on patient monitoring data using causal probabilistic networks. *Methods of Information in Medicine,* 33:111–115.

Littman ML, Dean TL, Kaelbling LP (1995). On the complexity of solving Markov decision problems. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence,* pp 394–402, Morgan Kaufmann, San Francisco.

Long W (1996). Temporal reasoning for diagnosis in a causal probabilistic knowledge base. *Artificial Intelligence in Medicine,* 8:193–215.

Luenberger DG (1984). *Linear and Nonlinear Programming.* Addison-Wesley, Reading, MA.

McDonald CJ (1976). Protocol-based computer reminders, the quality of care and the non-perfectability of man. *New England Journal of Medicine,* 295:1351–1355.

Meijler AP (1987). *Automation in Anesthesia: A Relief?* Springer-Verlag, Berlin.

Monahan GE (1982). A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science,* 28:1–16.

Moret-Bonillo V, Alonso-Betanzos A, Truemper EJ, Searle JR (1992). Intelligent monitoring and symbolic representation of clinical knowledge: An application in acute ventilatory management. In [Hedley-Whyte, 1992], pp 39–45.

Orr JA, Westenskow DR (1994). A breathing circuit alarm system based on neural networks. *Journal of Clinical Monitoring,* 10:101–109.

Oye RK and Bellamy RK (1989). Patterns of resource consumption in medical intensive care. *Chest,* 99(3): 685–689.

Páte-Cornell ME (1986). Warning systems in risk management. *Risk analysis,* 6:223–234.

Pearl (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

Philip JH (1989). Overview: Creating practical alarms for the future. *Journal of Clinical Monitoring,* 5:194–195.

Puterman ML (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York.

Quinn ML (1989). Semipractical alarms: A parable. *Journal of Clinical Monitoring,* 5:196–200.

Raffin TA, Shurkin JN, Sinkler W (1989). *Intensive Care: Facing the Critical Choices.* Freeman, New York.

Rich E (1983). *Artificial Intelligence,* pp 184–199. McGraw-Hill, New York.

Ross SM (1983). *Introduction to Stochastic Dynamic Programming.* Academic Press, New York.

Rutledge GW (1995). *Dynamic Selection of Models.* Doctoral dissertation. Section on Medical Informatics, Stanford University, Stanford, CA, March 1995.

Rutledge GW, Thomsen GE, Farr BR, Tovar MA, Polaschek JX, Beinlich IA, Sheiner LB, Fagan LM (1993). The design and implementation of a ventilator-management advisor. *Artificial Intelligence in Medicine,* 5:67–82.

Savage L (1954). *The Foundations of Statistics.* Dover, New York.

Seiver A (1992). *Decision Analysis: A Framework for Critical Care Decision Making.* Doctoral dissertation. Department of Engineering-Economic Systems, Stanford University, Stanford, CA, June 1992.

Shachter R (1988). Probabilistic inference and influence diagrams. *Operations Research,* 36:589–604.

Shafer G, Pearl J (1990), editors. *Readings in Uncertain Reasoning.* Morgan Kaufmann, San Mateo, CA.

Shahar Y (1994). *A Knowledge-Based Method for Temporal Abstraction of Clinical Data.* Doctoral dissertation. Section on Medical Informatics, Stanford University, Stanford, CA, October 1994.

Shortliffe EH (1987). Computer programs to support clinical decision making. *Journal of the American Medical Association,* 258:61–66.

Shortliffe EH, Buchanan BG (1975). A model of inexact reasoning in medicine. *Mathematical Biosciences,* 23:351–379.

Shortliffe EH, Perreault LE (1990), editors. *Medical Informatics: Computer Applications in Health Care.* Addison-Wesley, Reading MA.

Shwe M, Middleton B, Heckerman D, Henrion M, Horvitz E, Lehmann H, Cooper G (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base I: The probabilistic model and inference algorithms. *Methods of Information in Medicine,* 30:241–255.

Simon H (1969). *The Sciences of the Artificial.* MIT Press, Cambridge, MA.

Sondik EJ (1978). The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research,* 26:282–304.

Sondik E (1971). *The Optimal Control of Partially Observable Markov Processes.* Doctoral Dissertation. Department of Electrical Engineering, Stanford University, Stanford, CA, May 1971.

Speedie S, Skarupa S, Blaschke TF, Kondo J, Leatherman E, Perreault L (1987). MENTOR: Integration of an expert system with a hospital information system. In *Proceedings of the 11th Annual Symposium on Computer Applications in Medical Care,* pp 220–223.

Sykes MA (1989). Introduction, Panel on Practical Alarms: Fifth Internal Symposium on Computing in Anesthesia and Intensive Care. *Journal of Clinical Monitoring,* 5:192–193.

Tudor RS, Hovorka R, Cavan DA, Meeking D, Hejlesen OK, Andreassen S (1998). DIAS-NIDDM—a model-based decision support system for insulin dose adjustment in insulin-treated subjects with NIDDM. *Computer Methods and Programs in Biomedicine,* 56:175–191.

Tversky A, Kahneman D (1981). The framing of decisions and the psychology of choice. *Science,* 211:453–458.

Tversky A, Kahneman D (1974). Judgment under uncertainty: Heuristics and biases. *Science,* 185:1124–1131.

Uckun S (1994). Intelligent systems in patient monitoring and therapy management. *International Journal of Clinical Monitoring and Computing,* 11:241–253.

van der Aa JJ (1990). *Intelligent Alarms in Anesthesia: A Real Time Expert System Application.* Doctoral dissertation. Department of Anesthesiology, University of Florida, Gainesville, FL, May 1990.

van Oostrom JH, Gravenstein C, Gravenstein JS (1993). Acceptable ranges for vital signs during general anesthesia. *Journal of Clinical Monitoring,* 9:321–325.

von Neumann J, Morgenstern O (1947). *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, NJ.

Weigend A, Gershenfeld N (1993), editors. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, Reading, MA.

West M, Harrison J (1989). *Bayesian Forecasting and Dynamic Models.* Springer-Verlag, New York.

Zadeh L (1983). The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems,* 11:199–227.