Chapter 6

MULTI-ARMED BANDIT PROBLEMS

Aditya Mahajan

University of Michigan, Ann Arbor, MI, USA

Demosthenis Teneketzis

University of Michigan, Ann Arbor, MI, USA

1. Introduction

Multi-armed bandit (MAB) problems are a class of sequential resource allocation problems concerned with allocating one or more resources among several alternative (competing) projects. Such problems are paradigms of a fundamental conflict between making decisions (allocating resources) that yield high current rewards, versus making decisions that sacrifice current gains with the prospect of better future rewards. The MAB formulation models resource allocation problems arising in several technological and scientific disciplines such as sensor management, manufacturing systems, economics, queueing and communication networks, clinical trials, control theory, search theory, etc. (see [88] and references therein).

In the classical MAB problem (discussed in Section 2) at each instant of time a single resource is allocated to one of many competing projects. The project to which the resource is allocated can change its state; the remaining projects remain frozen (do not change state). In the variants of the MAB problem (discussed in Section 3) one or more resources are dynamically allocated among several projects; new projects may arrive; all projects may change state; delays may be incurred by the reallocation of resources, etc.

In general, sequential resource allocation problems can be solved by dynamic programming, as discussed in Chapter 2. Dynamic programming, which is based on backwards induction, provides a powerful method for the solution of dynamic optimization problems, but suffers from the "curse of dimensionality." The special structure of the classical MAB problem has led to the discovery of optimal "index-type" allocation policies that can be computed by forward induction (see Section 2.2), which is computationally less intensive than backward induction. Researchers have also discovered conditions under which forward induction leads to the discovery of optimal allocation policies for variants of the classical MAB (see Section 3). Discovering conditions under which "index-type" allocation policies are optimal or nearly optimal for sequential resource allocation problems remains an active area of research.

In this chapter we present a qualitative description of the classical MAB problem and its variants. All problems are formulated in discrete time. MAB problems have also been studied in continuous time (see for example [167, 80]). The emphasis in this chapter is on describing the key features of MAB problems and explaining the nature of their solutions. A rigorous proof of the results can be found in the references. The chapter is organized as follows. In Section 2 we present the formulation and the nature of the solution of the classical MAB problem. In Section 3 we describe the formulation and the nature of the solution of several variants of the classical MAB problem. We briefly discuss the relations of the MAB problem and its variants to sensor management in Section 4. A presentation of σ-fields and stopping times, concepts that are important in understanding the nature of the solution of the classical MAB problem, appears in Section 3 of the Appendix.

Remark on notation. Throughout this Chapter, we use uppercase letters to represent random variables and the corresponding lowercase letters to represent their realizations.

2. The Classical Multi-armed Bandit

In this section we present a general formulation of the MAB problem, highlight its salient features and describe its optimal solution. We also discuss forward induction, which provides insight into the nature of an optimal solution.

2.1 Problem Formulation

2.1.1 A Bandit Process. A (single-armed) bandit process is described by a machine/arm/project and is characterized by the pair of random sequences $\Big(\{X(0), X(1), \dots, \}, \{R\big(X(0)\big), R\big(X(1)\big), \dots \}\Big)$, where $X(n)$ denotes the state of the machine[1] after it has been operated n times, and $R\big(X(n)\big)$ denotes the reward obtained when the machine is operated for the n^{th} time. The state $X(n)$ is a real-valued random variable and $R\big(X(n)\big)$ is a random variable taking values in $\mathbb{R}_+$. In general when the machine is operated for the n^{th} time its state changes according to

$$X(n) = f_{n-1}\big(X(0), \dots, X(n-1), W(n-1)\big), \tag{6.1}$$

where $f_{n-1}(\cdot)$ is given and $\{ W(n); n = 0, 1, \dots \}$ is a sequence of independent real-valued random variables that are also independent of $X(0)$ and have known statistical description. Thus a (single-armed) bandit process is not necessarily described by a Markov process.

2.1.2 The Classical Multi-armed Bandit Problem. A multi-armed (k-armed) bandit process is a collection of k independent single-armed bandit processes. The classical MAB problem consists a multi-armed bandit process and one controller (also called a processor). At each time, the controller can choose to operate *exactly* one machine; all other machines remain frozen. Each machine i, $i = 1, 2, \dots, k$, is described by sequences $\{\big(X_i(N_i(t)), R_i\big(X_i(N_i(t))\big)\big); N_i(t) = 0, 1, 2, \dots, t; t = 0, 1, 2, \dots \}$, where $N_i(t)$ denotes the number of times machine i has been operated until time t. $N_i(t)$ is machine i's local time. Let $U(t) := \big(U_1(t), \dots, U_k(t)\big)$ denote the control action[2] taken by the controller at time t. Since the controller can operate on exactly one machine at each time, the control action $U(t)$ takes values in $\{ e_1, \dots, e_k \}$, where $e_j = (0, \dots, 0, 1, 0, \dots, 0)$ is a unit k-vector with 1 at the j^{th} position. Machines that are not operated remain frozen. Specifically, the system evolves according to

$$X_i\big(N_i(t+1)\big) = \begin{cases} f_{N_i(t)}\big(X_i(0), \dots, X_i(N_i(t)), W_i(N_i(t))\big), & \text{if } U_i(t) = 1, \\ X_i\big(N_i(t)\big), & \text{if } U_i(t) = 0, \end{cases} \tag{6.2}$$

[1] The state $X(n)$ is similar to state s_n of Markov decision model adopted in Chapter 2, even though, as noted in this chapter, machines are not necessarily described by Markov processes.

[2] The control action $U(t)$ is similar to the control action a_t of the Markov decision model of Chapter 2.

and

$$N_i(t+1) = \begin{cases} N_i(t), & \text{if } U_i(t) = 0, \\ N_i(t)+1, & \text{if } U_i(t) = 1, \end{cases} \tag{6.3}$$

for all $i = 1, 2, \ldots, k$. Thus $N_i(t)$, the local time of machine i, is incremented only when the controller operates on machine i at t; i.e., only when $U_i(t) = 1$. $\{W_i(n);\ i = 1, \ldots, k;\ n = 0, 1, \ldots\}$ is a sequence of independent primitive random variables that are independent of $\{X_1(0), \ldots, X_k(0)\}$ and have known statistical description[3].

Machine i generates a reward only when it is operated. Let $R_i(t)$ denote the reward generated by machine i at time t; then

$$R_i(t) = R_i\big(X(N_i(t)), U_i(t)\big) = \begin{cases} R_i\big(X_i(N_i(t))\big), & \text{if } U_i(t) = 1, \\ 0, & \text{if } U_i(t) = 0, \end{cases} \tag{6.4}$$

A *scheduling policy* $\gamma := (\gamma_1, \gamma_2, \ldots)$ is a decision rule such that at each time instant t, the control action $U(t)$ takes values in $\{e_1, \ldots, e_k\}$ according to[4]

$$U(t) = \gamma_t\big(Z_1(t), \ldots, Z_k(t), U(0), \ldots, U(t-1)\big), \tag{6.5}$$

where

$$Z_i(t) = [X_i(0), \ldots, X_i(N_i(t))].$$

The MAB problem is the following: Determine a scheduling policy that maximizes

$$J^\gamma := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{k} R_i\big(X_i(N_i(t)), U_i(t)\big) \,\middle|\, Z(0)\right]. \tag{6.6}$$

subject to (6.2)–(6.5).

This problem was originally formulated around 1940. It was known that it can be solved by a stochastic dynamic programming (SDP) approach, but no substantial progress in understanding the nature of its optimal solution was made until Gittins and Jones [89] showed that an optimal solution is of the *index type*. That is, for each bandit process one can compute what Gittins called a *dynamic allocation index* (DAI), which depends only on that process, and then at each time the controller operates on the bandit process with the

[3] As it is usually assumed, the processes $\{X_i(N_i(t)), W_i(N_i(t)); N_i(t) = 0, 1, \ldots t; t = 0, 1, 2, \ldots; i = 1, \ldots, k\}$ are defined on the same probability space $(\Omega, \mathcal{F}, P)$.

[4] Formally $U(t)$ needs to be measurable with respect to appropriate σ-fields. However, in this exposition we adopt an informal approach and do not explicitly talk about measurability requirements.

highest index. Thus, finding an optimal scheduling policy, which originally requires the solution of a k-armed bandit problem, reduces to determining the DAI for k single-armed bandit problems, thereby reducing the complexity of the problem exponentially. The DAI was later referred to as the *Gittins index* in honor of Gittins's contribution.

Following Gittins several researchers provided alternative proofs of the optimality of the *Gittins index* rule (see [249, 240, 246, 85, 22, 29, 232, 166, 115, 116, 123, 129, 166, 167] and [88] and references therein). Another formulation of the MAB problem with a performance criterion described by a "learning loss" or "regret" has been investigated in the literature [156, 7, 8, 3, 5, 4]. We will not discuss this formulation as we believe it is not directly related to sensor management problems.

Before we present the solution of the MAB problem we briefly present the method of forward induction. Knowing when forward induction is optimal is critical in understanding the nature of the solution of the MAB problem.

2.2 On Forward Induction

For centralized stochastic control problems, SDP provides a methodology for sequentially determining optimal decision rules/control laws/decision strategies. The SDP algorithm proceeds backward in time and at every stage t determines an optimal decision rule by quantifying the effect of every decision on the current and future conditional expected rewards. This procedure is called *backward induction*. SDP provides a powerful methodology for stochastic optimization, but it is computationally hard in many instances.

A procedure (computationally) simpler than backward induction is to make, at each decision time t, a decision that maximizes the conditional expected reward acquired at t. This procedure concentrates only on the present and completely ignores the future; it is called a *myopic approach* and results in a *myopic policy*. In general, myopic policies are not optimal.

The notion of a myopic policy can be extended to form T*-step-look-ahead policies*: make, at each decision time t, a decision that maximizes the conditional expected reward acquired at t plus the conditional expected reward acquired over the next T stages. In general, T-step-look-ahead policies are suboptimal. As T increases their performance improves at the cost of an increase in computational complexity.

The notion of a T-step-look-ahead policy can be extending as follows: allow the number τ of steps over which we look ahead at each stage to depend on how the system evolves while these steps are taking place; thus the number τ

of steps is a stopping time[5] with respect to the increasing family of σ-fields that describe the information accumulated by the decision maker/controller during the evolution of the process. This extension of a T-step-look-ahead policy involves two maximizations. An inner maximization that assumes that decision rule for taking a sequence of decisions is given and chooses a stopping time τ to maximize the conditional expected reward rate. The outer maximization is to choose a decision rule to maximize the result of the inner maximization for that decision rule. This extension of the T-step-look-ahead policy works as follows. At $t = 0$, given the information about the initial state of the process, select a decision rule and a stopping time τ_1 and follow it for the next τ_1 steps. The process of finding a new decision rule and a corresponding stopping time τ_2 is then repeated by conditioning on the information accumulated during the first τ_1 steps. The new rule is followed for the next τ_2 steps, and this procedure is repeated indefinitely. This procedure determines a policy for the entire horizon and is called *forward induction*; the policy produced by this procedure is called a *forward induction policy*.

In general forward induction results in suboptimal policies for stochastic control/optimization problems. This is demonstrated by the the following example from Gittins [87, pg 152].

Consider the problem of choosing a route for a journey by car. Suppose there are several different possible routes all of the same length which intersect at various points, and the objective is to choose a route which minimizes the time taken for the journey. The problem may be modeled as a Markov decision process by interpreting the distance covered so far as the "time" variable, the time taken to cover each successive mile as negative reward, the position as the state, and by choosing a value just less than one for the discount factor β. The action space $\mathcal{U}(x)$ has more than one element when the state x corresponds to cross-roads, the different control actions representing the various possible exits.

For this problem the first stage in a forward induction policy is to find a route ζ_1, and a distance σ_1 along ζ_1 from the starting point, such that the average speed in traveling the distance σ_1 along ζ_1 is maximized. Thus a forward induction policy might start with a short stretch of highway, which is followed by a very slow section, in preference to a trunk road which permits a good steady average speed. The trouble is that irrevocable decisions have to be taken at each cross-roads in the sense that those exits which are not chosen are not available later on.

[5]For the definition of stopping time, we refer the reader to Section 3 of the Appendix.

The above example illustrates the reason why, in general, forward induction results in suboptimal policies. Irrevocable decisions have to be made at some stage of the decision process, that is, some alternatives that are available at that stage and are not chosen, do not remain available later.

Forward induction policies are optimal if the decisions made at any stage are not irrevocable; that is, any alternative that is available at any stage and is not chosen, may be chosen at a later stage and with exactly the same sequence of rewards (apart from a discount factor). Thus, there is no later advantage of not choosing a forward induction policy.

In the next section we explain why the MAB problem belongs to a class of stochastic controlled processes for which forward induction results in optimal policies.

2.3 Key Features of the Classical MAB Problem and the Nature of its Solution

Four features delimit the MAB problem within the general class of stochastic control problems:

(F1) only one machine is operated at each time instant. The evolution of the machine that is being operated is uncontrolled; that is, the processor chooses which machine to operate but not how to operate it;

(F2) machines that are not operated remain frozen;

(F3) machines are independent;

(F4) frozen machines contribute no reward.

Features (F1)–(F4)[6] imply that an optimal policy can be obtained by forward induction. Decisions made at any instant of time t are not irrevocable since any bandit process that is available for continuation at t but is not chosen at t, can be continued at any later time instant with exactly the same resulting sequence of rewards, apart from the discount factor. This means that there is no later compensation for any initial loss resulting from not choosing a forward induction policy. Consequently, without any loss of optimality, we can restrict attention to forward induction policies. The first stage of a forward induction policy must be such that the expected discounted reward per unit of expected

[6]In Section 3.6 we show that feature (F4) is not essential for obtaining an optimal policy by forward induction.

discounted time up to an arbitrary stopping time is maximized. Gittins [87] argued (and proved rigorously) that this maximum can be achieved by a policy under which only one bandit process is continued up to the stopping time in question. To determine the bandit process to be continued in the first stage and the corresponding stopping time the following arguments, due to Whittle [249], can be used. Consider arm i of the MAB process and let $x_i(0)$ be its state at $t = 0$ and let $N_i(t) = 0$. Suppose two options are available at $t = 0$: continue the process, or retire and receive a retirement reward v. Let $v_{X_i}\big(x_i(0)\big)$ be the retirement reward that can be offered at $x_i(0)$ so that the controller is indifferent to both options. This reward is given by

$$v_{X_i}\big(x_i(0)\big) := \max_{\tau>0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t R_i\big(X_i(t)\big) \,\middle|\, x_i(0)\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t \,\middle|\, x_i(0)\right]}. \tag{6.7}$$

Then $v_{X_i}\big(x_i(0)\big)$ is the maximum expected discounted reward per unit of expected discounted time that can be obtained at the first stage of a forward induction policy that continues the arm i with initial state $x_i(0)$. The corresponding stopping time $\tau_i\big(x_i(0)\big)$ is the first time at which the expected discounted reward per unit of expected discounted time equals $v_{X_i}\big(x_i(0)\big)$. Consequently, at $t = 0$ an optimal forward induction policy continues an arm j such that

$$v_{X_j}\big(x_j(0)\big) = \max_i v_{X_i}\big(x_i(0)\big).$$

Arm j is continued until $\tau_j\big(x_j(0)\big) - 1$. This constitutes the first stage of an optimal forward induction policy, and can be summarized as follows:

Step 1 Determine $v_{X_i}\big(x_i(0)\big)$, $i = 1, \ldots, k$.

Step 2 Select an arm j such that

$$j = \arg\max_i v_{X_i}\big(x_i(0)\big).$$

Continue operating arm j until the minimum time that achieves the maximum in the right hand side of (6.7).

At the end of the first stage, the forward induction proceeds in the same way as at $t = 0$. Let τ_l be the end of the l^{th} stage of an optimal forward induction policy for the MAB problem. At time τ_l we must decide which arm to operate next. The random time τ_l is a stopping time with respect to the family of σ-fields $\big\{ \bigvee_{i=1}^k \mathcal{F}^i(t),\, t = 0, 1, 2, \ldots \big\}$, where $\mathcal{F}^i(t)$ is defined as the sigma field $\sigma\big(X_i(0), \ldots, X_i(N_i(t))\big)$, $i = 1, 2, \ldots, k$. For any sample point ω in

the sample space Ω of the MAB process (see footnote 3), let $\{x_i(0), x_i(1), \ldots, x_i(N_i(\tau_l(\omega))), i = 1, 2, \ldots, k\}$ be the realization of the MAB process obtained under an optimal forward induction policy up to $\tau_l(\omega)$. The decision made by an optimal forward induction policy at $\tau_l(\omega)$ can be described by the following two-step process:

Step 1 For each $i = 1, \ldots, k$, let $x_i^l(\omega) := \big(x_i(0), \ldots, x_i(N_i(\tau_l(\omega)))\big)$, and determine

$$v_{X_i}\big(x_i^l(\omega)\big) = \max_{\tau > \tau_l(\omega)} \frac{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\tau-1} \beta^t R_i\big(X_i\big(N_i(\tau_l) + t - \tau_l(\omega)\big)\big) \,\middle|\, x_i^l(\omega)\right]}{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\tau-1} \beta^t \,\middle|\, x_i^l(\omega)\right]}, \tag{6.8}$$

and the stopping time $\tau_i\big(x_i^l(\omega)\big)$ achieves the maximum at the right hand side of (6.8).

Step 2 Select arm j such that

$$j = \arg\max_i v_{X_i}\big(x_i^l(\omega)\big)$$

and operate it for $\tau_j\big(x_j^l(\omega)\big) - 1 - \tau_l(\omega)$ steps/time units.

The number $v_{X_i}\big(x_i^l(\omega)\big)$ that denotes the maximum expected discounted reward per unit of expected discounted time obtained by arm i at $x_i^l(\omega)$ is the *Gittins index* of arm i at $x_i^l(\omega)$.

We examine now what happens between the stopping times τ_l and τ_{l+1}, $l = 0, 1, \ldots$. Suppose arm j is continued at τ_l. Then at $t = \tau_l + 1, \ldots, \tau_{l+1} - 1$ the *Gittins index* of arm j is higher than its index at τ_l (see [240]). The Gittins indices of all other arms remain the same as in τ_l since these arms are frozen. Thus an equivalent method to describe the above procedure is to consider the MAB problem at each instant of time t and continue the arm with the highest index. This observation allows us to understand the nature of the solution of some generalizations of the MAB problem (e.g., the arm-acquiring bandit).

In summary, an optimal scheduling policy for the MAB problem can be obtained by forward induction. Forward induction is computationally less intensive than backward induction; an optimal forward induction policy can be obtained by solving k one-dimensional problems (by computing the Gittins indices of each bandit process) instead of one k-dimensional problem.

In many formulations of the MAB problem it is assumed that the machines are Markovian[7], that is

$$f_{N_i(t)}\big(X_i(N_i(0)),\ldots,X_i(N_i(t)),W_i(N_i(t))\big) = f_{N_i(t)}\big(X_i(N_i(t)),W_i(N_i(t))\big). \quad (6.9)$$

In this case (6.8) reduces to

$$v_{X_i}\big(x_i^l(\omega)\big) = v_{X_i}\big(x_i(N_i(\tau_l))\big) = \max_{\tau>\tau_l(\omega)} \frac{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\tau-1}\beta^t R_i\big(X_i\big(N_i(\tau_l(\omega))+t-\tau_l(\omega)\big)\big)\,\middle|\, x_i(N_i(\tau_l))\right]}{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\tau-1}\beta^t \,\middle|\, x_i(N_i(\tau_l(\omega)))\right]}, \quad (6.10)$$

and such an index is considerably easier to compute (see [240, 127, 100, 126]).

2.4 Computational Issues

We concentrate on the classical MAB problem where the machines are time-homogeneous finite-state Markov chains (MCs)[8]. We assume that machine i, $i = 1, 2, \ldots, k$, has state space $\{1, 2, \ldots, \Delta_i\}$ and matrix of transition probabilities $P^{(i)} := \big\{P^{(i)}_{a,b},\ a, b \in \{1, 2, \ldots, \Delta_i\}\big\}$.

In this case we do not need to keep track of the local time of the machines because of the Markovian property and the time-homogeneity of the Markov chains. The evolution of machine i, $i = 1, 2, \ldots, k$, can be described by the following set of equations. If $X_i(t) = a$, $a \in \{1, 2, \ldots, \Delta_i\}$, then

$$X_i(t+1) = a, \quad \text{if } U_i(t) = 0, \quad (6.11)$$

$$\mathrm{P}\big(X_i(t+1) = b \mid X_i(t) = a\big) = P^{(i)}_{a,b}, \quad \text{if } U_i(t) = 1. \quad (6.12)$$

[7]Such formulations are considered in Section 2 of Chapter 7 where applications of MAB theory to sensor management is considered.

[8]Throughout this chapter we assume that the state of each machine is perfectly observable. For the case where state is imperfectly observable we refer the reader to [171].

Further, $\mathbf{X}(t) := \big(X_1(t), X_2(t), \ldots, X_k(t)\big)$ is an information state (sufficient statistic, see Chapter 3 and [153]) at time t. Thus (6.10) can be rewritten as

$$\nu_{X_i}\big(x_i(t)\big) = \max_{\tau > t} \frac{\mathbb{E}\left[\sum_{t'=t}^{\tau-1} \beta^{t'} R_i\big(X_i(t')\big) \,\middle|\, x_i(t)\right]}{\mathbb{E}\left[\sum_{t'=t}^{\tau-1} \beta^{t'} \,\middle|\, x_i(t)\right]} \tag{6.13}$$

Thus to implement the index policy we need to compute, for each machine i, $i = 1, \ldots, k$, the Gittins index of state $x_i(t)$. This can be done in either an off-line or an on-line manner. For an off-line implementation, we must compute the Gittins index corresponding to each state $x_i \in \{1, 2, \ldots, \Delta_i\}$ of each machine i, $i = 1, \ldots, k$. This computation can be done off-line and we only need to store the values of $\nu_{X_i}(x_i)$, $x_i \in \{1, \ldots, \Delta_i\}$ for each machine i, $i = 1, \ldots, k$. For an on-line implementation, at stage 0 we need to compute $\nu_{X_i}\big(x_i(0)\big)$ for each machine i, $i = 1, \ldots, k$ where $x_i(0)$ is given[9]. We operate machine $j = \arg\max_i \nu_{X_i}\big(x_i(0)\big)$ until the smallest time τ_1 at which machine j achieves its Gittins index. At any subsequent stage l, we only need to compute the Gittins index of the machine operated at stage $l-1$. The computation of these Gittins indices has to be done on-line, but only for the stopping states that are reached during the evolution of the bandit processes. To achieve such a computation we need to store the reward vector and the matrix of transition probabilities for each machine.

We next describe the notions of continuation and stopping sets, which are key concepts for the off-line and on-line computation of the Gittins index rule (see [87]). Suppose we start playing machine i which is initially in state x_i. Then the state space $\{1, 2, \ldots, \Delta_i\}$ can be partitioned into two sets $C_i(x_i)$ (the continuation set of x_i) and $S_i(x_i)$ (the stopping set of x_i). When the state of machine i is in $C_i(x_i)$ we continue processing the machine. We stop processing machine i the first instant of time the state of the machine is in $S_i(x_i)$. Therefore, the Gittins index policy can be characterized by determining $C_i(x_i)$ and $S_i(x_i)$ for each $x_i \in \{1, 2, \ldots, \Delta_i\}$.

A computational technique for the off-line computation of the Gittins index rule, proposed by Varaiya, Walrand and Buyukkoc in [240], is based on the following observation. If for $a, b \in \{1, 2, \ldots, \Delta_i\}$ $\nu_{X_i}(a) > \nu_{X_i}(b)$, then $b \in S_i(a)$ and $a \in C_i(b)$. If $\nu_{X_i}(a) = \nu_{X_i}(b)$ then either $a \in C_i(b)$ and $b \in S_i(a)$, or $a \in S_i(b)$ and $b \in C_i(a)$. Thus, to determine $C_i(x_i)$ and $S_i(x_i)$ for each $x_i \in \{1, 2, \ldots, \Delta_i\}$ we must find first an ordering $l_1, l_2, \ldots, l_{\Delta_i}$ of

[9] It is the observed state of machine i at time 0.

the states of machine i such that

$$\nu_{X_i}(l_1) \geq \nu_{X_i}(l_2) \geq \cdots \geq \nu_{X_i}(l_{\Delta_i}), \tag{6.14}$$

and then set for all l_j, $j = 1, 2, \ldots, \Delta_i$,

$$\begin{aligned} C_i(l_j) &= \{l_1, l_2, \ldots, l_j\}, \\ S_i(l_j) &= \{l_{j+1}, l_{j+2}, \ldots, l_{\Delta_i}\}. \end{aligned} \tag{6.15}$$

To obtain such an ordering $l_1, l_2, \ldots, l_{\Delta_i}$ the following computational procedure was proposed in [240].

Given a machine i, $i = 1, 2, \ldots, k$, with state space $\{1, 2, \ldots, \Delta_i\}$, matrix of transition probabilities $P^{(i)} := \left\{P^{(i)}_{a,b},\ a, b \in \{1, 2, \ldots, \Delta_i\}\right\}$, and reward function $R_i(x_i),\ x_i \in \{1, 2, \ldots, \Delta_i\}$ set

$$l_1 = \arg\max_{x_i} R_i(x_i). \tag{6.16}$$

Break ties by choosing the smallest x_i that satisfies (6.16). The Gittins index of state l_1 is

$$\nu_{X_i}(l_1) = R_i(l_1). \tag{6.17}$$

States $l_2, l_3, \ldots, l_{\Delta_i}$ can be recursively determined by the following procedure. Suppose $l_1, l_2, \ldots, l_{n-1}$ have been determined; then

$$\nu_{X_i}(l_1) \geq \nu_{X_i}(l_2) \geq \cdots \geq \nu_{X_i}(l_{n-1}). \tag{6.18}$$

Define

$$P^{(i,n)}_{a,b} := \begin{cases} P^{(i)}_{a,b}, & \text{if } b \in \{l_1, l_2, \ldots, l_{n-1}\} \\ 0, & \text{otherwise} \end{cases}, \tag{6.19}$$

and the vectors

$$\mathbf{R}_i := \big(R_i(1), \quad R_i(2), \quad \cdots, \quad R_i(\Delta_i)\big)^{\intercal}, \tag{6.20}$$

$$\mathbf{1} := \underbrace{\big(1, \quad 1, \quad \cdots, \quad 1\big)^{\intercal}}_{\Delta_i \text{times}}, \tag{6.21}$$

$$D^{(i,n)} := \beta\big[I - \beta P^{(i,n)}\big]^{-1}\mathbf{R}_i = \begin{pmatrix} D_1^{(i,n)} \\ D_2^{(i,n)} \\ \cdots \\ D_{\Delta_i}^{(i,n)} \end{pmatrix}, \tag{6.22}$$

$$B^{(i,n)} := \beta\big[I - \beta P^{((i,n))}\big]^{-1}\mathbf{1} = \begin{pmatrix} B_1^{(i,n)} \\ B_2^{(i,n)} \\ \cdots \\ B_{\Delta_i}^{(i,n)} \end{pmatrix}. \tag{6.23}$$

Then

$$l_n = \arg \max_{a \in \{1,2,\ldots,\Delta_i\} \setminus \{l_1, l_2, \ldots, l_{n-1}\}} \frac{D_a^{(i,n)}}{B_a^{(i,n)}}, \tag{6.24}$$

and

$$\nu_{X_i}(l_n) = \frac{D_{l_n}^{(i,n)}}{B_{l_n}^{(i,n)}}. \tag{6.25}$$

Another method for off-line computation of Gittins index, which has the same complexity as the algorithm of [240] presented above, appears in [29].

The following method for on-line implementation of the Gittins index was proposed by Katehakis and Veinott in [127]. As explained earlier, to obtain the Gittins index for state x_i only the sets $C_i(x_i)$ and $S_i(x_i)$ need to be determined. In [127], Katehakis and Veinott proposed the "restart in x_i" method to determine these sets. According to this method, we consider an alternative problem where in any state $a \in \{1, \ldots, \Delta_i\}$ we have the option either to continue operating machine i from state a or to instantaneously switch to state x_i and continue operating the machine from state x_i. The objective is to choose the option that results in the maximum expected discounted reward over an infinite horizon (see Chapter 2, Section 2.2). This approach results in a dynamic

program

$$V(a) = \max \left\{ R_i(a) + \beta \sum_{b \in \{1,\ldots,\Delta_i\}}, P_{a,b}^{(i)} V(b) \ , \ R(x_i) + \beta \sum_{b \in \{1,\ldots,\Delta_i\}}, P_{x_i,b}^{(i)} V(b) \right\}, \quad a \in \{1, \ldots, \Delta_i\} \quad (6.26)$$

that can be solved by various standard computational techniques for finite state Markov decision problems (see Chapter 2). The solution of this dynamic program determines the sets $C_i(x_i)$ and $S_i(x_i)$. These sets are given by

$$C_i(x_i) = \left\{ a \in \{1, \ldots, \Delta_i\} : R_i(a) + \beta \sum_{b \in \{1,\ldots,\Delta_i\}} P_{a,b}^{(i)} V(b) \geq V(x_i) \right\} \quad (6.27)$$

$$S_i(x_i) = \left\{ a \in \{1, \ldots, \Delta_i\} : R_i(a) + \beta \sum_{b \in \{1,\ldots,\Delta_i\}} P_{a,b}^{(i)} V(b) < V(x_i) \right\} \quad (6.28)$$

and the Gittins index is given by

$$\nu_{X_i}(x_i) = (1 - \beta) V(x_i) \quad (6.29)$$

Another method for on-line implementation similar in spirit to [240] appears in E. L. M. Beale's discussion in [87].

Several variations of the classical MAB problem have been considered in the literature. We briefly present them in Section 3.

3. Variants of the Multi-armed Bandit Problem

In this section we present various extensions of the classical MAB problem. In general, in these extensions, forward induction does not provide a methodology for determining an optimal scheduling policy. Index-type solutions are desirable because of their simplicity, but, in general, they are not optimal. We identify conditions under which optimal index-type solutions exist.

3.1 Superprocesses

A superprocess consists of k independent components and one controller/ processor. At each time t each component $i = 1, 2, \ldots, k$ accepts control inputs $U_i(t) \in \mathcal{U}_i := \{0, 1, \ldots, M_i\}$. The control action $U_i(t) = 0$ is a freezing control; the action $U_i(t) = j, j = 1, 2, \ldots, M_i$ is a continuation control. Thus, each component of a superprocess is a generalization of an arm of a classical MAB problem (where at each t $U_i(t) \in \{0, 1\}$.) In fact, each

component of a superprocess is a controlled stochastic process. For any fixed control law this stochastic process is a single-armed bandit process. Consequently, each component of a superprocess consists of a collection of bandit processes/machines, each corresponding to a distinct control law. Component $i = 1, 2, \ldots, k$ evolves as follows

$$X_i\big(N_i(t+1)\big) = X_i\big(N_i(t)\big) \quad \text{if } U_i(t) = 0, \tag{6.30}$$

and

$$X_i\big(N_i(t+1)\big) = f_{N_i(t)}\big(X_i(0), \ldots, X_i(N_i(t)), U_i(t), W_i(N_i(t))\big) \quad \text{if } U_i(t) \neq 0, \tag{6.31}$$

where $N_i(t)$ is the local time of component i at each t and $\{W_i(n), n = 1, 2, \ldots\}$ is a sequence of independent random variables that are also independent of $\{X_1(0), X_2(0), \ldots, X_k(0)\}$. Furthermore, the sequences $\{W_i(n); n = 1, 2, \ldots\}$, $\{W_j(n); n = 1, 2, \ldots\}$, $i \neq j$, $i, j = 1, 2, \ldots, k$, are independent.

A reward sequence $\{R_i\big(X_i(t), U_i(t)\big); t = 1, 2, \ldots\}$ is associated with each component $i = 1, 2, \ldots, k$, such that

$$R_i(t) = R_i\big(X_i(t), U_i(t)\big), \quad \text{if } U_i(t) \neq 0, \tag{6.32}$$

and

$$R_i(t) = 0, \quad \text{if } U_i(t) = 0\,. \tag{6.33}$$

At each time t the controller/processor can choose to operate/continue exactly one component. If the controller chooses component j at t, i.e., $U_i(t) = 0$ for all $i \neq j$, and $U_j(t)$ takes values in $\{1, 2, \ldots, M_j\}$, a reward $R_j\big(X_j(N_j(t)), U_j(t)\big)$ is acquired according to (6.32).

A scheduling policy $\gamma := (\gamma_1, \gamma_2, \ldots)$ is a decision rule such that the action $U(t) = (U_1(t), U_2(t), \ldots, U_k(t))$ is a random variable taking values in $\bigcup_{i=1}^{k} \{0\}^{i-1} \times \{1, 2, \ldots, M_i\} \times \{0\}^{k-i}$, and

$$U(t) = \gamma_t\big(Z_1(t), Z_2(t), \ldots, Z_k(t), U(0), \ldots, U(t-1)\big), \tag{6.34}$$

where

$$Z_i(t) := [X_i(0), X_i(1), \ldots, X_i(N_i(t))]. \tag{6.35}$$

The objective in superprocesses is to determine a scheduling policy γ that maximizes

$$J^{\gamma} := \mathbb{E}^{\gamma}\left[\sum_{t=0}^{\infty} \beta^t \sum_{j=1}^{k} R_j\big(X_j(N_j(t)), U_j(t)\big) \,\middle|\, Z(0)\right] \tag{6.36}$$

subject to (6.30)–(6.35) and the above constraints on $U(t)$, $t = 0, 1, 2, \ldots$ where

$$Z(0) := [X_1(0), X_2(0), \ldots, X_k(0)].$$

Even though features (F2)-(F4) of the MAB problem are present in the superprocess problem, (F1) is not, and as a result of this superprocesses do not in general admit an index-type of solution. Specifically: in the MAB problem, once a machine/process is selected for continuation, the evolution of this process and the accumulated rewards are uncontrolled; on the contrary, in superprocesses, once a component is selected for continuation, the evolution of this component and the accumulated rewards are controlled. Choosing the control law that maximizes the infinite horizon expected discounted reward for the component under consideration leads to a standard stochastic control problem which can only be solved optimally by backward induction.

Consequently, superprocesses are more complex problems than standard MAB problems. There is one situation where superprocesses admit an index form type of solution, namely, when each component has a *dominating machine*.

The concept of a dominating machine can be formally described as follows. Consider a machine $\{X(n), R\big(X(n)\big), n = 0, 1, 2, \ldots\}$ and let $\mu \in \mathbb{R}$. Define

$$\mathcal{L}(X, \mu) := \max_{\tau > 0} \mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t \left[R\big(X(t)\big) - \mu\right]\right], \tag{6.37}$$

where τ ranges over all stopping times of $\{\mathcal{F}_t := \sigma\big(X(0), X(1), \ldots, X(t)\big), t = 0, 1, 2, \ldots\}$ (see Appendix Section 3 for a discussion on σ-fields and stopping times). Notice that $\mathcal{L}(X, \mu) \geq 0$ since for $\tau = 1$ the right hand side of (6.37) is equal to zero with probability one.

DEFINITION 6.1 *We say that machine* $\{X(n), R\big(X(n)\big); n = 0, 1, 2, \ldots\}$ dominates *machine* $\{Y(n), R\big(Y(n)\big); n = 0, 1, 2, \ldots\}$ *if*

$$\mathcal{L}(X, \mu) \geq \mathcal{L}(Y, \mu), \quad \forall \mu \in \mathbb{R}. \tag{6.38}$$

This inequality can be interpreted as follows. Suppose one operates machines $\mathcal{M}(X) := \{X(n), R\big(X(n)\big); n = 0, 1, 2, \ldots\}$ and $\mathcal{M}(Y) := \{Y(n), R\big(Y(n)\big); n = 0, 1, 2, \ldots\}$ up to some random time after which one retires and receives the constant reward μ at each subsequent time. Then $\mathcal{M}(X)$ dominates $\mathcal{M}(Y)$ if and only if it is optimal to choose $\mathcal{M}(X)$ over $\mathcal{M}(Y)$ for any value of the retirement reward μ.

As mentioned above, a component of a superprocess is a collection of bandit processes $\{X^{\gamma_i}(n), R\big(X^{\gamma_i}(n), U^{\gamma_i}(n)\big); n = 0, 1, 2, \dots\}$, $\gamma_i \in \Gamma_i$, where Γ_i is the set of all possible control laws for component i.

DEFINITION 6.2 *A component of a superprocess is said to have a dominating control law γ^* and a corresponding dominating machine $\{X^{\gamma^*}(n), R\big(X^{\gamma^*}(n)\big); n = 1, 2, \dots\}$ if*

$$\mathcal{L}\big(X^{\gamma^*}, \mu\big) \geq \mathcal{L}\big(X^{\gamma}, \mu\big), \qquad \forall \gamma \in \Gamma, \forall \mu \in \mathbb{R}$$

where Γ is the set of all control laws for that component.

When each component of the superprocess has a dominating machine an index-type solution is optimal for the following reason. In every component of the superprocess one can restrict attention, without any loss of optimality, to its dominating machine. Each dominating machine is a single-armed bandit process. Thus, the superprocess problem reduces to a MAB problem for which an optimal solution is of the index type.

The condition that each component of a superprocess has a dominating machine is quite restrictive and difficult to satisfy in most problems.

3.2 Arm-acquiring Bandits

The arm-acquiring bandit problem is a variation of the MAB problem where one permits arrival of new machines. At time t, $K(t)$ independent machines are available. The machines available at t were either available at $t = 0$ or arrived during $1, \dots, t-1$. Denote these machines by $\{\big(X_i(N_i(t)), R_i(X_i(N_i(t)))\big); N_i(t) = 0, 1, 2, \dots, t; i = 1, 2, \dots, K(t); t = 0, 1, 2, \dots\}$. At each time instant, the controller decides to apply a continuation control to only one of the available machines and all other machines remain frozen. Define $U(t) := \big(U_1(t), \dots, U_{K(t)}(t)\big)$. Then $U(t) \in \{\, e_1\big(K(t)\big), \dots, e_{K(t)}\big(K(t)\big)\,\}$, where $e_i(j) = (0, \dots, 0, 1, 0, \dots, 0)$ is a j-dimensional unit vector with 1 at the i^{th} position. The machines available at time t are independent and evolve in the same way as in the classical MAB problem.

At time t a set $A(t)$ of new machines arrive. These machines are available for operation from $(t+1)$ on and are independent of each other and of the $K(t)$ previous machines. Let $|A(t)|$ denote the number of machines in $A(t)$. Then,

$$K(t+1) = K(t) + |A(t)|$$

It is assumed that $\{\, |A(t)|; t = 1, 2, \dots\}$ is a sequence of i.i.d. random variables. Further, $|A(t)|$ is independent of $U(0), \dots, U(t)$.

In this context, a *scheduling policy* $\gamma := (\gamma_1, \gamma_2, \dots)$ is a decision rule such that the action $U(t)$ at any time t is a random variable taking values in $\{ e_1\big(K(t)\big), \dots, e_{K(t)}\big(K(t)\big) \}$ and

$$U(t) = \gamma_t\big(Z_1(t), \dots, Z_{k(t)}(t), U(0), \dots, U(t-1)\big), \tag{6.39}$$

where

$$Z_i(t) = [X_i(0), \dots, X_i(N_i(t))];$$

that is, $U(t)$ denotes the machine operated at time t, and this decision depends on all past states of all machines.

The arm-acquiring bandit problem is to determine a scheduling policy that maximizes

$$J^\gamma := \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{K(t)} R_i\big(X_i(N_i(t)), U_i(t)\big) \,\middle|\, Z(0) \right], \tag{6.40}$$

subject to the aforementioned constraints on the evolution of the machines and the arrival of new machines.

Nash [179] first considered the arm-acquiring bandit problem using Hamiltonian and dynamic programming and he did not obtain an index-type of solution. Whittle [250] first showed that the *Gittins index* policy is optimal for the arm-acquiring bandit. Similar results on the optimality of the Gittins index rule for arm-acquiring bandits were later obtained by [240, 116]. Here we present briefly the arguments that lead to the optimality of the Gittins index rule.

Decisions are not irrevocable due to the following: bandit processes are independent; processes that are not operated on remain frozen; future arrivals are independent of past decisions; and the arrival process is a sequence of independent identically distributed random variables. Therefore, by the arguments presented in Section 2.3, forward induction obtains an optimal scheduling policy—at each instant of time continue the machine with the highest Gittins index. The expressions for the Gittins index of each machine are the same as in Equation (6.8). If the machines are described by Markov processes then their dynamics evolve as in (6.9) and the Gittins indices are given by (6.10).

3.3 Switching Penalties

In MAB problem with switching penalties we have the same model as in the classical MAB problem with one additional feature. Every time the processor switches from one machine to another, a switching penalty (switching cost c or

switching delay d) is incurred. The inclusion of switching penalties is a realistic consideration. When the processor switches between different machines a new setup may be needed; during this setup time no bandit process is continued (therefore, no reward is acquired) and this lack of continuation can be modeled by a switching cost or switching delay.

The inclusion of switching penalties drastically alters the nature of the bandit problem. An index form of solution is no longer optimal. This has been shown in [12] and is illustrated by the following example from [11].

Consider a two-armed bandit problem with switching penalties. Each arm is described by a three-state Markov chain. The transition probabilities of the both are given by $P_{X_{t+1}|X_t}(2|1) = 1$, $P_{X_{t+1}|X_t}(3|2) = 1$, $P_{X_{t+1}|X_t}(3|3) = 1$, further, both Markov chains start in state 1. The rewards of the first arm are given by $R_1(1) = 20$, $R_1(2) = 18$, $R_1(3) = 0$; and of the second arm are given by $R_2(1) = 19$, $R_2(2) = 17$, $R_1(3) = 0$. Assume the switching cost $c = 3$ and the discount factor $\beta = 0.5$. If we operate the arms according to the Gittins index policy, the order of operation is 1,2,1,2 and the corresponding rewards are $(20-3)+(19-3)\beta+(18-3)\beta^2+(17-3)\beta^3 = 30.5$, whereas a policy that operates in order 1,1,2,2 yields a reward $(20-3)+18\beta+(19-3)\beta^2+17\beta^3 = 32.125$. Thus, the Gittins index policy is not optimal.

The nature of optimal scheduling/allocation strategies for the general MAB problem with switching penalties and an infinite horizon expected discounted reward including switching penalties is not currently known. Explicit solutions of special cases of the problem have been determined in Van Oyen et al. [234, 235]. Agrawal et al. [1] determined an optimal allocation strategy for the MAB problem with switching cost and the "learning loss" or "regret" criterion. Asawa and Teneketzis [11] determined qualitative properties of optimal allocation/scheduling strategies for the general MAB problem with an infinite horizon expected discounted reward minus switching penalties performance criterion. In this chapter we only consider switching costs. The main result in [11] states the following. Suppose that at $t = 0$, it is optimal to select machine j for continuation. If at $t = 0$ no switching penalty is incurred, it is optimal to continue the operation of machine j until its Gittins index corresponding to $x_j(0)$ is achieved. If at $t = 0$ a switching cost c is incurred, it is optimal to continue the operation of machine j until a switching index

$$\nu^s_{X_j}(x_j(0)) = \max_{\tau>0} \frac{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t R_j(t) - c \,\middle|\, x_j(0)\right]}{\mathbb{E}\left[\sum_{t=0}^{\tau-1} \beta^t \,\middle|\, x_j(0)\right]} \tag{6.41}$$

corresponding to $x_j(0)$ is achieved. In general, suppose that at decision epoch $\tau_l(\omega)$ it is optimal to select machine i for continuation. If machine i was operated at $\tau_l(\omega) - 1$ then it is optimal to continue the operation of machine i until its Gittins index corresponding to $(x_i^l(\omega))$ (and given by (6.8)) is achieved. If machine i was not operated at $\tau_l(\omega) - 1$, then it is optimal to continue its operation until a switching index

$$\nu_{X_i}^s\big(x_i^l\big) \;=\; \max_{\tau>\tau_l} \frac{\mathbb{E}\left[\sum_{t=\tau_l}^{\tau-1} \beta^t R_i\big(X_i\big(N_i(\tau_l)+t-\tau_l\big)\big) - \beta^{\tau_l} c \,\middle|\, x_i^l\right]}{\mathbb{E}\left[\sum_{t=\tau_l}^{\tau} \beta^t \,\middle|\, x_i^l\right]} \tag{6.42}$$

corresponding to $x_i^l(\omega)$ is achieved. (Recall that $x_i^l := x_i(0), \ldots, x_i(N_i(\tau_l)))$.

The stopping time $\tau^s\big(x_i^l\big)$ that achieves the maximum on the RHS of (6.42) is related to the stopping time $\tau\big(x_i^l(\omega)\big)$ that achieves the Gittins index as follows:

$$\tau^s\big(x_i^l(\omega)\big) \geq \tau\big(x_i^l(\omega)\big) \tag{6.43}$$

almost surely for all $x_i^l(\omega)$.

The main result in [11] does not describe which machine to select for continuation at each decision epoch. Such a selection must be determined by backward induction. Conditions under which it is possible to further simplify the search for an optimal allocation policy also appear in [11].

3.4 Multiple Plays

In MABs with multiple plays we have k independent processes/machines labeled $1, 2, \ldots, k$ and one controller that has m processors available ($m < k$). At each time instant the controller can allocate each processor to exactly one process. No process can be operated by more than one processor. Each bandit process and its evolution are modeled as in the classical MAB problem. A scheduling policy $\gamma := (\gamma_1, \gamma_2, \ldots)$ is a decision rule such that the action $U(t)$ at time t is a random variable taking values in $(d_1, d_2, \ldots, d_{\binom{k}{m}})$ where each d_i is a k-dimensional row vector consisting of m ones and $(k-m)$ zeros, and the positions of the ones indicate the machines/processes to which the processors are allocated. The objective in MABs with multiple plays is to determine a scheduling policy γ that maximizes

$$J^\gamma := \mathbb{E}^\gamma\left[\sum_{t=1}^{\infty} \beta^t \sum_{i=1}^{k} R_i\big(X_i(N_i(t)), U_i(t)\big) \,\middle|\, Z(0)\right], \tag{6.44}$$

subject to the constraints describing the evolution of the machines and the allocation of the processors, where

$$Z(0) := [X_1(0), X_2(0), \ldots, X_k(0)] \tag{6.45}$$

and

$$R_i\big(X_i(N_i(t)-1), U_i(t)\big) = \begin{cases} R_i\big(X_i(N_i(t))\big), & \text{if } U_i(t) = 1\,, \\ 0, & \text{otherwise.} \end{cases}$$

In general operating machines with the m highest Gittins indices is not an optimal policy for MAB with multiple plays (see [115, 198]). Anantharam et al. [7, 8] determined an optimal scheduling policy for MABs with multiple plays and the "learning loss" or "regret" criterion. Furthermore, Agrawal et al. [2] determined an optimal scheduling policy for the MAB problem with multiple plays, a switching cost, and the "learning loss" criterion. Pandelis and Teneketzis [188] determined a condition sufficient to guarantee the optimality of the policy that operates the machines with the m highest Gittins indices at each instant of time. (We call this strategy the *Gittins index rule for MABs with multiple plays* or briefly the *Gittins index rule*.) The sufficient condition of [188] can be described as follows. For each machine i, $i = 1, 2, \ldots, k$, let τ_l^i denote the successive stopping times at which the Gittins indices of machine i are achieved, and let $\nu_{X_i}\big(X_i(0), \ldots, X_i(\tau_l^i)\big)$ denote the $(l+1)^{\text{th}}$ successive Gittins index of the process i. For every realization ω of the evolution of machine i we have the corresponding realizations $\tau_l^i(\omega)$ and $\nu_{X_i}\big(X_i(0,\omega), \ldots, X_i(\tau_l^i(\omega),\omega)\big)$, $l = 1, 2, \ldots$ of machine i, $i = 1, \ldots, k$. Consider the following condition.

(C1) For any realization ω of the problem, for any machines i, j such that $i \neq j$ and positive integers p, q such that

$$\nu_{X_i}\big(X_i(0,\omega), \ldots, X_i(\tau_p^i(\omega),\omega)\big) > \nu_{X_j}\big(X_j(0,\omega), \ldots, X_j(\tau_q^j(\omega),\omega)\big)$$

we have

$$\nu_{X_i}\big(X_i(0,\omega), \ldots, X_i(\tau_p^i(\omega),\omega)\big)(1-\beta) > \nu_{X_j}\big(X_j(0,\omega), \ldots, X_j(\tau_q^j(\omega),\omega)\big)$$

The main result in [188] states that if condition (C1) is satisfied then the Gittins index rule is optimal. The essence of the result of [188] is the following. Forward induction does not, in general, lead to optimal processor allocation decisions in MABs with multiple plays because at each stage of the allocation

process the optimal scheduling policy jointly decides the m machines to be processed; thus, the forward induction arguments used in the classical MAB problem (and were discussed in Section 2.2) do not hold. Consequently, the full effect of future rewards has to be taken into account in determining an optimal scheduling policy. However, if the Gittins indices of different machines are sufficiently separated, the expected reward rate maximizing portions of each bandit process starting from its current history become the dominant factors in determining an optimal scheduling policy. In such situations, an optimal scheduling strategy can be determined by forward induction, and the Gittins index rule is optimal. Condition (C1) presents an instance where there is enough separation among the Gittins indices to guarantee the optimality of the Gittins index rule.

A search problem formulated as a MAB problem with multiple plays has been considered in [221]. Conditions under which the Gittins index rule is optimal for the above problem also appear in [221].

3.5 Restless Bandits

Restless bandits (RBs) consist of k independent machines and m identical processors, $m < k$. Each machine evolves over time even when it is not being processed, and hence is not a bandit process. Specifically, the evolution of machine i, $i = 1, 2, \ldots, k$, is described by

$$X_i(t+1) = f_{i,t}\left(X_i(0), \ldots, X_i(t), U_i(t), W_i(t)\right), \tag{6.46}$$

where $U_i(t) \in \{0, 1\}$, $U_i(t) = 0$ (respectively 1) means that machine i is not processed (respectively processed) at time t, and $\{W_i(t), t = 0, 1, 2, \ldots\}$ is a sequence of primitive random variables that are independent of $X_1(0), X_2(0), \ldots, X_k(0)$ and have known statistical description; furthermore, $\{W_i(t), t = 0, 1, 2, \ldots\}$ and $\{W_j(t), t = 0, 1, 2, \ldots\}$, $i \neq j$ are independent. (The reader is invited to contrast (6.46) with (6.2)). The reward received from machine i at time t is $R_i\left(X_i(t), U_i(t)\right)$[10]. At each instant of time each processor can process exactly one machine. Each machine can be processed by at most one processor. A scheduling policy is defined in exactly the same way as in the MAB problem with multiple plays. The performance criterion is defined by (6.44) and (6.45). The objective is to determine a scheduling policy to maximize an infinite horizon expected discounted reward criterion given by (6.44).

In general, forward induction does not result in an optimal allocation strategy for this problem. To see this, consider separately the cases where $m = 1$

[10]In [30] it was shown that without loss of generality we can assume that $R_i\big(X_i(t), 0\big) = 0$.

and $m > 1$. For $m = 1$, the model and optimization problem are a generalization of the classical MAB problem described in Section 2.1.2. In RBs any machine that is available for continuation at t but is not chosen at t changes its state after t, so decisions made at time t are irrevocable. Consequently, for $m = 1$ forward induction does not result in an optimal scheduling policy. For $m > 1$, the model and problem are a generalization of the MAB with multiple plays described earlier in this section. Forward induction does not, in general, result in an optimal scheduling policy even for MABs with multiple plays, and therefore does not, in general, result in an optimal scheduling policy for RBs.

Nevertheless, there are cases where RBs have an optimal solution that is of the index type. We describe two such cases below.

Case 1. Consider the situation where all machines are identical and each machine is describe by a finite-state controlled Markov chain that is irreducible under any stationary Markov policy. That is, (6.46) simplifies to

$$X_i(t+1) = f_{i,t}\left(X_i(t), U_i(t), W_i(t)\right), \tag{6.47}$$

$i = 1, 2, \ldots, k$. Assume that the performance criterion is given by the infinite horizon average reward-per-unit-time-per-machine, that is

$$r^{\hat{\gamma}}(\alpha) = \frac{1}{k}\left[\lim_{T\to\infty}\frac{1}{T}\mathbb{E}^{\hat{\gamma}}\left[\sum_{t=1}^{T}\sum_{i=1}^{k} R_i\left(X_i(t-1), U_i(t)\right)\right]\right].$$

Such a performance criterion is not significantly different from the one considered so far, as infinite horizon expected discounted reward and infinite horizon expected reward-per-unit-time are related with one another [153, 251].

For the above model and problem assume that a subsidy Q is provided at time t to each machine that is not operated at t. Let $\nu(x_i)$ be the value of Q for which the expected reward-per-unit-time resulting from processing a machine (currently in state x_i) is equal to that resulting from not processing it plus the subsidy.

DEFINITION 6.3 *The value $\nu(x_i)$ is defined to be the index of a machine in state x_i.*

The notion of subsidy defined above can be used to introduce the concept of indexability that plays an important role in determining conditions sufficient to guarantee the optimality of an index-type solution for the above model.

DEFINITION 6.4 *Machine i is* indexable *if the following condition is satisfied.*

(C2) *Consider a value Q of the subsidy and suppose that when the machine is in state x, it is optimal not to operate it. Then, it is also optimal not to operate the machine in state x for any value of the subsidy higher than Q.*

A restless bandit is indexable if all its arms/machines are indexable.

The above notion of indexability may appear to be trivially true for any machine, but this it is not the case. In fact, indexability is a very strict requirement (see [252]). To proceed further we define:

1 an index policy $\hat{\gamma}$ according to which the machines with the m highest indices (specified by Definition 6.3) are operated at each time instant

2 for the index policy $\hat{\gamma}$

$$r^{\hat{\gamma}}(\alpha) := \lim_{\substack{k\to\infty \\ m\to\infty \\ \alpha=\frac{m}{k}}} \frac{1}{k}\left[\lim_{T\to\infty}\frac{1}{T}\mathbb{E}^{\hat{\gamma}}\left[\sum_{t=1}^{T}\sum_{i=1}^{k} R_i\left(X_i(t-1), U_i(t)\right)\right]\right] \tag{6.48}$$

Then the following result holds. If the RB process is indexable and certain technical conditions described in [247] hold,

$$r^{\hat{\gamma}}(\alpha) = \lim_{\substack{k\to\infty \\ m\to\infty \\ \alpha=\frac{m}{k}}} \frac{1}{k}\left[\sup_{\gamma\in\Gamma}\lim_{T\to\infty}\frac{1}{T}\mathbb{E}^{\gamma}\left[\sum_{t=1}^{T}\sum_{i=1}^{k} R_i\left(X_i(t-1), U_i(t)\right)\right]\right] \tag{6.49}$$

where Γ is the set of stationary Markov policies. That is, an optimal allocation strategy for the above class of RBs is an index policy. The above conditions are sufficient but not necessary to guarantee the optimality of an index policy $\hat{\gamma}$.

As pointed out above, indexability is a strict requirement and is often hard to check. These difficulties motivated the work in [30, 90, 183, 184]. In [30] Bertsimas and Niño-Mora provided a sufficient condition for indexability of a single restless bandit. In [183] Niño-Mora investigated an allocation policy for restless bandits and showed that if a set of conditions called Partial Conservation Laws (PCLs) are satisfied and the rewards associated with each machine belong to a certain "admissible region" (see [183]) then the allocation policy investigated in [183] is optimal. The ideas of [183] have been further refined in [184]. An approach to evaluating the sub-optimality of the index proposed

in [183] when the PCL conditions are satisfied but the rewards are not in the "admissible region" has been presented in [90]. An application of the results of [183] to queueing networks appears in [9].

Case 2. Consider k machines and one controller that has m processors available ($m < k$). Each machine $i, i = 1, \ldots, k$, is described by a controlled random process $\{X_i(t); t = 0, 1, 2, \ldots\}$ with a countable state-space $\{0, 1, 2, \ldots\}$ such that

$$X_i(t+1) = f_{i,t}\big(X_i(t), U_i(t), W_i(t)\big) \tag{6.50}$$

where $U_i(t) \in \{0, 1\}$ and $U_i(t) = 0$ (respectively, $U_i(t) = 1$) means that the machine is not processed (respectively, processed) at time t. For each $i, i = 1, \ldots, k$, $\{W_i(s), s = 1, 2, \ldots\}$ is a sequence of random variables that take values in $\{0, 1, \ldots, m_i\}$, are not necessarily independent and have known statistics. The sequences $\{W_i(s), s = 1, 2, \ldots\}$ and $\{W_j(s), s = 1, 2, \ldots\}$ are independent for all i, j, $i \neq j$. Furthermore, each sequence $\{W_i(s), s = 1, 2, \ldots\}$ is perfectly observed; that is, for each $\omega \in \Omega$ and i, $W_i(0, \omega)$, $W_i(1, \omega), \ldots, W_i(t-1, \omega)$ are known by the controller at time t, before the allocation decision at t is made. The functions $f_{i,t}(\cdot), i = 1, 2 \ldots, k$, are

$$f_{i,t}\big(X_i(t), U_i(t), W_i(t)\big) = \begin{cases} X_i(t) + W_i(t), & \text{if } U_i(t) = 0, \\ X_i(t) - \Lambda_i + W_i(t), & \text{if } X_i(t) \neq 0, U_i(t) = 1, \\ W_i(t), & \text{if } X_i(t) = 0, \end{cases} \tag{6.51}$$

where Λ_i is a random variable taking values in $\{0, 1\}$ with $\mathrm{P}[\Lambda_i = 0] = q_i > 0$ and $W_i(t)$ is a random variable that is not necessarily independent of $W_i(0), \ldots, W_i(t-1)$.

At each instant of time t a machine is either available for processing (it is "connected"), or it is not (it is "not connected"). The probability that machine i is connected at t is p_i, $i = 1, 2, \ldots, k$, for all t. The reward received at time t from a connected machine is

$$R_i\big(X_i(t), U_i(t)\big) = \begin{cases} Y_i, & \text{if } X_i(t) \neq 0 \text{ and } U_i(t) = 1, \\ 0, & \text{if } U_i(t) = 0 \text{ or } X_i(t-1) = 0, \end{cases} \tag{6.52}$$

where Y_i is a random variable taking values in $\{0, c_i\}$ with $\mathrm{P}(Y_i = c_i) = q_i$. The reward received at time t from a machine that is not connected is zero. The performance criterion is given by (6.44)[11]. The objective is to determine a scheduling/processor allocation policy to maximize the performance criterion.

[11] In [161], where the model of Case 2 is proposed, the performance criterion is given by a holding cost instead of a reward. Using the transformation in [235] we can convert this performance criterion into (6.44).

The model described above arises in single-hop mobile radio systems (see [161] and references therein). The same model has also independent interest as a specific problem in queueing theory.

Consider a machine i, $i = 1, 2, \ldots, k$, which at time t is in state $x_i \neq 0$ and is connected. The *Gittins index* $\nu_i(x_i)$ of machine i is

$$\nu_i(x_i) = q_i c_i.$$

Define the *Gittins index policy* to be the allocation policy γ_{GI} that operates at each time t the connected machines that are not in the zero state and have the m highest Gittins indices. The following condition (C3) describes an instance where the allocation policy γ_{GI} is optimal.

(C3) $c_1 q_1 > c_2 q_2 > \cdots > c_k q_k$, and in addition

$$c_i q_i \left[\frac{1-\beta}{1-(1-q_i)\beta} \right] \geq c_j q_j$$

for all $i, j, 1 \leq i < j \leq k$.

The proof of optimality of the Gittins index policy γ_{GI} under (C3) can be found in [161].

The essence of the results of [161] is the following: If we were guaranteed that the system described above operated away from the $\mathbf{0} := (0, 0, \ldots, 0)$ (k times) state then it would be optimal to allocate the m processors to the connected machines with the m highest Gittins indices. Near the state $\mathbf{0}$, processor utilization becomes a critical issue in determining an optimal allocation policy. The Gittins index policy may result in processor under-utilization; thus, it may not be optimal in some neighborhood of the state $\mathbf{0}$. Therefore, if we require optimality of the Gittins index policy for the problem under consideration, we must identify conditions to ensure that the advantage gained by always allocating the processors to the highest index machines overcompensates potential losses resulting from processor under-utilization near the $\mathbf{0}$ state. Such a condition is expressed by (C3) which requires that the indices associated with the machines should be sufficiently separated from one another. Such a separation results in a priority ordering of the machines sufficient to guarantee the optimality of the Gittins index policy.

Variations of the model of Case 2 were considered by Ehshan and Liu in [78, 79, 76, 77, 75]. In [78, 77] the authors investigate RBs with imperfect (delayed) observations and a single processor; they identify conditions sufficient to guarantee the optimality of an index policy. In [76] the authors consider identical machines and a linear symmetric holding cost criterion that can be

converted into (6.44) (see footnote 11). For this model they prove the optimality of the index rule. In [75, 79] the model is similar to that of [76] but the holding cost is convex. The authors identify conditions sufficient to guarantee the optimality of an index policy.

3.6 Discussion

Several of the extensions of the MAB problem presented in this chapter are related with one another. Specifically, the arm-acquiring bandit problem can be converted into a superprocess [240] and the MAB problem with switching cost can be converted into a RB problem [91].

Furthermore, there are other stochastic dynamic scheduling problems that are equivalent to the classical MAB problem. Two such problems are the tax problem [240, 253] and certain classes of Sensor Resource Management (SRM) problems [245].

In the tax problem there are k machines, each evolving according to (6.2) and (6.3). At each time instant exactly one machine is operated; the machines that are not operated remain frozen. If machine i is not operated at t a *tax* $T_i\big(X_i(t)\big)$, depending on the state $X_i(t)$ of machine i at t, is charged. The objective is to determine a scheduling/processor allocation policy γ to minimize

$$\mathbb{E}^{\gamma}\left[\sum_{t=0}^{\infty}\beta^t\sum_{i=1}^{k}T_i\big(X_i(t),U_i(t)\big)\,\middle|\,Z(0)\right],$$

where

$$Z(0):=[X_1(0),X_2(0),\ldots,X_k(0)],$$

$U(t):=\big(U_1(t),\ldots,U_k(t)\big)$, $t=0,1,2,\ldots$, is a random variable taking values in $\{e_1,\ldots,e_k\}$, and

$$T_i\big(X_i(t),U_i(t)\big)=\begin{cases}T_i\big(X_i(N_i(t))\big), & \text{if } U_i(t)=0,\\ 0, & \text{if } U_i(t)=1.\end{cases}$$

Even though feature (F4) of the MAB problem is not present in the tax problem, the two problems are equivalent. For the details of transforming the tax problem into a MAB problem, we refer the reader to [240]. An example of a tax problem is Klimov's problem in queueing networks [132, 133, 240, 31].

Sensor management problems that are equivalent to the MAB problem are presented and discussed in Chapter 7.

4. Example

In this section we present a simple search problem and show how it can be viewed as a classical MAB problem. We also present briefly variations of the problem which lead to different variants of the classical MAB problem. The search model that we consider is not rich enough to lead to an arm-acquiring bandit variation. We refer the reader to Chapter 7 for more realistic sensor management scenarios.

We consider a search problem in which a stationary target is hidden in one of k cells. The *a priori* probability that the target is in cell i is denoted by $p_i(0)$, $i = 1, 2, \ldots, k$. One sensor capable of operating in only one mode is available to search the target; the sensor can search only one cell at every instant of time. The sensor is imperfect in the sense that if the target is in cell i and the sensor looks at cell i, it does not necessarily find the target, i.e.,

$$\mathrm{P}(\text{sensor finds target in cell } i \mid \text{target is in cell } j) = \delta_{ij} q_j, \tag{6.53}$$

where δ_{ij} is the Kronecker delta function. The search is completed when the target is found. Successful completion of the search at time t gives a reward β^t, where $0 < \beta < 1$. Such a reward captures the fact that the target must be identified as quickly as possible. The objective is to determine a sequential sensor allocation policy γ that maximizes the expected reward.

We show how the problem described above can be formulated as a classical MAB problem. Associate each cell with one machine/bandit process. Let $p_i(t)$ be the posterior probability that the target is in location i at time t given all previous search locations and the event that the target has not been found. The probability $p_i(t)$ can be considered as the state of machine i at time t; let $p(t) := \big(p_1(t), p_2(t), \ldots, p_k(t)\big)$ be the state of all machines. Denote by $U(t) := \big(U_1(t), U_2(t), \ldots, U_k(t)\big)$ the sensor allocation at t. $U(t)$ is a random variable taking values in $\{e_1, e_2, \ldots, e_k\}$ (see Section 2.1). The expected reward corresponding to any sequential sensor allocation policy γ is $\mathbb{E}^\gamma\left[\beta^\tau \mathbb{1}(\text{target is found at } \tau)\right]$, where $\mathbb{1}(E)$ is the indicator function of event E and τ is the time when the search process stops. For any arbitrary but fixed sensor allocation policy γ this expected reward can be alternatively written as,

$$\begin{aligned}\mathbb{E}^\gamma\left[\beta^\tau \mathbb{1}(\text{target is found at } \tau)\right] &= \sum_{t=0}^{\infty} \beta^t \, \mathrm{P}^\gamma(\tau = t) \\ &= \sum_{t=0}^{\infty} \beta^t \Big[\sum_{i=1}^{k} \mathrm{P}^\gamma(\tau = t, U(t) = e_i) \Big]\end{aligned}$$

$$= \sum_{t=0}^{\infty} \beta^t \Big[\sum_{i=1}^{k} p_i(t) q_i \, \mathrm{P}^{\gamma} \big(U(t) = e_i \big) \Big]$$
$$= \sum_{t=0}^{\infty} \beta^t \Big[\sum_{i=1}^{k} R_i\big(p_i(t), u_i(t)\big) \Big],$$

where

$$R_i\big(p_i(t), u_i(t)\big) = \begin{cases} p_i(t) q_i, & \text{if } u_i(t) = 1, \\ 0, & \text{if } u_i(t) = 0, \end{cases}$$

and

$$u(t) = \big(u_1(t), u_2(t), \ldots, u_k(t)\big) = \gamma_t\big(p(t)\big).$$

By a careful examination of the above search problem, we find that features (F1), (F3), and (F4) of the MAB problem are present, but feature (F2) is not. This is because if we search location i at time t and do not find the target, then the state $p(t)$ evolves as follows:

$$p_i(t+1) = \frac{p_i(t)(1-q_i)}{c}, \tag{6.54}$$

$$p_j(t+1) = \frac{p_j(t)}{c}, \qquad j \neq i, \tag{6.55}$$

where $c = 1 - p_i(t) q_i$. Thus a particular allocation at t changes the state of all machines/cells.

The above problem can be converted into a classical MAB by considering an unnormalized probability $\hat{p}_i(t)$ as the state of machine i, $i = 1, 2, \ldots, k$ at time t and an appropriately modified reward function $\widehat{R}_i\big(\hat{p}_i(t), u_i(t)\big)$. Specifically the state $\hat{p}_i(t)$ of machine i, $i = 1, 2, \ldots, k$ evolves as follows:

$$\hat{p}_i(0), = p_i(0) \qquad \forall i, \tag{6.56}$$

$$\hat{p}_i(t+1) = \begin{cases} \hat{p}_i(t), & \text{if } u_i(t) = 0, \\ \hat{p}_i(t)(1-q_i), & \text{if } u_i(t) = 1, \end{cases} \tag{6.57}$$

and the modified reward function $\widehat{R}_i\big(\hat{p}_i(t), u_i(t)\big)$ is given by

$$\widehat{R}_i\big(\hat{p}_i(t), u_i(t)\big) = \begin{cases} \hat{p}_i(t) q_i, & \text{if } u_i(t) = 1, \\ 0, & \text{if } u_i(t) = 0. \end{cases} \tag{6.58}$$

The objective is to determine a sensor scheduling policy γ to maximize

$$\sum_{t=0}^{\infty} \beta^t \sum_{i=1}^{k} \widehat{R}_i\big(\hat{p}_i(t), u_i(t)\big),$$

where $u(t) = \gamma_t\big(\hat{p}(t)\big)$. By using $\hat{p}_i(t)$ as the state of machine[12] i at time t, the modified problem has features (F1)–(F4) and is thus a classical MAB.

Even though the modified problem has a different reward structure and different cell/machine dynamics from the original problem, any optimal policy for one problem is also optimal for the other (for details, see [68, Sec 14.14]). The Gittins index of every machine is always achieved at $\tau = 1$ and is given by

$$\nu_{X_i}\big(\hat{p}_i(t)\big) = \hat{p}_i(t)q_i$$

(see [68]).

We call the above described model the *basic model*. Changes in the number and types of sensors or the target dynamics result in problems that can be transformed into one of the variants of the MAB problem, and are described below.

When the sensor can operate in one of M modes and everything else is the same as in the basic model, the resulting problem can be formulated as a superprocess where the state of cell/machine i is the unnormalized probability $\hat{p}_i(t)$.

When there is a setup cost or setup delay for switching the sensor from one location to another and everything else is the same as in the basic model, the resulting problem can be formulated as a MAB problem with switching penalties.

If there are m sensors $(1 < m < k)$ and everything else is the same as in the basic model, the resulting problem can be formulated as a MAB problem with multiple plays.

Finally, if the target is moving, there are m sensors $(1 < m < k)$, each with one mode, and everything else is the same as in the basic model, the resulting problem can be formulated as a restless bandit.

By considering various combinations of changes in target motion and the number and types of available sensors, we can obtain generalizations of the bandit problems. Some such generalizations appear in Chapter 7.

[12] $\hat{p}(t)$ is an information state for the modified problem (see Chapter 2).

5. Chapter Summary

We have presented the formulation of the classical MAB problem, and discussed its key features and the ideas that lead to the Gittins index solution. We also presented different variants of the classical MAB problem, explained their key differences from Gittins's original formulation, and highlighted conditions under which a Gittins-type index rule is optimal. We emphasized the qualitative features of MAB problem and gave an intuitive explanation of the key results. The technical details are available in the literature we cite in this chapter. We illustrated how different variants of a single search problem can be transformed to the classical MAB problem and its variants. More realistic sensor management problems and their relation to the MAB problem are discussed in Chapter 7.

Chapter 12

APPENDICES

Alfred O. Hero, Aditya Mahajan, Demosthenis Teneketzis

University of Michigan, Ann Arbor, MI, USA

Edwin Chong

Colorado State University, Fort Collins, CO, USA

1. Information Theory

Given the option of making one of several different types of measurements, e.g., $Y_1, \ldots, Y_m$, one would generally prefer making the measurement that leads to maximal uncertainty reduction or, equivalently, maximal information gain about a signal of interest S. This is one of the prime motivations behind information theoretic sensor management since information theory provides a way to systematically quantify uncertainty and information. In this appendix we present those elements of information theory pertinent to sensor management. We will limit our coverage to Shannon's definitions of entropy and conditional entropy, the data processing theorem and mutual information, and information divergence.

1.1 Entropy and Conditional Entropy

Let Y be a measurement and S be a quantity of interest, e.g. the position of a target or the target id. We assume that Y and S are random variables or random vectors with joint distribution $p_{Y,S}(y, s)$ and marginal distributions p_Y and p_S, respectively. The entropy of S, denoted $\mathcal{H}(S)$, quantifies uncertainty in the value of S before any measurement is made, called the prior uncertainty in S. High values of $\mathcal{H}(S)$ imply high uncertainty about the value of S. The

of a continuous random variable with density p_S is defined as

$$\mathcal{H}(S) = -\int p_S(s) \log p_S(s) ds, \tag{12.1}$$

where p_S denotes the probability density of S. If $S = \mathbf{S}$ is a continuous random vector the definition of is similar except that the expression on the right involves a multidimensional integral over each component of the vector valued $\mathbf{s}$. For a discrete random variable the Shannon entropy is

$$\mathcal{H}(S) = -\sum_{s \in \mathcal{S}} p_S(s) \log p_S(s),$$

where p_S is now a probability mass function and $\mathcal{S}$ is its support set, i.e., the discrete set of values s for which $p_S(s) > 0$.

Oftentimes one is interested in the entropy of a random variable S conditioned on another random variable Y. For example, the amount by which an observation of Y reduces the entropy of S indicates the value of this observation in predicting S. There are two possible ways of defining such an entropy quantity: the entropy of the conditional distribution $p_{S|Y}$ of S given Y, which is a function of Y, and the of S given Y.

The Shannon entropy of the conditional distribution of S given Y, also called the point conditioned Shannon entropy, is denoted $\mathcal{H}(S|Y = y)$ and is defined as follows. We assume for simplicity that, given $Y = y$, S is a conditionally continuous random variable with conditional (posterior) density $p_{S|Y}(s|y)$ and define the entropy of this conditional density as

$$\mathcal{H}(S|Y = y) \stackrel{\text{def}}{=} -\int p_{S|Y}(s|y) \log p_{S|Y}(s|y) ds.$$

The point conditioned entropy is a function of y. It becomes a random variable when y is replaced by the random variable Y.

The conditional Shannon entropy $\mathcal{H}(S|Y)$ of S given Y is defined as the Shannon entropy of the conditional distribution $p_{S|Y}$. This conditional entropy can be interpreted as the uncertainty in S after the measurement Y is made, called the posterior uncertainty. When S and Y are continuous random variables with joint density $p_{S,Y}$ and conditional (posterior) density $p_{S|Y}$

$$\mathcal{H}(S|Y) = -\int dy\, p_Y(y) \int ds\, p_{S|Y}(s|y) \log p_{S|Y}(s|y).$$

The conditional entropies $\mathcal{H}(S|Y)$ and $\mathcal{H}(S|Y = y)$ are related

$$\mathcal{H}(S|Y) = \int \mathcal{H}(S|Y = y) p_Y(y) dy.$$

When S and Y are discrete random variables an analogous expression holds for $\mathcal{H}(S|Y)$ with conditional and marginal densities replaced by conditional and marginal probability mass functions and integrals replaced by summations. A special "mixed discrete-continuous" case that frequently arises in target tracking problems is: S is a continuously evolving random vector, e.g., a target state vector, while Y is a discrete random vector, e.g., the binary output of the signal detector of a radar receiver. In this case the conditional entropy is

$$\mathcal{H}(S|Y) = -\sum_{y\in\mathcal{Y}} p_Y(y) \int p_{S|Y}(s|y) \log p_{S|Y}(s|y) ds \ ,$$

where $\mathcal{Y}$ is a discrete set containing all possible measurement values, p_Y is the probability mass function for Y, and $p_{S|Y}(s|y)$ is the (assumed continuous) posterior density of S given Y.

There are subtle but important differences between entropy for discrete vs continuous random variables. For discrete S the entropy is always non-negative, while for continuous S the entropy can be negative. For discrete random variables the entropy is directly related to the maximal attainable compression-rate without loss of information about S.

1.2 Information Divergence

Let p and q be two candidate probability densities of a real random variable S. The Kullback-Liebler (KL) divergence between p and q is defined as [152]

$$\mathrm{KL}(p\|q) = \int p(s) \log \frac{p(s)}{q(s)} ds.$$

The KL divergence is not symmetric in p and q and is thus is not true measure of distance between densities. However, it does behave like a similarity measure, sometimes called a pseudo-distance, in that it is concave(convex) in p (q), it is non-negative, and it is equal to zero when $p = q$.

1.3 Shannon's Data Processing Theorem

The average reduction in uncertainty about S due to observing Y can be quantified by the difference:

$$\Delta\mathcal{H}(S|Y) = \mathcal{H}(S) - \mathcal{H}(S|Y).$$

The data processing theorem asserts that this difference is always non-negative regardless of whether S is continuous or discrete. This theorem is easily proven

by invoking convexity of the log function and the Jensen inequality [64] and mathematically captures the obvious: observations are never harmful in that they can never increase uncertainty about a signal.

1.4 Shannon Mutual Information

The difference $\Delta\mathcal{H}(S|Y)$ is better known as the Shannon mutual information , denoted $I(S;Y)$, between S and Y. The more reduction there is in uncertainty the higher is the mutual information. An equivalent expression for Shannon mutual information that applies to continuous random variables is

$$I(S;Y) = \int dy \int ds\, p_{S,Y}(s,y) \log \frac{p_{S,Y}(s,y)}{p_S(s)p_Y(y)}.$$

An analogous expression applies to discrete random variables. Shannon's mutual information can be recognized as the Kullback-Liebler (KL) divergence between $p_{S,Y}$ and $p_S p_Y$ and can be interpreted as a measure of closeness to independence of the joint density of S and Y.

The obvious symmetry of the mutual information expression in the random variables S and Y implies that

$$I(S;Y) = \mathcal{H}(S) - \mathcal{H}(S|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|S).$$

This relation is often used in the implementation of mutual information driven strategies of sensor management since the quantities on the right hand side of the equality are usually more easily computed than those on the left hand side.

1.5 Further Reading

Information theory is a mature subject and there are many good sources for the beginner. One of the most popular textbooks used in introductory graduate courses on information theory is the textbook by Cover and Thomas [64] that is accessible to electrical engineers. The book by MacKay [164] covers the topic from the unique perspective of machine learning and contains many interesting applications. The classic book by Kullback [152] is a treatment of information theory that is firmly motivated by mathematical statistics. More mathematically advanced treatments of the subject are the books by Csiszár and Korner [67] and Yeung [257].

2. Markov Processes

Our ability to make effective sensor-management decisions is based fundamentally on our access to models. In particular, nonmyopic decision making relies on modeling the random processes that represent uncertainty in the system, such as target motion in tracking problems. In this section, we review a framework for uncertainty modeling based on Markov processes. The material discussed here provides the necessary background for understanding the methods discussed throughout this book, including Kalman filtering and partially observable Markov decision processes (POMDPs). We also provide some pointers to sources for further study.

2.1 Definition of Markov Process

A Markov process is a stochastic process satisfying a particular property called the Markov property, which we will define precisely below. Here, we consider only discrete-time Markov processes, and use $k = 0, 1, \ldots$ as the time index. (So a stochastic process here is no different from a sequence of random variables.)

We use $\mathcal{X}$ to denote the *state space* of the process, which is the set of values that the process can take at each time step. We also assume that associated with $\mathcal{X}$ is a collection of subsets $\mathcal{F}$ forming a σ-algebra. In the case where $\mathcal{X}$ is countable (discrete), we take $\mathcal{F}$ to be the power set of $\mathcal{X}$. If $\mathcal{X}$ is a Euclidean space, we take $\mathcal{F}$ to be the Borel σ-algebra. Throughout this appendix, we will refer to these two special cases simply by the terms *discrete* and *continuous*.

A stochastic process $X_0, X_1, \ldots$ is a *Markov process* (also called a *Markov chain*) if for each $k = 1, 2, \ldots$ and $E \in \mathcal{F}$,

$$\mathrm{P}(X_{k+1} \in E | X_k, \ldots, X_0) = \mathrm{P}(X_{k+1} \in E | X_k).$$

We call X_k the *state* of the Markov process at time k.

The condition above is called the *Markov property*, which boils down to this: the conditional distribution of X_{k+1} given the entire history up to time k depends only on X_k. In other words, the future of the process is conditionally independent of the past, given the present. To put it a different way, the "memory" in the process lasts only one time step.

The Markov property is in fact not as stringent a requirement as it may first appear to be. Indeed, suppose we are given a stochastic process that fails to satisfy the Markov property, but instead satisfies, for each $k = 1, 2, \ldots$ and

$E \in \mathcal{F}$,

$$\mathrm{P}(X_{k+1} \in E | X_k, \ldots, X_0) = \mathrm{P}(X_{k+1} \in E | X_k, X_{k-1});$$

in other words, the memory in the process is two instead of one. Then, it is easy to see that this process gives rise to a Markov process $\{Y_k\}$ by defining $Y_k = (X_k, X_{k-1})$. Indeed, using this construction, any stochastic process with memory lasting only a bounded time into the past gives rise to a Markov process. It turns out that many scenarios in practice can be modeled as Markov processes provided we define the state spaces appropriately.

2.2 State-transition Probability

In the discrete case, the Markov property can be expressed more simply as follows: for each $k = 1, 2, \ldots$ and $i, j, i_0, \ldots, i_{k-1} \in \mathcal{X}$,

$$\mathrm{P}(X_{k+1} = j | X_k = i, \ldots, X_0 = i_0) = \mathrm{P}(X_{k+1} = j | X_k = i).$$

It is clear that for a Markov process, once we specify $\mathrm{P}(X_0 = i)$ and $\mathrm{P}(X_{k+1} = j | X_k = i)$ for each $k = 0, 1, \ldots$ and $i, j \in \mathcal{X}$, the probability law of the process is completely specified. In many problems of interest, the conditional probabilities $\mathrm{P}(X_{k+1} = j | X_k = i)$ do not depend on k. In this case, we say that the Markov process is *time-homogeneous* (or simply *homogeneous*).

For a homogeneous Markov process with discrete state space, write $p_{ij} = \mathrm{P}(X_{k+1} = j | X_k = i)$, $i, j \in \mathcal{X}$. We call this set of probabilities the *state-transition law* (or simply the *transition law*) of the Markov process. Each p_{ij} is called a *state-transition probability* (or simply a *transition probability*). It is often convenient to represent the transition law using a graph, where the nodes are the states and the arcs are labeled with transition probabilities. In the case where $\mathcal{X}$ is finite (whence we can write, without loss of generality, $\mathcal{X} = \{1, 2, \ldots, N\}$), the transition law can also be represented by a square matrix $[p_{ij}]$, called the *state-transition matrix* (or *transition matrix*). Note that any transition matrix has the property that each entry is nonnegative and each row sums to one. Any matrix satisfying this property is called a *stochastic matrix*, and is in fact the transition matrix of some Markov chain.

In the continuous case, we assume that the state-transition law can be written in terms of conditional densities $p_{X_{k+1}|X_k}(x_{k+1}|x_k)$, $x_k, x_{k+1} \in \mathcal{X}$. (The reader may assume for simplicity that $\mathcal{X}$ is the real line or a subset of it; the case of multidimensional Euclidean spaces involves treating all densities as multivariable functions.) We also assume that X_0 has a density p_{X_0}. If $p_{X_{k+1}|X_k}$ does not depend on k, then we say that the Markov process is time-homogeneous. In the remainder of this discussion, we consider only time-

homogeneous Markov processes. Also, for simplicity, we will drop the subscripts in the notation for the conditional densities.

2.3 Chapman-Kolmogorov Equation

Consider a discrete-state Markov process. Given $n = 0, 1, \ldots$, we define the n-step transition law by $p_{ij}^{(n)} = \mathrm{P}(X_n = j | X_0 = i)$, $i, j \in \mathcal{X}$. The n-step transition law satisfies the *Chapman-Kolmogorov equation*:

$$p_{ij}^{(n+m)} = \sum_{k \in \mathcal{X}} p_{ik}^{(n)} p_{kj}^{(m)}, \qquad i, j \in \mathcal{X}.$$

In the case of a finite state space, the Chapman-Kolmogorov equation has a natural interpretation in terms of the transition matrix: the n-step transition law is given by the nth power of the transition matrix.

In the continuous case, we can similarly define the n-step transition law in terms of the conditional density $f^{(n)}(x_n|x_0)$, $x_n, x_0 \in \mathcal{X}$. The Chapman-Kolmogorov equation then takes the form

$$f^{(n+m)}(x_{n+m}|x_0) = \int_{\mathcal{X}} f^{(n)}(x_n|x_0) f^{(m)}(x_{n+m}|x_n)\, dx_n, \quad x_{n+m}, x_0 \in \mathcal{X}.$$

2.4 Markov reward processes

Given a Markov process, suppose we associate with each state $x \in \mathcal{X}$ a real number $R(x)$, called a *reward*. A Markov process so endowed with a reward function is called a *Markov reward process*. We define the *mean total reward* over a horizon H as

$$\mathbb{E}\left[\sum_{k=0}^{H-1} R(X_k)\right].$$

Many problems in practice can be modeled as Markov reward processes—in such a model, the mean total reward represents some quantity of interest (such as the value of a performance metric).

It is often the case that the horizon H is very large. In such cases, for technical reasons relevant to the analysis of Markov processes, the objective function is often expressed as a limit (i.e., with an infinite horizon). A sensible limiting objective function is the *infinite horizon* (or *long-term*) *average* reward:

$$\lim_{H \to \infty} \mathbb{E}\left[\frac{1}{H} \sum_{k=0}^{H-1} R(X_k)\right].$$

Another common limiting objective function is the *infinite horizon discounted* reward:

$$\lim_{H\to\infty} \mathbb{E}\left[\sum_{k=0}^{H-1} \beta^k R(X_k)\right],$$

where β is a number between 0 and 1 called the *discount factor*. .

2.5 Partially Observable Markov Processes

Given a Markov process and a set $\mathcal{Y}$ (with a σ-algebra $\mathcal{F}$ of subsets of $\mathcal{Y}$), suppose we associate with each state $x \in \mathcal{X}$ a conditional distribution $\mathrm{P}(Y \in E|x)$, $E \in \mathcal{F}$. We call $\mathcal{Y}$ the *observation space*, and the conditional distribution $\mathrm{P}(Y \in E|x)$, $E \in \mathcal{F}$, the *observation law*. A Markov process so endowed with an observation law is called a *partially observable Markov processes* or a *hidden Markov model*.

The reason we use the terms "partially observable" and "hidden" is that we think of $\mathcal{Y}$ as the set of observations we have of the "underlying" Markov process, but we cannot directly observe the underlying process. If $X_0, X_1, \ldots$ represents the underlying Markov process, then all we can observe is the random sequence $Y_0, Y_1, \ldots$, where each Y_k has conditional distribution given X_k specified by the observation law. The sequence $Y_0, Y_1, \ldots$ is assumed to be conditionally independent given $X_0, X_1, \ldots$. Many practical processes, especially in sensing applications, are well modeled by hidden Markov models. For example, X_k may represent the location of a target at time k, and Y_k may be a radar measurement of that location. The transition law in this case represents the motion of the target, and the observation law represents the relationship between the target location and the radar measurement.

Even though we cannot directly access X_k, the observations provide us with some information on X_k. In fact, at each k, we can compute the *a posteriori* (or posterior) distribution of X_k given the history of observations $\mathcal{I}_k = \{Y_0, \ldots, Y_{k-1}\}$. We call this posterior distribution the *belief state* or *information state* at time k, and here it is denoted π_k. The sequence of belief states satisfies the Markov property, and is therefore a legitimate Markov process, albeit with a rather unwieldy state space—the set of all distributions on $\mathcal{X}$. It turns out that given the belief state at time k and the observation Y_k, we can calculate the belief state at time $k+1$ using a simple update procedure, as we will show below.

For the case of a discrete Markov process with discrete observation space, suppose the transition law is given by p_{ij}, $i, j \in \mathcal{X}$. Suppose $y_0, y_1, \ldots$ are the observations. Let π_k represent the belief state at time k, which is a conditional

probability mass function:

$$\pi_k(i) = \mathrm{P}(X_k = i | Y_0 = y_0, \ldots, Y_{k-1} = y_{k-1}), \quad i \in \mathcal{X}.$$

Also define the "updated belief state" taking into account the observation y_k:

$$\hat{\pi}_k(i) = \mathrm{P}(X_k = i | Y_0 = y_0, \ldots, Y_k = y_k), \quad i \in \mathcal{X}.$$

Then π_{k+1} can be derived from π_k and Y_k using the following two-step procedure:

1. Calculate the "updated belief state" $\hat{\pi}_k$ (taking into account the observation y_k) using Bayes' rule:

$$\hat{\pi}_k(j) = \frac{\mathrm{P}(Y_k = y_k | X_k = j)\pi_k(j)}{\sum_{\ell \in \mathcal{X}} \mathrm{P}(Y_k = y_k | X_k = \ell)\pi_k(\ell)}, \qquad j \in \mathcal{X}.$$

2. Calculate the belief state π_{k+1} based on $\hat{\pi}_k$ and the transition law:

$$\pi_{k+1}(j) = \sum_{i \in \mathcal{X}} \hat{\pi}_k(i) p_{ij}, \qquad j \in \mathcal{X}.$$

By reversing the order of 1. and 2. one obtains an equivalent algorithm for updating $\hat{\pi}_k$ to $\hat{\pi}_{k+1}$.

For the case of a continuous Markov process with continuous observation space (real numbers), suppose the transition law is given by the conditional density $p(x_{k+1}|x_k)$, and the observation law is given by the conditional density $q(y_k|x_k)$. The belief state at time k is then represented by a density function π_k. The two-step update procedure to calculate π_{k+1} based on π_k and y_k is given analogously as follows:

1. Calculate the "updated belief state" taking into account the observation y_k, using Bayes' rule:

$$\hat{\pi}_k(x_k) = \frac{q(y_k|x_k)\pi_k(x_k)}{\int_{\mathcal{X}} q(y_k|x)\pi_k(x)\,dx}, \qquad x_k \in \mathcal{X}.$$

2. Calculate the belief state π_{k+1} based on $\hat{\pi}_k$ and the transition law:

$$\pi_{k+1}(x_{k+1}) = \int_{\mathcal{X}} \hat{\pi}_k(x_k) p(x_{k+1}|x_k)\,dx_k, \qquad x_{k+1} \in \mathcal{X}.$$

If the transition and observation laws both arise from linear equations, and the initial density p_{X_0} is Gaussian, then the belief states remain Gaussian over

time. In this case, the above update procedure can be reduced to a procedure to update just the mean and variance (or covariance in the multidimensional case) of the belief state. This procedure is called the *Kalman filter*.

If we augment the definition of a partially observable Markov process with control actions, then we obtain a partially observable Markov *decision* process (POMDP), as defined in Chapter 2. The two-step update procedure for belief states remains valid provided we include the action into the observation and transition laws.

2.6 Further Reading

Our discussion here assumes a basic understanding of probability and stochastic processes. An excellent recent book that provides this background and that also includes a chapter on Markov chains is by Gubner [97]. Many books focusing on Markov processes exist. A small but useful book by Ross [197] remains an accessible classic. The book by Çinlar [53] provides an excellent in-depth treatment of discrete state space Markov processes. Meyn and Tweedie [170] treat continuous (and other even more general) state space Markov processes—their treatment necessarily involves heavy mathematical machinery.

3. Stopping Times

We briefly present the concept of stopping time which plays an important role in the solution of the MAB problem discussed in Chapter 6. We proceed as follows: We first present all relevant definitions in Section 3.1. We give an example of a stopping time in Section 3.2. Finally, we characterize the stopping times that achieve the Gittins index in the classical MAB problem in Section 3.3 and suggest some further reading material for the advanced reader in Section 3.4.

3.1 Definitions

DEFINITION 12.1 (PROBABILITY SPACE) *A probability space $(\Omega, \mathcal{F}, P)$ consists of a sample space Ω, a σ-field (σ-algebra) $\mathcal{F}$ of the subsets of Ω, and a probability measure P on the elements of $\mathcal{F}$.*

DEFINITION 12.2 (σ-FIELD GENERATED BY A RANDOM VARIABLE) *Let $X : (\Omega, \mathcal{F}, P) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}), \hat{P})$ be a random variable. Denote by $\sigma(X)$ the*

smallest σ-field with respect to which X is measurable. Then

$$\sigma(X) = \{A \in \mathcal{F} : \exists B \in \mathcal{B}(\mathbb{R}), X^{-1}(B) = A\}.$$

The σ-field $\sigma(X)$ represents the "information" obtained about the experiment described by $(\Omega, \mathcal{F}, P)$ after observing X. This can be explained as follows. First consider a probability space $(\Omega, \mathcal{F}, P)$ which represents a random experiment. An event $E \in \mathcal{F}$ can be thought of as a "yes-no question" that can be answered after we observe the outcome of the experiment. Then $\sigma(X)$ is the collection of all "yes-no questions" that can be answered after observing X.

DEFINITION 12.3 (INCREASING FAMILY OF σ-FIELDS) *A family $\{\mathcal{F}, \mathcal{F}_t; t = 0, 1, 2, \dots\}$ of σ-fields is called* increasing *if $\mathcal{F}_t \subset \mathcal{F}_{t+1} \subset \mathcal{F}$ for all $t = 0, 1, 2, \dots$.*

$\mathcal{F}_t$ represents the information about the evolution of a system that is available to an observer/decision-maker at time t, $t = 0, 1, 2, \dots$. When the observer has perfect recall, (that is, it remembers everything that it has seen and everything that it has done in the past) then $\mathcal{F}_t \subset \mathcal{F}_{t+1}$, $\forall t$ and $\{\mathcal{F}, \mathcal{F}_t; t = 0, 1, 2, \dots\}$ is an *increasing family* of σ-fields.

DEFINITION 12.4 (STOPPING TIME) *Let $\bar{N} := \{0, 1, 2, \dots, +\infty\}$. A random variable $\tau : (\Omega, \mathcal{F}, P) \to (\bar{N}, 2^{\bar{N}}, \hat{P})$ is a* stopping time *with respect to the increasing family of σ-fields $\{\mathcal{F}, \mathcal{F}_t; t = 0, 1, 2, \dots\}$ if the event $\{\tau = t\} := \{\omega : \tau(\omega) = t\} \in \mathcal{F}_t$ for all $t = 0, 1, 2, \dots$.*

Any constant random variable equal to a non-negative integer or $+\infty$ is a stopping time. A stopping time can be thought of as the time when a given random event E happens, with the convention that it takes the value $+\infty$ if E never happens. Alternatively, τ can be thought of as the time when a gambler playing a game decides to quit. Whether or not he quits at time t depends only on the information up to and including time t; so $\{\tau = t\} \in \mathcal{F}_t$.

3.2 Example

Let $\{X_t; t = 0, 1, 2, \dots\}$ be a time-homogeneous finite-state Markov chain defined on $(\Omega, \mathcal{F}, P)$ with state space S, and matrix of transition probabilities $\{Q_{ij}; i, j \in S\}$. Assume that the evolution of the Markov chain is perfectly observed by an observer that has perfect recall. The observer has perfect recall, its information $\mathcal{F}_t$ at time t is given by $\sigma(X_0, X_1, \dots, X_t)$, since $\sigma(X_0, X_1, \dots, X_t)$ represents all the "yes-no questions" about events in $\mathcal{F}$

that the observer can answer after observing $X_0, X_1, \ldots, X_t$. Furthermore, $\{\mathcal{F}, \mathcal{F}_t; t = 0, 1, 2, \ldots\}$ is an increasing family of σ-fields. Consider a non-empty subset A of the state of space, that is, $A \subset S$, $A \neq \emptyset$. Define for all $\omega \in \Omega$,

$$\tau_A(\omega) := \min\{t : X_t(\omega) \in A\}.$$

The random variable τ_A defines the first instant of time the Markov chain enters set A, and is called the *hitting time* of A. It is a stopping time with respect to the family of σ-fields $\{\mathcal{F}, \mathcal{F}_t; t = 0, 1, 2, \ldots\}$.

3.3 Stopping Times for Multi-armed Bandit Problems

Consider the classical MAB problem of Chapter 6 with finite-state Markovian machines. The Gittins index of machine i, $i = 1, \ldots, k$ in this case is given by Eq (6.13). The stopping time that maximizes the RHS of (6.13) is the hitting time of an appropriate subset of the state space $\{1, 2, \ldots, \Delta_i\}$ of machine i. This set is the stopping set $S_i\big(x_i(N_i(\tau_l))\big)$ determined in Section 2.4.

In the case of non-Markovian machines the Gittins index of machine i, $i = 1, \ldots, k$ is given the (6.8). The stopping time that maximizes the RHS of (6.8) can be described as follows: Let $\mathbf{x}_i^{\tau_l} := \big(x_i(0), \ldots, x_i(N_i(\tau_l))\big)$. Define an appropriate family $\mathcal{S}(\tau_l)$ as $\{S_i^{N_i(\tau_l)+r}; r = 1, 2, 3, \ldots\}$ where,

$$S_i^{N_i(\tau_l+r)}(\mathbf{x}_i^{\tau_l}) \subset \mathbb{R}^{N_i(\tau_l)+r}, \quad r = 1, 2, \ldots$$

Let

$$\hat{\tau}_{l+1}(\mathcal{S}(\tau_l)) = \min\{t > \tau_l : \mathbf{x}_i^t \in S_i^{N_i(t)}; N_i(t) = N_i(\tau_l) + t - \tau_l + 1\}$$

Define $\mathcal{S}^*(\tau_l)$ by

$$\mathcal{S}^*(\tau_l) = \arg\max_{\mathcal{S}(\tau_l)} \frac{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\hat{\tau}_{l+1}\left(S(\tau_1)\right)-1} \beta^t R_i\big(X_i\big(N_i(\tau_l) + t - \tau_l(\omega)\big)\big) | \mathbf{x}_i^{\tau_l}\right]}{\mathbb{E}\left[\sum_{t=\tau_l(\omega)}^{\hat{\tau}_{l+1}\left(S(\tau_1)\right)-1} \beta^t | \mathbf{x}_i^{\tau_l}\right]}. \tag{12.2}$$

Then $\tau_{l+1} = \hat{\tau}_{l+1}(\mathcal{S}^*(\tau_l))$. In general, for non-Markovian machines, maximizing RHS of (12.2) over all choices of $\mathcal{S}(\tau_l)$ is difficult, and computing the index is non-trivial.

3.4 Further Reading

Even though the notion of stopping times is intuitively simple, its formal treatment tends to be at an advanced level. Most graduate-level textbooks on probability theory contain a treatment of stopping times. See, for example, Billingsley [32], Shireyaev [210], and Jacod and Protter [117]. Stopping times is a fundamental concept in martingale theory and in optimal stopping problems. Reference books on these topics contain a more exhaustive treatment of stopping times. We refer the reader to Dellacherie and Meyer [69] for a treatment of stopping times in the context of martingales, and to Chow, Robins, and Siegmund [59], and Shireyaev [209] for a treatment of stopping times in the context of optimal stopping problems.

References

[1] R. Agrawal, M. V. Hegde, and D. Teneketzis. Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching cost. *IEEE Transactions on Automatic Control*, 33:899–906, 1988.

[2] R. Agrawal, M. V. Hegde, and D. Teneketzis. Multi-armed bandits with multiple plays and switching cost. *Stochastics and Stochastic Reports*, 29:437–459, 1990.

[3] R. Agrawal and D. Teneketzis. Certainty equivalence control with forcing: revisited. *Systems and Control Letters*, 13:405–412, 1989.

[4] R. Agrawal, D. Teneketzis, and V. Anantharam. Asymptotically efficient adaptive allocation schemes for controlled Markov chains: finite parameter space. *IEEE Transactions on Automatic Control*, 34:1249–1259, 1989.

[5] R. Agrawal, D. Teneketzis, and V. Anantharam. Asymptotically efficient adaptive control schemes for controlled I.I.D. processes: finite parameter space. *IEEE Transactions on Automatic Control*, 34:258–267, 1989.

[6] S.-I. Amari. *Methods of Information Geometry*. American Mathematical Society - Oxford University Press, Providence, RI, 2000.

[7] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays — part I: I.I.D. rewards. *IEEE Transactions on Automatic Control*, 32:968–976, 1987.

[8] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays — part II: Markovian rewards. *IEEE Transactions on Automatic Control*, 32:977–982, 1987.

[9] P. S. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O'Keefe. Whittle's index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57:21–39, 2003.

[10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[11] M. Asawa and D. Teneketzis. Multi-armed bandits with switching penalties. *IEEE Transactions on Automatic Control*, 41:328–348, 1996.

[12] J. Banks and R. Sundaram. Switching costs and the Gittins index. *Econometrica*, 62:687–694, 1994.

[13] Y. Bar-Shalom. *Multitarget Multisensor Tracking: Advanced Applications*. Artech House, Boston, MA, 1990.

[14] Y. Bar-Shalom and W. D. Blair. *Multitarget-Multisensor Tracking: Applications and Advances, Volume III*. Artech House, Boston, MA, 2000.

[15] A. R. Barron. Complexity regularization with application to artificial neural networks. In *Nonparametric Functional Estimation and Related Topics*, pages 561–576. Kluwer Academic Publishers, 1991.

[16] A. G. Barto, W. Powell, and J. Si, editors. *Learning and Approximate Dynamic Programming*. IEEE Press, New York, NY, 2004.

[17] M. Beckmann. *Dynamic Programming of Economic Decisions*. Springer-Verlag, New York, NY, 1968.

[18] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38:716–719, 1952.

[19] R. Bellman. A problem in the sequential design of experiments. *Sankhia*, 16:221–229, 1956.

[20] R. Bellman. *Adaptive Control Processes: a Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[21] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, NJ, 1962.

[22] D. A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall, 1985.

[23] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 1995.

[24] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 1995.

[25] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vols. I-II*. Athena Scientific, Belmont, MA, 3rd edition, 2005.

[26] D. P. Bertsekas and D. A. Castañón. Rollout algorithms for stochastic scheduling. *Heuristics*, 5:89–108, 1999.

[27] D. P. Bertsekas and S. E. Shreve. *Stochastic Optimal Control: The Discrete Time Case*, volume 1. Academic Press, 1978.

[28] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

[29] D. Bertsimas and J. Niño-Mora. Conservation laws, extended polymatroids and multiarmed bandit problems; a polyhedral approach to indexable systems. *Mathematics of Operations Research*, 21:257–306, 1996.

[30] D. Bertsimas and J. Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48:80–90, 2000.

[31] D. Bertsimas, I. C. Paschalidis, and J. N. Tsitsiklis. Branching bandits and Klimov's problem: achievable region and side constraints. *IEEE Transactions on Automatic Control*, 40:2063–2075, 1995.

[32] P. Billingsley. *Probability and Measure*. John Wiley and Sons, New York, NY, 1995.

[33] S. S. Blackman. *Multiple-Target Tracking with Radar Applications*. Artech House, Boston, MA, 1986.

[34] D. Blackwell. Discrete dynamic programming. *Annals of Mathematical Statistics*, 33:719–726, 1962.

[35] D. Blackwell. Discounted dynamic programming. *Annals of Mathematical Statistics*, 36:226–235, 1965.

[36] W. D. Blair and M. Brandt-Pearce. Unresolved Rayleigh target detection using monopulse measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 34:543–552, 1998.

[37] G. Blanchard and D. Geman. Hierarchical testing designs for pattern recognition. *Annals of Statistics*, 33(3):1155–1202, 2005.

[38] D. Blatt and A. O. Hero. From weighted classification to policy search. In *Neural Information Processing Symposium*, volume 18, pages 139–146, 2005.

[39] D. Blatt and A. O. Hero. Optimal sensor scheduling via classification reduction of policy search (CROPS). In *International Conference on Automated Planning and Scheduling*, 2006.

[40] H. A. P. Blom and E. A. Bloem. Joint IMMPDA particle filter. In *International Conference on Information Fusion*, 2003.

[41] A. G. B. S. J. Bradtke and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72:81–138, 1995.

[42] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1983.

[43] M. V. Burnashev and K. S. Zigangirov. An interval estimation problem for controlled observations. *Problems in Information Transmission*, 10:223–231, 1974. Translated from *Problemy Peredachi Informatsii*, 10(3):51–61, July-September, 1974.

[44] L. Carin, H. Yu, Y. Dalichaouch, A. R. Perry, P. V. Czipott, and C. E. Baum. On the wideband EMI response of a rotationally symmetric permeable and conducting target. *IEEE Transactions on Geoscience and Remote Sensing*, 39:1206–1213, June 2001.

[45] A. R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Department of Computer Science, Brown University, 1998.

[46] A. R. Cassandra, M. L. Littman, and L. P. Kaelbling. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Uncertainty in Artificial Intelligence*, 1997.

[47] D. A. Castañón. Approximate dynamic programming for sensor management. In *IEEE Conference on Decision and Control*, pages 1202–1207. IEEE, 1997.

[48] D. A. Castañón. A lower bound on adaptive sensor management performance for classification. In *IEEE Conference on Decision and Control*. IEEE, 2005.

[49] D. A. Castañón and J. M. Wohletz. Model predictive control for dynamic unreliable resource allocation. In *IEEE Conference on Decision and Control*, volume 4, pages 3754–3759. IEEE, 2002.

[50] R. Castro, R. Willett, and R. Nowak. Coarse-to-fine manifold learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, May, Montreal, Canada, 2004.

[51] R. Castro, R. Willett, and R. Nowak. Faster rates in regression via active learning. In *Neural Information Processing Systems*, 2005.

[52] R. Castro, R. Willett, and R. Nowak. Faster rates in regression via active learning. Technical report, University of Wisconsin, Madison, October 2005. ECE-05-3 Technical Report.

[53] E. Çinlar. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ, 1975.

[54] H. S. Chang, R. L. Givan, and E. K. P. Chong. Parallel rollout for online solution of partially observable Markov decision processes. *Discrete Event Dynamic Systems*, 14:309–341, 2004.

[55] H. Chernoff. Sequential design of experiments. *Annals of Mathematical Statistics*, 30:755–770, 1959.

[56] H. Chernoff. *Sequential Analysis and Optimal Design*. SIAM, 1972.

[57] A. Chhetri, D. Morrell, and A. Papandreou-Suppappola. Efficient search strategies for non-myopic sensor scheduling in target tracking. In *Asilomar Conference on Signals, Systems, and Computers*, 2004.

[58] E. K. P. Chong, R. L. Givan, and H. S. Chang. A framework for simulation-based network control via hindsight optimization. In *IEEE Conference on Decision and Control*, pages 1433–1438, 2000.

[59] Y. S. Chow, H. Robins, and D. Siegmund. *Great Expectations: The theory of Optimal Stopping*. Houghton Mifflin Company, Boiston, MA, 1971.

[60] D. Cochran. Waveform-agile sensing: opportunities and challenges. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 877–880, Philadelphia, PA, 2005.

[61] D. Cochran, D. Sinno, and A. Clausen. Source detection and localization using a multi-mode detector: a Bayesian approach. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1173–1176, Phoenix, AZ, 1999.

[62] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Advances in Neural Information Processing Systems*, 7:705–712, 1995.

[63] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, pages 129–145, 1996.

[64] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, NY, 1991.

[65] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and Other Kernel Based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[66] I. Csiszár. Information-type measures of divergence of probability distributions and indirect observations. *Studia Sci. Math. Hung.*, 2:299–318, 1967.

[67] I. Csiszár and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, Orlando FL, 1981.

[68] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw Hill, 1970.

[69] C. Dellacherie and P. A. Meyer. *Probabilities and Potential B: Theory of Martingales*. North-Holland, Amsterdam, 1982.

[70] E. V. Denardo. *Dynamic Programming Models and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[71] D. Donoho. Compressed sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, April 2006.

[72] A. Doucet. On sequential Monte Carlo methods for Bayesian filtering. Uk. tech. rep., Dept. Eng. Univ. Cambridge, 1998.

[73] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Publishing, New York, NY, 2001.

[74] A. Doucet, B.-N. Vo, C. Andrieu, and M. Davy. Particle filtering for multi-target tracking and sensor management. In *International Conference on Information Fusion*, 2002.

[75] N. Ehsan and M. Liu. Optimal bandwidth allocation in a delay channel. submitted to *JSAC*.

[76] N. Ehsan and M. Liu. Optimal channel allocation for uplink transmission in satellite communications. submitted to IEEE Transactions on Vehicular Technology.

[77] N. Ehsan and M. Liu. Server allocation with delayed state observation: sufficient conditions for the optimality an index policy. submitted to PEIS.

[78] N. Ehsan and M. Liu. On the optimal index policy for bandwidth allocation with delayed state observation and differentiated services. In *IEEE*

Annual Conference on Computer Communications, volume 3, pages 1974–1983, Hong Kong, April 2004.

[79] N. Ehsan and M. Liu. Properties of optimal resource sharing in delay channels. In *IEEE Conference on Decision and Control*, volume 3, pages 3277–3282, Paradise Island, Bahamas, 2004.

[80] N. El Karoui and I. Karatzas. Dynamic allocation problems in continuous time. *Annals of Applied Probability*, 4(2):255–286, 1994.

[81] V. V. Federov. *Theory of optimal experiments*. Academic Press, Orlando, 1972.

[82] R. A. Fisher. *The design of experiments*. Oliver and Boyd, Edinburgh, 1935.

[83] T. E. Fortmann, Y. Bar-Shalom, M. Scheffé, and S. Gelfand. Detection thresholds for tracking in clutter — A connection between estimation and signal processing. *IEEE Transactions on Automatic Control*, 30(3):221–229, March 1985.

[84] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, August 1997.

[85] E. Frostig and G. Weiss. Four proofs of Gittins' multi-armed bandit theorem. Technical report, The University of Haifa, Mount Carmel, 31905, Israel, November 1999.

[86] N. Geng, C. E. Baum, and L. Carin. On the low-frequency natural response of conducting and permeable targets. *IEEE Transactions on Geoscience and Remote Sensing*, 37:347–359, January 1999.

[87] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–177, 1979.

[88] J. C. Gittins. *Multi-Armed Bandit Allocation Indices*. John Wiley and Sons, New York, NY, 1989.

[89] J. C. Gittins and D. M. Jones. A dynamic allocation index for sequential design of experiments. *Progress in Statistics, Euro. Meet. Statis.*, 1:241–266, 1972.

[90] K. D. Glazebrook, J. Niño Mora, and P. S. Ansell. Index policies for a class of discounted restless bandits. *Advances in Applied Probability*, 34(4):754–774, 2002.

[91] K. D. Glazebrook and D. Ruiz-Hernandez. A restless bandit approach to stochastic scheduling problems with switching costs. Preprint, March 2005.

[92] G. Golubev and B. Levit. Sequential recovery of analytic periodic edges in the binary image models. *Mathematical Methods of Statistics*, 12:95–115, 2003.

[93] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. A novel approach to non-linear and non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, 1993.

[94] E. Gottlieb and R. Harrigan. The umbra simulation framework. Sand2001-1533 (unlimited release), Sandia National Laboratory, 2001.

[95] C. H. Gowda and R. Viswanatha. Performance of distributed CFAR test under various clutter amplitudes. *IEEE Transactions on Aerospace and Electronic Systems*, 35:1410–1419, 1999.

[96] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, Apr. 1984.

[97] J. A. Gubner. *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge University Press, New York, NY, 2006.

[98] P. Hall and I. Molchanov. Sequential methods for design-adaptive estimation of discontinuities in regression curves and surfaces. *Annals of Statistics*, 31(3):921–941, 2003.

[99] P. Hanselman, C. Lawrence, E. Fortunato, B. Tenney, and E. Blasch. Dynamic tactical targeting. In *Conference on Battlefield Digitization and Network-Centric Systems IV*, volume SPIE 5441, pages 36–47, 2004.

[100] J. P. Hardwick and Q. F. Stout. Flexible algorithms for creating and analyzing adaptive sampling procedures. In N. Flournoy, W. F. Rosenberger, and W. K. Wong, editors, *New Developments and Applications in Experimental Design*, volume 34 of *Lecture Notes - Monograph Series*, pages 91–105. Institute of Mathematical Statistics, 1998.

[101] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, Basel, CH, 2001.

[102] J. Havrda and F. Chárvat. Quantification method of classification processes. *Kiberbetika Cislo*, 1(3):30–34, 1967.

[103] Y. He and E. K. P. Chong. Sensor scheduling for target tracking in sensor networks. In *IEEE Conference on Decision and Control*, pages 743–748, 2004.

[104] Y. He and E. K. P. Chong. Sensor scheduling for target tracking: A Monte Carlo sampling approach. *Digital Signal Processing*, 16(5):533–545, September 2006.

[105] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom. Multisensor resource deployment using posterior Cramér-Rao bounds. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):399– 416, April 2004.

[106] A. O. Hero, B. Ma, O. Michel, and J. Gorman. Applications of entropic spanning graphs. *IEEE Signal Processing Magazine*, 19(2):85–95, 2002.

[107] A. O. Hero, B. Ma, O. Michel, and J. D. Gorman. Alpha divergence for classification, indexing and retrieval. Technical Report Technical Report 328, Comm. and Sig. Proc. Lab. (CSPL), Dept. EECS, The University of Michigan, 2001.

[108] K. J. Hintz. A measure of the information gain attributable to cueing. *IEEE Transactions on Systems, Man and Cybernetics*, 21(2):237–244, 1991.

[109] K. J. Hintz and E. S. McVey. Multi-process constrained estimation. *IEEE Transactions on Systems, Man and Cybernetics*, 21(1):434–442, January/February 1991.

[110] M. Horstein. Sequential decoding using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963.

[111] R. Howard. *Dynamic Programming and Markov Processes*. John Wiley and Sons, New York, NY, 1960.

[112] C. Hue, J.-P. Le Cadre, and P. Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50:309–325, 2002.

[113] C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38:791–812, 2002.

[114] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *International Conference on Computer Vision*, 2001.

[115] T. Ishikida. *Informational Aspects of Decentralized Resource Allocation*. PhD thesis, University of California, Berkeley, 1992.

[116] T. Ishikida and P. Varaiya. Multi-armed bandit problem revisited. *Journal of Optimization Theory and Applications*, 83:113–154, 1994.

[117] J. Jacod and P. Protter. *Probability Essentials*. Springer-Verlag, 2003.

[118] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, NY, 1970.

[119] S. Ji, R. Parr, and L. Carin. Non-myopic multi-aspect sensing with partially observable Markov decision processes. *IEEE Transactions on Signal Processing*, 55(6):2720–2730, 2007.

[120] S. Julier and J. Uhlmann. Unscented filtering and non-linear estimation. *Proceedings of the IEEE*, 92:401–422, 2004.

[121] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[122] R. Karlsson and F. Gustafsson. Monte Carlo data association for multiple target tracking. In *IEE Workshop on Target Tracking: Algorithms and Applications*, 2001.

[123] H. Kaspi and A. Mandelbaum. Multi-armed bandits in discrete and continuous time. *Annals of Applied Probability*, 8:1270–1290, 1998.

[124] K. Kastella. Discrimination gain for sensor management in multitarget detection and tracking. In *IEEE-SMC and IMACS Multiconference*, volume 1, pages 167–172, 1996.

[125] K. Kastella. Discrimination gain to optimize classification. *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, 27(1), January 1997.

[126] M. N. Katehakis and U. G. Rothblum. Finite state multi-armed bandit problems: Sensitive-discount, average-reward and average-overtaking optimality. *Annals of Applied Probability*, 6:1024–1034, 1996.

[127] M. N. Katehakis and A. F. Veinott, Jr. The multi-armed bandit problem: Decomposition and computation. *Mathematics of Operations Research*, 12:262–268, 1987.

[128] M. J. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *International Joint Conference on Artificial Intelligence*, pages 1324–1331, 1999.

[129] F. P. Kelly. Multi-armed bandits with discount factor near one: The Bernoulli case. *Annals of Statistics*, 9:987–1001, 1981.

[130] D. J. Kershaw and R. J. Evans. Optimal waveform selection for tracking systems. *IEEE Transactions on Information Theory*, 40(5):1536–50, September 1994.

[131] D. J. Kershaw and R. J. Evans. Waveform selective probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 33(4):1180–88, October 1997.

[132] G. P. Klimov. Time sharing service systems I. *Theory of Probability and its Applications (in Russian: Teoriya Veroyatnostei i ee Primeneniya)*, 19:532–551, 1974.

[133] G. P. Klimov. Time sharing service systems II. *Theory of Probability and its Applications (in Russian: Teoriya Veroyatnostei i ee Primeneniya)*, 23:314–321, 1978.

[134] E. D. Kolaczyk and R. D. Nowak. Multiscale likelihood analysis and complexity penalized estimation. *Annals of Statistics*, 32(2):500–527, 2004.

[135] A. Korostelev and J.-C. Kim. Rates of convergence for the sup-norm risk in image models under sequential designs. *Statistics and Probability Letters*, 46:391–399, 2000.

[136] A. P. Korostelev. On minimax rates of convergence in image models under sequential design. *Statistics and Probability Letters*, 43:369–375, 1999.

[137] A. P. Korostelev and A. B. Tsybakov. *Minimax Theory of Image Reconstruction*. Springer Lecture Notes in Statistics, 1993.

[138] C. Kreucher, D. Blatt, A. Hero, and K. Kastella. Adaptive multimodality sensor scheduling for detection and tracking of smart targets. *Digital Signal Processing*, 16(5):546–567, 2005.

[139] C. Kreucher, A. Hero, K. Kastella, and D. Chang. Efficient methods of non-myopic sensor management for multitarget tracking. In *IEEE Conference on Decision and Control*, 2004.

[140] C. Kreucher, A. O. Hero, and K. Kastella. Multiple model particle filtering for multi-target tracking. In *Workshop on Adaptive Sensor Array Processing*, 2004.

[141] C. Kreucher, K. Kastella, and A. Hero. Multi-target sensor management using alpha divergence measures. In *International Conference on Information Processing in Sensor Networks*, 2003.

[142] C. M. Kreucher, A. O. Hero, and K. Kastella. A comparison of task driven and information driven sensor management for target tracking. In *IEEE Conference on Decision and Control*, 2005.

[143] C. M. Kreucher, A. O. Hero, K. D. Kastella, and M. R. Morelande. An information-based approach to sensor management in large dynamic networks. *Proceedings of the IEEE*, 95(5):978–999, May 2007.

[144] C. M. Kreucher, K. Kastella, and A. O. Hero. Information based sensor management for multitarget tracking. In *SPIE Conference on Signal and Data Processing of Small Targets*, 2003.

[145] C. M. Kreucher, K. Kastella, and A. O. Hero. Multitarget tracking using the joint multitarget probability density. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1396–1414, 2005.

[146] C. M. Kreucher, K. Kastella, and A. O. Hero. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, 2005.

[147] V. Krishnamurthy. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

[148] V. Krishnamurthy and R. J. Evans. Hidden Markov model multiarmed bandits: A methodology for beam scheduling in multitarget tracking. *IEEE Transactions on Signal Processing*, 49(12):2893–2908, 2001.

[149] V. Krishnamurthy and R. J. Evans. Correction to hidden Markov model multi-arm bandits: A methodology for beam scheduling in multi-target tracking. *IEEE Transactions on Signal Processing*, 51(6):1662–1663, 2003.

[150] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, 7:231–238, 1995.

[151] W. S. Kuklinski. Adaptive sensor tasking and control. In *MITRE 2005 Technology Symposium*. MITRE Corporation, 2005.

[152] S. Kullback. *Information Theory and Statistics*. Dover, 1978.

[153] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Prentice Hall, 1986.

[154] H. Kushner. *Introduction to Stochastic Control*. Holt, Rinehart and Winston, New York, NY, 1971.

[155] B. F. La Scala, B. Moran, and R. Evans. Optimal scheduling for target detection with agile beam radars. In *NATO SET-059 Symposium on Target Tracking and Sensor Data Fusion for Military Observation Systems*, 2003.

[156] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

[157] R. E. Larson and J. L. Casti. *Principles of Dynamic Programming, Parts 1-2*. Marcel Dekker, New York, NY, 1982.

[158] X. Liao, H. Li, and B. Krishnapuram. An *m*-ary KMP classifier for multi-aspect target classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 61–64, 2004.

[159] M. L. Littman. The witness algorithm: Solving partially observable Markov decision processes. Technical Report CS-94-40, Brown University, 1994.

[160] J. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 1998.

[161] C. Lott and D. Teneketzis. On the optimality of an index rule in multichannel allocation for single-hop mobile networks with multiple service classes. *Probability in the Engineering and Informational Sciences*, 14:259–297, 2000.

[162] W. S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28(1):47–65, 1991.

[163] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4:590–604, 1992.

[164] D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2004.

[165] R. Mahler. Global optimal sensor allocation. In *National Symposium on Sensor Fusion*, volume 1, pages 167–172, 1996.

[166] A. Mandelbaum. Discrete multiarmed bandits and multiparameter processes. *Probability Theory and Related Fields*, 71:129–147, 1986.

[167] A. Mandelbaum. Continuous multi-armed bandits and multiparameter processes. *Annals of Probability*, 15:1527–1556, 1987.

[168] S. Maskell, M. Rollason, N. Gordon, and D. Salmond. Efficient particle filtering for multiple target tracking with application to tracking in structured images. In *SPIE Conference on Signal and Data Processing of Small Targets*, 2002.

[169] M. McClure and L. Carin. Matched pursuits with a wave-based dictionary. *IEEE Transactions on Signal Processing*, 45:2912–2927, December 1997.

[170] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.

[171] J. Mickova. Stochastic scheduling with multi-armed bandits. Master's thesis, University of Melbourne, Australia, 2000.

[172] M. I. Miller, A. Srivastava, and U. Grenander. Conditional mean estimation via jump-diffusion processes in multiple target tracking/recognition. *IEEE Transactions on Signal Processing*, 43:2678–2690, 1995.

[173] G. E. Monahan. A survey of partially observable Markov decision processes: Theory, models and algorithms. *Management Science*, 28(1):1–16, 1982.

[174] M. Morelande, C. M. Kreucher, and K. Kastella. A Bayesian approach to multiple target detection and tracking. *IEEE Transactions on Signal Processing*, 55(5):1589–1604, 2007.

[175] S. Musick and R. Malhotra. Chasing the elusive sensor manager. In *IEEE National Aerospace and Electronics Conference*, volume 1, pages 606–613, 1994.

[176] D. Mušicki, S. Challa, and S. Suvorova. Multi target tracking of ground targets in clutter with LMIPDA-IMM. In *International Conference on Information Fusion*, Stockholm, Sweden, July 2004.

[177] D. Mušicki and R. Evans. Clutter map information for data association and track initialization. *IEEE Transactions on Aerospace and Electronic Systems*, 40(4):387–398, April 2001.

[178] D. Mušicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1240, June 1994.

[179] P. Nash. *Optimal Allocation of Resources Between Research Projects*. PhD thesis, Cambridge University, 1973.

[180] F. Nathanson. *Radar Design Principles*. McGraw Hill, New York, 1969.

[181] A. Nedic and M. K. Schneider. Index rule-based management of a sensor for searching, tracking, and identifying. In *Tri-Service Radar Symposium*, Boulder Colorado, June 2003.

[182] A. Nehorai and A. Dogandžić. Cramér-Rao bounds for estimating range, velocity and direction with an active array. *IEEE Transactions on Signal Processing*, 49(6):1122–1137, June 2001.

[183] J. Niño-Mora. Restless bandits, partial conservation laws, and indexability. *Advances in Applied Probability*, 33:76–98, 2001.

[184] J. Niño-Mora. Dynamic allocation indices for restless projects and queuing admission control: a polyhedral approach. *Mathematical Programming, Series A*, 93:361–413, 2002.

[185] R. Niu, P. Willett, and Y. Bar-Shalom. From the waveform through the resolution cell to the tracker. In *IEEE Aerospace Conference*, March 1999.

[186] R. Nowak, U. Mitra, and R. Willett. Estimating inhomogeneous fields using wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):999–1006, 2004.

[187] M. Orton and W. Fitzgerald. A Bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50:216–223, 2002.

[188] D. G. Pandelis and D. Teneketzis. On the optimality of the Gittins index rule in multi-armed bandits with multiple plays. *Mathematical Methods of Operations Research*, 50:449–461, 1999.

[189] J. Pineau, G. Gordon, and S. Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence*, August 2003.

[190] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.

[191] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley and Sons, New York, NY, 1993.

[192] M. L. Puterman, editor. *Dynamic Programming and its Applications*. Academic Press, New York, NY, 1978.

[193] M. L. Puterman. *Markov Decision Problems: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, New York, NY, 1994.

[194] R. Raich, J. Costa, and A. O. Hero. On dimensionality reduction for classification and its application. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, May 2006.

[195] A. Rényi. On measures of entropy and information. In *Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561, 1961.

[196] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, January 2004.

[197] S. M. Ross. *Applied Probability Models with Optimization Applications*. Dover Publications, New York, NY, 1970.

[198] S. M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, New York, NY, 1983.

[199] N. Roy, G. Gordon, and S. Thrun. Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, 23:1–40, 2005.

[200] P. Runkle, P. Bharadwaj, and L. Carin. Hidden Markov model multiaspect target classification. *IEEE Transactions on Signal Processing*, 47:2035–2040, July 1999.

[201] P. Runkle, L. Carin, L. Couchman, T. Yoder, and J. Bucaro. Multi-aspect identification of submerged elastic targets via wave-based matching pursuits and hidden Markov models. *J. Acoustical Soc. Am.*, 106:605–616, August 1999.

[202] J. Rust. Chapter 14: Numerical dynamic programming in economics. In H. Amman, D. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*. Elsevier, North Holland, 1996.

[203] J. Rust. Using randomization to break the curse of dimensionality. *Econometrica*, 65:487–516, 1997.

[204] W. Schmaedeke and K. Kastella. Event-averaged maximum likelihood estimation and information-based sensor management. *Proceedings of SPIE*, 2232:91–96, June 1994.

[205] M. K. Schneider, G. L. Mealy, and F. M. Pait. Closing the loop in sensor fusion systems: Stochastic dynamic programming approaches. In *American Control Conference*, 2004.

[206] D. Schulz, D. Fox, and J. Hightower. People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filter. In *International Joint Conference on Artificial Intelligence*, 2003.

[207] N. Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49:796–802, 2001.

[208] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

[209] A. N. Shireyaev. *Optimal Stopping Rules*. Springer-Verlag, 1978.

[210] A. N. Shireyaev. *Probability*. Springer-Verlag, 1995.

[211] A. Singh, R. Nowak, and P. Ramanathan. Active learning for adaptive mobile sensing networks. In *International Conference on Information Processing in Sensor Networks*, Nashville, TN, April 2006.

[212] D. Sinno. *Attentive Management of Configurable Sensor Systems*. PhD thesis, Arizona State University, 2000.

[213] D. Sinno and D. Cochran. Dynamic estimation with selectable linear measurements. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2193–2196, Seattle, WA, 1998.

[214] D. Sinno, D. Cochran, and D. Morrell. Multi-mode detection with Markov target motion. In *International Conference on Information Fusion*, volume WeD1, pages 26–31, Paris, France, 2000.

[215] S. P. Sira, D. Cochran, A. Papandreou-Suppappola, D. Morrell, W. Moran, S. Howard, and R. Calderbank. Adaptive waveform design for improved detection of low RCS targets in heavy sea clutter. *IEEE Journal on Selected Areas in Signal Processing*, 1(1):55–66, June 2007.

[216] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell. Time-varying waveform selection and configuration for agile sensors in tracking applications. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 881–884, March 2005.

[217] M. I. Skolnik. *Introduction to Radar Systems*. McGraw-Hill, 3rd edition, 2001.

[218] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

[219] E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University, 1971.

[220] E. J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2):282–304, 1978.

[221] N. O. Song and D. Teneketzis. Discrete search with multiple sensors. *Mathematical Methods of Operations Research*, 60:1–14, 2004.

[222] Statlog. Landsat MSS data.

[223] L. D. Stone, C. A. Barlow, and T. L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, Boston, MA, 1999.

[224] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36:111–147, 1974.

[225] C. Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12:576–592, 1965.

[226] K. Sung and P. Niyogi. Active learning for function approximation. *Proc. Advances in Neural Information Processing Systems*, 7, 1995.

[227] R. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[228] S. Suvorova, S. D. Howard, W. Moran, and R. J. Evans. Waveform libraries for radar tracking applications: Maneuvering targets. In *Defence Applications of Signal Processing*, 2004.

[229] I. J. Taneja. New developments in generalized information measures. *Advances in Imaging and Electron Physics*, 91:37–135, 1995.

[230] G. Tesauro. Temporal difference learning and TD-gammon. *Communications of the ACM*, 38(3), March 1995.

[231] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *International Conference on Machine Learning*, pages 999–1006, 2000.

[232] J. N. Tsitsiklis. A lemma on the multiarmed bandit problem. *IEEE Transactions on Automatic Control*, 31:576–577, 1986.

[233] B. E. Tullsson. Monopulse tracking of Rayleigh targets: A simple approach. *IEEE Transactions on Aerospace and Electronic Systems*, 27:520–531, 1991.

[234] M. Van Oyen, D. Pandelis, and D. Teneketzis. Optimality of index policies for stochastic scheduling with switching penalties. *Journal of Applied Probability*, 29:957–966, 1992.

[235] M. P. Van Oyen and D. Teneketzis. Optimal stochastic scheduling of forest networks with switching penalties. *Advances in Applied Probability*, 26:474–479, 1994.

[236] H. L. van Trees. *Detection, Estimation, and Modulation Theory: Part I*. John Wiley and Sons, New York, NY, 1968.

[237] H. L. van Trees. *Detection, Estimation and Modulation Theory, Part III*. John Wiley and Sons, New York, NY, 1971.

[238] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, NY, 1998.

[239] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[240] P. P. Varaiya, J. C. Walrand, and C. Buyukkov. Extensions of the multiarmed bandit problem: The discounted case. *IEEE Transactions on Automatic Control*, 30:426–439, 1985.

[241] M. Veth, J. Busque, D. Heesch, T. Burgess, F. Douglas, and B. Kish. Affordable moving surface target engagement. In *IEEE Aerospace Conference*, volume 5, pages 2545–2551, 2002.

[242] P. Vincent and Y. Bengio. Kernel matching pursuit. *Machine Learning*, 48:165–187, 2002.

[243] A. Wald. *Sequential Analysis*. John Wiley and Sons, New York, NY, 1947.

[244] J. Wang, A. Dogandžić, and A. Nehorai. Maximum likelihood estimation of compound-Gaussian clutter and target parameters. *IEEE Transactions on Signal Processing*, 54:3884–3898, October 2006.

[245] R. B. Washburn, M. K. Schneider, and J. J. Fox. Stochastic dynamic programming based approaches to sensor resource management. In *International Conference on Information Fusion*, volume 1, pages 608–615, 2002.

[246] R. R. Weber. On Gittins index for multiarmed bandits. *Annals of Probability*, 2:1024–1033, 1992.

[247] R. R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27:637–648, 1990.

[248] C. C. White III. Partially observed Markov decision processes: A survey. *Annals of Operations Research*, 32, 1991.

[249] P. Whittle. Multi-armed bandits and Gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42:143–149, 1980.

[250] P. Whittle. Arm-acquiring bandits. *Annals of Probability*, 9:284–292, 1981.

[251] P. Whittle. *Optimization Over Time: Dynamic Programming and Stochastic Control*. John Wiley and Sons, New York, NY, 1983.

[252] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25A:287–298, 1988.

[253] P. Whittle. Tax problems in the undiscounted case. *Journal of Applied Probability*, 42(3):754–765, 2005.

[254] R. Willett, A. Martin, and R. Nowak. Backcasting: Adaptive sampling for sensor networks. In *Information Processing in Sensor Networks*, 26-27 April, Berkeley, CA, USA, 2004.

[255] I. J. Won, D. A. Keiswetter, and D. R. Hanson. GEM-3: A monostatic broadband electromagnetic induction sensor. *J. Environ. Eng. Geophys.*, 2:53–64, March 1997.

[256] G. Wu, E. K. P. Chong, and R. L. Givan. Burst-level congestion control using hindsight optimization. *IEEE Transactions on Automatic Control*, 47:979–991, 2002.

[257] R. W. Yeung. *A First Course in Information Theory*. Springer, 2002.

[258] H. Yu and D. P. Bertsekas. Discretized approximations for pomdp with average cost. In *Conference on Uncertainty in Artificial Intelligence*, pages 619–627, 2004.

[259] Y. Zhang, L. M. Collins, H. Yu, C. E. Baum, and L. Carin. Sensing of unexploded ordnance with magnetometer and induction data: Theory and signal processing. *IEEE Transactions on Geoscience and Remote Sensing*, 41:1005–1015, May 2003.

[260] Y. Zhang, X. Liao, and L. Carin. Detection of buried targets via active selection of labeled data: application to sensing subsurface uxo. *IEEE Transactions on Geoscience and Remote Sensing*, 42(11):2535–2543, 2004.

[261] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, pages 61–72, March 2002.

Index

SIGNALS AND COMMUNICATION TECHNOLOGY

(continued from page ii)

Digital Interactive TV and Metadata
Future Broadcast Multimedia
A. Lugmayr, S. Niiranen, and S. Kalli
ISBN 3-387-20843-7

Adaptive Antenna Arrays
Trends and Applications
S. Chandran (Ed.)
ISBN 3-540-20199-8

Digital Signal Processing with Field Programmable Gate Arrays
U. Meyer-Baese
ISBN 3-540-21119-5

Neuro-Fuzzy and Fuzzy Neural Applications in Telecommunications
P. Stavroulakis (Ed.) ISBN 3-540-40759-6

SDMA for Multipath Wireless Channels
Limiting Characteristics and Stochastic Models
I.P. Kovalyov ISBN 3-540-40225-X

Digital Television
A Practical Guide for Engineers
W. Fischer ISBN 3-540-01155-2

Speech Enhancement
J. Benesty (Ed.)
ISBN 3-540-24039-X

Multimedia Communication Technology
Representation, Transmission and Identification of Multimedia Signals
J.R. Ohm ISBN 3-540-01249-4

Information Measures
Information and its Description in Science and Engineering
C. Arndt ISBN 3-540-40855-X

Processing of SAR Data
Fundamentals, Signal Processing, Interferometry
A. Hein ISBN 3-540-05043-4

Chaos-Based Digital Communication Systems
Operating Principles, Analysis Methods, and Performance Evalutation
F.C.M. Lau and C.K. Tse
ISBN 3-540-00602-8

Adaptive Signal Processing
Application to Real-World Problems
J. Benesty and Y. Huang (Eds.)
ISBN 3-540-00051-8

Multimedia Information Retrieval and Management Technological
Fundamentals and Applications D. Feng, W.C. Siu, and H.J. Zhang (Eds.)
ISBN 3-540-00244-8

Structured Cable Systems
A.B. Semenov, S.K. Strizhakov,and I.R. Suncheley
ISBN 3-540-43000-8

UMTS
The Physical Layer of the Universal Mobile Telecommunications System
A. Springer and R. Weigel
ISBN 3-540-42162-9

Advanced Theory of Signal Detection
Weak Signal Detection in Generalized Obeservations
I. Song, J. Bae, and S.Y. Kim
ISBN 3-540-43064-4

Wireless Internet Access over GSM and UMTS
M. Taferner and E. Bonek
ISBN 3-540-42551-9