Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking

Vikram Krishnamurthy, *Senior Member, IEEE,* and Robin J. Evans

***Abstract*—In this paper, we derive optimal and suboptimal beam scheduling algorithms for electronically scanned array tracking systems. We formulate the scheduling problem as a multiarm bandit problem involving hidden Markov models (HMMs). A finite-dimensional optimal solution to this multiarm bandit problem is presented. The key to solving any multiarm bandit problem is to compute the Gittins index. We present a finite-dimensional algorithm that computes the Gittins index. Suboptimal algorithms for computing the Gittins index are also presented. Numerical examples are presented to illustrate the algorithms.**

***Index Terms*—Dynamic programming, hidden Markov models, optimal beam steering, scheduling, sequential decision procedures.**

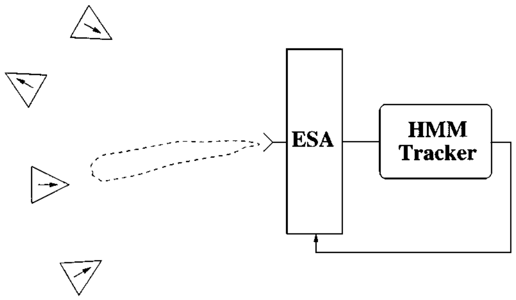


Fig. 1. Multitarget tracking with one intelligent sensor. The scheduling problem involves which single target to track with the steerable electronic scanned array (ESA).

I. Introduction and Problem Formulation

CONSIDER the following hidden Markov model (HMM) target tracking problem: P targets (e.g., aircraft) are being tracked by an electronically scanned array (ESA) with only one steerable beam. The coordinates of each target $s_k^{(p)}$, $p = 1, 2 \ldots, P$ evolve according to a finite state Markov chain (see [27]), and the conditional mean estimate target state is computed by an HMM tracker. We assume that the target coordinates evolve independently of each other—that is, the Markov chains $s_k^{(p)}$ $p = 1, 2 \ldots, P$ are independent of each other. Since we have assumed that there is only one steerable beam, we can obtain noisy measurements $y_k^{(p)}$ of **only one target** at any given time instant. Our aim is to answer the following question: *Which single target should the tracker choose to observe at each time instant in order to optimize some specified cost function?* Fig.1 shows a schematic representation of the problem.

In this paper, we give a complete solution to this problem. We formulate the problem as an *HMM multiarm bandit problem*. Then, a finite-dimensional optimal algorithm and suboptimal algorithms are derived for computing the solution.

The problem stated above is an integral part of any agile beam tracking system [1], [6], [7]. Practical implementation of such systems is quite complex, and beam scheduling must handle a variety of search, identification, and tracking functions. Our work considers one important part of a complete agile beam tracking system. We show that the multiarm bandit formulation can provide a rigorous and potentially useful approach to the problem of scheduling an agile beam among a finite number of established target tracks.

The multiarm bandit problem is an example of a dynamic stochastic scheduling problem (resource allocation problem). The "standard" multiarm bandit problem involves a fully observed finite state Markov chain and is merely a finite state Markov decision process (MDP) problem with a rich structure. We will refer to this standard problem as the Markov chain multiarm bandit problem. Numerous applications of the Markov chain multiarm bandit problem appear in the operations research and stochastic control literature; see [15] and [30] for examples in job scheduling and resource allocation for manufacturing systems and [5] for applications in scheduling of traffic in broadband networks. Several fast algorithms have been recently proposed to solve this Markov chain multiarm bandit problem (see [16] and [29]).

However, due to measurement noise at the sensor, the above multitarget tracking example cannot be formulated as a standard Markov chain multiarm bandit problem. Instead, we will formulate the problem as a multiarm bandit problem involving HMMs, which is considerably more difficult to solve. To develop suitable algorithms for the tracking problem, in this paper, we first give a complete solution to the multiarm bandit problem for an HMM. In the operations research literature, such problems are also known as partially observed Markov decision processes (POMDPs) [20], [21].

Our paper focuses on designing optimal beam scheduling algorithms by formulating the problem as an HMM multiarm bandit problem. Other applications of the HMM multiarm

Manuscript received February 1, 2000; revised September 11, 2001. This work was supported by an ARC large grant and the Centre of Expertise in Networked Decision Systems. The associate editor coordinating the review of this paper and approving it for publication was Prof. Douglas Cochran.

The authors are with the Department of Electrical Engineering, University of Melbourne, Parkville, Victoria, Australia (e-mail: vikram@ee.mu.oz.au; r.evans@ee.mu.oz.au).

Publisher Item Identifier S 1053-587X(01)10488-5.

bandit problem include the above Markov chain multiarm bandit applications when noisy observations of the state are available, robot navigation systems [13], learning behavior [24], and fault-tolerant systems [14].

A. HMM Multiarm Bandit Problem

1) Markov Chain Multiarm Bandit Problem: To motivate the HMM multiarm bandit problem, we first outline the well known finite state Markov chain multiarm bandit problem [4], [18]. Consider P independent projects (or targets) $p = 1, \ldots, P$. At each discrete-time instant k, only one of these projects can be worked on. We assume that each project p has a finite number of states $\mathcal{N}_p$. Let $s_k^{(p)}$ denote the state of project p at time k. If project p is worked on at time k, one incurs an instantaneous cost of $\beta^k R(s_k^{(p)}, p)$, where $0 < \beta < 1$ denotes the discount factor; the state $s_k^{(p)}$ evolves according to an $\mathcal{N}_p$-state homogeneous Markov chain with transition probability

$$P\left(s_{k+1}^{(p)} = j \,\middle|\, s_k^{(p)} = i\right) = \left(a_{ij}^{(p)}\right)_{i,j \in \mathcal{N}_p} \quad \text{if project } p \text{ is worked on at time } k. \tag{1}$$

Let $A^{(p)}$ denote the transition probability matrix $(a_{ij}^{(p)})_{i,j \in \mathcal{N}_p}$. The states of all the other $(P-1)$ idle projects are unaffected, i.e.,

$$s_{k+1}^{(p)} = s_k^{(p)}, \quad \text{if project p is idle at time } k.$$

Let the control variable $u_k \in \{1, \ldots, P\}$ denote which project is worked on at time k. Consequently, $s_{k+1}^{(u_k)}$ denotes the state of the active project at time $k+1$. Let the policy μ denote the sequence of controls $\{u_k, k = 1, 2, \ldots\}$. The total expected discounted cost over an infinite time horizon is given by

$$J_\mu = \mathbf{E}\left[\sum_{k=0}^{\infty} \beta^k R\left(s_k^{(u_k)}, u_k\right)\right] \tag{2}$$

where $\mathbf{E}$ denotes mathematical expectation. For the above cost function to be well defined, we assume that $R(s_k^{(u_k)}, u_k)$ is uniformly bounded from above and below (see [4, p. 54].

Define $\mathcal{U}$ as the class of all admissible policies μ. Solving the Markov chain multiarm bandit problem [29] involves computing the optimal policy $\arg\min_{\mu \in \mathcal{U}} J_\mu$, where $\mathcal{U}$ denotes the set of sequences u_k that map s_k to $\{1, \ldots, P\}$ at each time instant.[1]

2) Preliminary Formulation of HMM Multiarm Bandit Problem: In this paper, motivated by the agile beam scheduling application, we consider the conceptually more difficult HMM multiarm bandit problem where we assume that the state of the active project $s_{k+1}^{(p)}$ is not directly observed. Instead, if $u_k = p$, noisy measurements (observations) $y_{k+1}^{(p)}$ of the active project state $s_{k+1}^{(p)}$ are available at time $k+1$. Assume that these observations $y_{k+1}^{(p)}$ belong to a finite set $\mathcal{M}_p$ indexed by $m^{(p)} = 1, \ldots, \mathcal{M}_p$. Denote the observation history[2] at time k as

$$Y_k = \left(y_1^{(u_0)}, \ldots, y_k^{(u_{k-1})}\right).$$

Let $B^{(p)} = (b_{im}^{(p)})_{i \in \mathcal{N}_p, m \in \mathcal{M}_p}$ denote the observation probability (symbol probability) matrix of the HMM, where each element $b_{im}^{(p)} \triangleq P(y_{k+1}^{(p)} = m \,|\, s_{k+1}^{(p)} = i, u_k = p)$. *Our aim is to solve the HMM multiarm bandit problem, i.e., determine the optimal policy that yields the minimum cost in* (2)

$$\mu^* = \arg\min_{\mu \in \mathcal{U}} J_\mu. \tag{3}$$

Note that unlike the standard Markov chain bandit problem, in the HMM case, the class of admissible policies $\mathcal{U}$ comprises of the set of sequences u_k that map Y_k to $\{1, \ldots, P\}$; see [4] for details. Let $J^* = \min_{\mu \in \mathcal{U}} J_\mu$ denote the optimal cost.

Remark: Because of the assumptions that $\mathcal{M}_p$ and $\mathcal{N}_p$ are finite sets and $R(s_k^{(u_k)}, u_k)$ are uniformly bounded from above and below, the optimization (3) is well defined [20].

B. Information State Formulation of HMM Multiarm Bandit Problem

The above HMM multiarm bandit problem is a infinite horizon POMDP with a rich structure that considerably simplifies the solution, as will be shown later. First, however, as is standard with partially observed stochastic control problems—we convert the partially observed multiarm bandit problem to a fully observed multiarm bandit problem defined in terms of the *information state*; see [4] or [18] for the general methodology.

For each target p, the information state at time k, which we will denote by $x_k^{(p)}$-column vector of dimension $\mathcal{N}_p$, is defined as the conditional filtered density of the Markov chain state $s_k^{(p)}$ given the observation history $Y_k^{(p)} = (y_0^{(p)}, \ldots, y_k^{(p)})$ and scheduling history $U_{k-1} = (u_0, \ldots, u_{k-1})$

$$x_k^{(p)}(i) \triangleq P\left(s_k^{(p)} = i \,\middle|\, Y_k, U_{k-1}\right), \quad i = 1, \ldots, \mathcal{N}_p. \tag{4}$$

The information state can be computed recursively by the HMM state filter (which is also known as the “forward algorithm” or “Baum’s algorithm” [23]) according to (5).

In terms of the information state formulation, the above HMM multiarm bandit problem can be viewed as the following scheduling problem. Consider P parallel HMM state filters: one for each target. The pth HMM filter computes the state estimate (filtered density) $x_k^{(p)}$ of the pth target $p \in \{1, \ldots, P\}$. At each time instant, the beam is directed toward only one of the P targets, say, target p, resulting in an observation $y_{k+1}^{(p)}$.

[1] Throughout this paper, we will deal with minimizing the cost function J_μ. Clearly, this is equivalent to maximizing the reward function $-J_\mu$.

[2] Our timing assumption that u_{k-1} determines y_k is quite standard in stochastic optimization [4]. It is straightforward to rederive the results in this paper if u_k determines y_k.

This is processed by the pth HMM state filter, which updates its estimate of the target's state as

$$x_{k+1}^{(p)} = \frac{B^{(p)}\left(y_{k+1}^{(p)}\right) A^{(p)\prime} x_k^{(p)}}{\mathbf{1}' B^{(p)}\left(y_{k+1}^{(p)}\right) A^{(p)\prime} x_k^{(p)}} \tag{5}$$

if beam is directed toward target p, where if $y_{k+1}^{(p)} = m$, then $B^{(p)}(m) = \text{diag}[b_{1m}^{(p)}, \dots, b_{\mathcal{N}_p,m}^{(p)}]$ is the diagonal matrix formed by the mth column of the observation matrix $B^{(p)}$, and $\mathbf{1}$ is an $\mathcal{N}_p$-dimensional column unit vector. (Note that throughout the paper, we use $'$ to denote transpose).

The state estimates of the other $P-1$ HMM state filters remain unaffected, i.e.,

$$x_{k+1}^{(q)} = x_k^{(q)} \quad \text{if target } q \text{ is not observed} \\ q \in \{1, \dots, P\}, \quad q \neq p \tag{6}$$

Let $\mathcal{X}^{(p)}$ denote the state space of information states $x^{(p)}$, $p \in \{1, 2, \dots, P\}$. That is

$$\mathcal{X}^{(p)} = \Big\{ x^{(p)} \in \mathbb{R}^{\mathcal{N}_p} : \mathbf{1}' x^{(p)} = 1, \\ 0 < x^{(p)}(i) < 1 \text{ for all } i \in \{1, \dots, \mathcal{N}_p\} \Big\}. \tag{7}$$

Note that $\mathcal{X}^{(p)}$ is a $\mathcal{N}_p - 1$ dimensional simplex. We will subsequently refer to $\mathcal{X}^{(p)}$ as the *information state space simplex*.

Using the smoothing property of conditional expectations, the cost functional (2) can be rewritten in terms of the information state as

$$J_\mu = \mathbf{E}\left[\sum_{k=0}^{\infty} R'(u_k) x_k^{(u_k)}\right] \tag{8}$$

where $R(u_k)$ denotes the $\mathcal{N}_{u_k}$-dimensional cost vector $[R(u_k, 1), \dots, R(u_k, \mathcal{N}_{u_k})]'$, and $R(u_k, i)$ is defined in (2). The aim is to compute the optimal policy $\arg\min_{\mu \in \mathcal{U}} J_\mu$. In terms of (5) and (8), the multiarm bandit problem reads thus: Design an optimal dynamic scheduling policy to choose which target and, hence, HMM filter to use at each time instant. Note that there is a computational cost of $O(\mathcal{N}_p^2)$ computations associated with running the pth HMM filter. This cost is easily incorporated in the cost vector $R(u)$ if required.

C. Modeling Assumptions for ESA Tracking via Beam Scheduling

In this subsection, we relate the modeling assumptions of the HMM multiarm bandit problem to the agile beam multitarget tracking example discussed above.

1) Partially Observed Formulation: An obvious advantage of the HMM multiarm bandit formulation (compared with the finite state fully observed Markov chain bandit problem) is the ability to model noisy measurements. More importantly, although mathematically equivalent to (3), the information state formulation (5), (8) has significant modeling advantages. If, at any time k, the beam is directed toward a particular target p, because no measurement is obtained for the other $P-1$ targets, we assume the estimates of the state of each of these $P-1$ unobserved targets $x_k^{(q)}$ $q = 1, 2, \dots, P, q \neq p$ remain frozen at the value when each target q was last observed. Note that this is far less stringent than assuming that the actual state $s_k^{(q)}$ of the unobserved target did not evolve, which is required for the standard fully observed multiarm bandit problem. In other words, *the information state formulation reduces to assumptions on the tracking* sensor *behavior (which we can control) rather than the actual underlying Markov chain or target (which we cannot control)*. For this reason, in the rest of the paper, we will deal with the information state formulation (5) and (8) of the HMM multiarm bandit problem.

2) Bandit Formulation: The multiarm bandit modeling assumptions (5) and (6) requires that the estimated densities of unobserved targets are held fixed. This assumption is violated if an HMM state predictor (which predicts the state of the Markov chain without requiring any observations) is used for each of the $P-1$ unobserved targets at each time instant. Unfortunately, although a HMM state predictor has deterministic dynamics, the problem is no longer a multiarm bandit problem and does not have an indexable (de-coupled) solution. There are at least two reasons that justify the bandit approximation.

i) *Slow Dynamics*: Models that involve multimode sensors that track slowly moving targets have a bandit structure, as we now show. Suppose $A^{(p)} = I + \epsilon Q^{(p)}$, $p = 1, \dots, P$, where $\epsilon \to 0$ and the matrix $Q^{(p)}$ has $Q^{(p)}(i,i) < 0$, $Q^{(p)}(i,j) > 0$, $i \neq j$, and $\sum_{j=1}^{\mathcal{N}_p} Q^{(p)}(i,j) = 0$. This structure of A ensures that the Markov chain targets $s_k^{(p)}$ evolve slowly. Then, if the beam is directed toward target p, the information state of this target evolves according to the HMM filter (5). The information state for the other $P-1$ unobserved targets evolve according to HMM predictors as

$$x_{k+1}^{(q)} = A^{(q)\prime} x_k^{(q)} = \left(I + \epsilon Q^{(q)'}\right) x_k^{(q)} \\ = x_k^{(q)} + O(\epsilon), \quad q \in \{1, \dots, P\}, \quad q \neq p \tag{9}$$

which is of the form (6) as $\epsilon \to 0$.

ii) *Decoupling Approximation*: Without the bandit (indexable) approximation, the optimal solution is intractable since the state-space dimension grows exponentially with the number of targets. The bandit model approximation is perhaps the only reasonable approximation available that leads to a computationally tractable solution. For this reason, the bandit model approximation has been widely used in several papers in queuing networks; see [2] and references therein. A suboptimal method that gets around the assumption that estimates of unobserved targets remain fixed is to reinitialize the HMM multiarm bandit algorithm at regular intervals with updated estimates from all targets. Section III-B presents one such algorithm for a hybrid sensor. Reinitialization of target tracking algorithms at regular intervals is widely used, for example, in image-based and image-enhanced target tracking [12], [17].

3) Finite State Markov Assumption: The finite state Markov assumption on the coordinates of each target $s_k^{(p)}$ has the following interpretations:

i) *Multitarget Tracking* [8]: Here, $s_k^{(p)} \in \{d_1, \ldots, d_S\}$ denotes the quantized distance of the pth target from a base station, and the target distance evolves according to a finite-state Markov chain. If the beam is directed toward target p, due to measurement errors, the state $s_k^{(p)}$ is observed in noise. The HMM filter computes the optimal filtered density $x_k^{(p)}$ (and hence filtered estimate) of the target position based on the noisy measurement history Y_k. Because, for the other $P-1$ targets, no measurement is obtained at time k, the estimated target state $x_k^{(q)}$, $q = 1, 2, \ldots, P$, $q \neq p$ is kept frozen at the value when the target was last observed.

ii) *Target Identification—Identification of Friend or Foe (IFF);* For each target p, the state $s_k^{(p)} \in \{\texttt{friend}, \texttt{foe}\}$. Because target p cannot switch from a friend to foe or vice versa, $A^{(p)} = I$. The measurements $y_k^{(p)} \in \{\texttt{friend}, \texttt{foe}\}$ at the base station reflect the uncertainty implicit in the fact that an intelligent foe target (aircraft) can emulate a friendly target. The interpretation of the HMM filters is then similar to the tracking case above, except for the type of measurements made. In this case, the model is exactly a multiarmed bandit because for unobserved targets, the HMM predictor (with $A^{(p)} = I$) is identical to freezing the filtered state estimate.

iii) *Joint Tracking/Identification*: It is, of course, possible to define the state to include both target identity and position. In addition, absorbing states such as a) base-destroyed, which corresponds to a foe getting close enough and successfully attacking the base and b) foe-destroyed, which results from the base station successfully attacking a foe aircraft can be defined if required. In such a case, additional actions u_k such as attack p, which means attack target p can be defined so that $u_k \in \{\texttt{track}\,1, \ldots, \texttt{track}\,P, \texttt{attack}\,1, \ldots, \texttt{attack}\,P\}$.

iv) *Modal Estimation*: In image-based and image-enhanced tracking [12], [17], two-dimensional (2-D) imagery is used to obtain the mode (orientation) of the target in three dimensions, apart from conventional measurements (position, velocity, etc). The mode $s_k^{(p)}$ of the target p is described as a finite-state Markov chain (for the profile of a T-62 tank with three different orientations; see [28]). The sensor processor response to the finite state orientation is blurred due to the range of the target, weather conditions, etc. Finally, the blurred images are processed by an imager that outputs an estimate of the target's mode $y_k^{(p)}$. The probabilistic mapping $B^{(p)}(y_k^{(p)})$ is called the *discernability* or *confusion* matrix. If the mode of the P targets change slowly with time, the bandit approximation (9) is valid.

4) Cost Structure: For the tracking problem, the cost structure $R(s_k^{(p)}, p)$ typically would depend on the distance d_i of the pth target to the base station. In the numerical examples of Section IV, we assume $R(s_k^{(p)} = d_i, p) = \rho^{(p)} d_i + r^{(p)}$ [where $\rho^{(p)}$ and $r^{(p)}$ are target dependent constants], meaning that targets close to the base station pose a greater threat and are given higher priority by the tracking algorithm.

The following cost structures are also permissible within the multiarm bandit framework.

i) *Tax Problem* [4], [29]: The tax problem is similar to the HMM multiarm bandit problem, except that the $P-1$ HMM filters that do not receive measurements of their targets incur a cost at each time instant. That is, at each time instant k, a cost of $\sum_{q=1, q\neq p}^{P} \beta^k C(s_k^{(q)}, q)$ is incurred for the $P-1$ unobserved targets, where p is the observed target. The aim is to minimize the discounted cost over an infinite horizon. As shown in [4] and [29], the tax problem can be converted to a multiarm bandit problem with cost function for HMM filter p equal to

$$R\left(s_k^{(p)}, p\right) = -C\left(s_k^{(p)}, p\right) + \beta \mathbf{E}\left[C\left(s_{k+1}^{(p)}, p\right)\right]. \qquad (10)$$

ii) *Target-Dependent Cost*: Let $W(1), \ldots, W(P)$ denote target-dependent costs that do not depend on the state of the targets. Such target-dependent costs are useful when the various targets have different characteristics—for example, a fighter aircraft would have a different associated cost than would a slower bomber aircraft. It is easily shown that without violating the multiarm bandit structure, one can add an additional target dependent cost $K(p) = \sum_{q=1, q\neq p}^{P} W(q)$ at each time instant for the un-observed $P-1$ targets, as long as these costs $W(q)$ do not depend on the state of the targets. The cost function takes the form $R(s_k^{(p)}, p) = C(s_k^{(p)}, p) + K(p)$. As described in Section III-B, this is also useful in the implementation of a hybrid tracking sensor where the constant target costs $W(q)$ for each target can be adjusted at regular intervals to reflect the most recent information and thereby improve tracking accuracy.

iii) *Retirement Cost* ([4, ch. 1.5]): Finally, one can add the option at time k of permanently retiring from tracking all targets with a cost $\beta^k M$, with no additional costs incurred in the future. This is useful in the joint tracking/identification problem discussed above, where, if, for example, a prespecified total tracking cost $\beta^k M$ was incurred until time k, then one could assume that none of the targets are foes and permanently stop tracking the targets.

D. Summary of Main Results

An obvious brute force approach to solving the above optimization problem (3) is to define a higher (enormous) dimensional Markov chain $s_k^{(1)} \otimes s_k^{(2)} \cdots \otimes s_k^{(P)}$ (where $\otimes$ denotes tensor product) with information state $x_k^{(1)} \otimes x_k^{(2)} \cdots \otimes x_k^{(P)}$. The resulting problem is then a standard partially observed Markov decision process—and numerous algorithms exist for computing the optimal policy. Of course, the computational complexity of this brute force approach is prohibitive.

We now summarize the main results of this paper. It is well known that the multiarm bandit problem has a rich structure that results in the optimization (3) decoupling into P independent optimization problems. Indeed, it turns out that the optimal policy has an *indexable rule* [18], [30]. For each target (HMM filter) p, there is a function $\gamma^{(p)}(x_k^{(p)})$ called the *Gittins index*, which is only a function of the project p and the information state $x_k^{(p)}$, whereby the optimal scheduling policy at time k is to steer the beam toward the target with the smallest Gittins index, i.e., steer beam toward target q, where

$$q = \min_{p\in\{1,\ldots,P\}} \left\{\gamma^{(p)}\left(x_k^{(p)}\right)\right\}. \tag{11}$$

For a proof of this index rule for general multiarm bandit problems, see [30]. Thus, computing the Gittins index is a key requirement for solving any multiarm bandit problem. (For a formal definition of the Gittins index in terms of stopping times, see [29]. An equivalent definition is given in [4] in terms of the parameterized retirement cost M.)

The fundamental problem with (11) is that the Gittins index $\gamma^{(p)}(x_k^{(p)})$ must be evaluated for each $x_k^{(p)} \in \mathcal{X}^{(p)}$, which is an uncountably infinite set. In contrast, for the standard finite-state Markov multiarm bandit problem considered extensively in the literature (e.g., [15]), the Gittins index can be straightforwardly computed.

The main contribution of our paper is to present a finite-dimensional algorithm for computing the Gittins index. The contributions and organization of this paper are as follows.

1) In Section II, we show that by using the return-to-state-x_0 argument [16], the computation of the Gittins index for the HMM multiarm bandit problem can be formulated as an infinite horizon dynamic programming recursion. A value-iteration based optimal algorithm[3] is given for computing the Gittins indices $\gamma^{(p)}(x_k^{(p)})$, $p = 1, 2, \ldots, P$. The value-iteration algorithm, while finite-dimensional, can have a high off-line computational cost. We also present three computationally efficient suboptimal algorithms for computing the Gittins index.
2) In Section III, we use the results in Section II to solve the beam scheduling problem for multitarget tracking. The complete beam scheduling algorithm is given in Algorithm 1 of Section III.
3) Finally, in Section IV, numerical examples of the optimal and suboptimal algorithms are given. These compare the performance of the various proposed algorithms and also illustrate the use of the optimal algorithm in the intelligent sensor tracking application.

II. Finite-Dimensional Solution for the Gittins Index

As explained in Section I-D, the key to solving the HMM multiarm bandit problem is to compute the Gittins index for each of the P targets. This section presents optimal and suboptimal algorithms for computing the Gittins index $\gamma^{(p)}(x^{(p)})$ (see (11)) for each target $p \in \{1, 2 \ldots, P\}$. It is important to note that the Gittins indices are independent of the actual sample path of the HMM and are also mutually independent. Hence, computation of the Gittins index for each target p is off-line, independent of the Gittins indices of the other $P-1$ targets, and can be done *a priori*. Thus, to simplify notation, in this section, we will sometimes omit the superscript "(p)" in $x^{(p)}$. The complete optimal scheduling algorithm in terms of the Gittins indices is summarized in Section III.

A. Solution Methodology

The procedure we will develop for computing the Gittins index for the HMM multiarm bandit problem consists of the following four steps.

1) Formulate the problem of computing the Gittins index function as an infinite horizon return-to-state-x_0 dynamic programming problem; see Section II-B.
2) Use a value-iteration based algorithm to approximate the infinite horizon problem in Step 1 by a finite horizon dynamic programming problem. We will show in Section II-C that this approximation can be made arbitrarily accurate.
3) Derive a finite-dimensional solution to the finite horizon problem of Step 3. This is done in Section II-D. In particular, we will show in Theorem 2 that the Gittins index of the finite horizon dynamic programming problem can be exactly represented as a convex combination of a finite number of piecewise linear segments.
4) With the above three steps, we present an algorithm for computing the piecewise linear segments of Step 3 that yield the Gittins index. This is done in Section II-E. The basic idea is to iterate the vectors from Step 3 over a finite horizon to obtain a finite vector representation of the Gittins index function over the entire state space of a HMM. Step 2 guarantees that the resulting solution is arbitrarily close to the optimal solution.

B. Return-to-State Formulation for Computing the Gittins Index

For arbitrary multiarm bandit problems, it is shown in [16] that the Gittins index can be calculated by solving an associated infinite horizon discounted stochastic control problem called the "return-to-state" (restart) problem. In this paper, we will use this approach to compute the Gittins index for the HMM multiarm bandit problem.

The return-to-state formalism in terms of our HMM multiarm bandit problem is as follows. For target p, consider the problem where, given information state $x_k^{(p)}$ at time k, there are two options:

1) Continue, which incurs a cost $\beta^k R(x_k^{(p)}, p)$ and evolves $x_{k+1}^{(p)}$ according to (5);
OR
2) Restart the project, which moves to a fixed state $\bar{x}^{(p)}$, incurs a cost $\beta^k R(\bar{x}^{(p)}, p)$, and moves the project state to x_{k+1} according to (5).

The optimal cost function of this problem and the multiarm bandit problem are identical. The return-to-state argument

[3] Strictly speaking, the value iteration algorithm is near optimal, that is, it yields a value of the Gittins index that is arbitrarily close to the optimal Gittins index. However, for brevity, we will refer to it as optimal.

yields the following result for the HMM multiarm bandit problem.

Result 1 [16]: The Gittins index of the state $x^{(p)}$ of project p is given by

$$\gamma^{(p)}\left(x^{(p)}\right) = V^{(p)}\left(x^{(p)}, x^{(p)}\right)$$

where for $x, \bar{x} \in \mathcal{X}^{(p)}$, $V^{(p)}(x^{(p)}, \bar{x}^{(p)})$ satisfies the dynamic programming functional (Bellman) equation

$$\begin{aligned} V^{(p)}(x^{(p)}, \bar{x}^{(p)}) = \min \Bigg[& R'(p)x^{(p)} \\ & + \beta \sum_{m=1}^{\mathcal{M}_p} V^{(p)}\left(\frac{B^{(p)}(m)A^{(p)\prime}x^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)\prime}x^{(p)}}, \bar{x}^{(p)}\right) \\ & \times \mathbf{1}'B^{(p)}(m)A^{(p)\prime}x^{(p)}, R'(p)\bar{x}^{(p)} \\ & + \beta \sum_{m=1}^{\mathcal{M}_p} V^{(p)}\left(x^{(p)}, \frac{B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)}}\right) \\ & \times \mathbf{1}'B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)} \Bigg]. \end{aligned} \tag{12}$$

C. Value Iteration Approximation and Convergence

The infinite horizon value-function $V^{(p)}(x^{(p)}, \bar{x}^{(p)})$ of (12) can be computed arbitrarily accurately via the following value-iteration algorithm[4] (see [4] for a detailed exposition of these algorithms). The Nth-order approximation is obtained as the following backward dynamic programming recursion:[5]

$$\begin{aligned} & V_{k+1}^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right) \\ & \quad = \min \Bigg[R'(p)x^{(p)} + \beta \sum_{m=1}^{\mathcal{M}_p} V_k^{(p)} \\ & \quad \times \left(\frac{B^{(p)}(m)A^{(p)\prime}x^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)\prime}x^{(p)}}, \bar{x}^{(p)}\right) \\ & \quad \times \mathbf{1}'B^{(p)}(m)A^{(p)\prime}x^{(p)}, R'(p)\bar{x}^{(p)} \\ & \quad + \beta \sum_{m=1}^{\mathcal{M}_p} V_k^{(p)}\left(x^{(p)}, \frac{B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)}}\right) \\ & \quad \times \mathbf{1}'B^{(p)}(m)A^{(p)\prime}\bar{x}^{(p)} \Bigg]; \quad k = 1, 2, \ldots, N \\ & V_0^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right) \\ & \quad = \min\left[R'(p)x^{(p)}, R'(p)\bar{x}^{(p)}\right]. \end{aligned} \tag{13}$$

In the above value iteration algorithm, $V_N(x^{(p)}, \bar{x}^{(p)})$ is the value function of a N-horizon dynamic programming recursion.

The following result about the uniform convergence of the value function $V_N^{(p)}(x^{(p)}, \bar{x}^{(p)})$ computed by the value iteration algorithm (13) is well known; see, for example, [11].

Result 2 [20]: The infinite horizon value function of project p, $V^{(p)}(x^{(p)}, \bar{x}^{(p)})$ defined in (12) can be uniformly approximated arbitrarily closely by a finite horizon value function $V_N^{(p)}(x^{(p)}, \bar{x}^{(p)})$ of (13). In particular, for any $\delta > 0$, there exists a positive integer $\bar{N}$ such that

$$\sup_{x^{(p)}, \bar{x}^{(p)} \in \mathcal{X}^{(p)}} \left| V_{\bar{N}-1}^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right) - V_{\bar{N}}^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right)\right| \le \delta.$$

For this finite horizon $\bar{N}$

$$\sup_{x^{(p)}, \bar{x}^{(p)} \in \mathcal{X}^{(p)}} \left| V_{\bar{N}-1}^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right) - V^{(p)}\left(x^{(p)}, \bar{x}^{(p)}\right)\right| \le \frac{2\beta\delta}{(1-\beta)}.$$

Recall that our primary objective is to compute the Gittins index $\gamma^{(p)}(x^{(p)}) = V^{(p)}(x^{(p)}, x^{(p)})$ from the functional recursion (12). Let $\gamma_N^{(p)}(x^{(p)})$ denote the approximate Gittins index computed via the value iteration algorithm (13), i.e.,

$$\gamma_N^{(p)}\left(x^{(p)}\right) \triangleq V_N^{(p)}\left(x^{(p)}, x^{(p)}\right). \tag{14}$$

For reasons described later in Corollary 1, we will refer to $\gamma_N^{(p)}(x^{(p)})$ as the "near optimal Gittins index."

A straightforward application of the Result 2 shows that the finite horizon Gittins index approximation $\gamma_N^{(p)}(x)$ of (14) can be made arbitrarily accurate by choosing the horizon N sufficiently large. This is summarized in the following corollary.

Corollary 1: The (infinite horizon) Gittins index $\gamma^{(p)}(x^{(p)})$ of state $x^{(p)}$ can be uniformly approximated arbitrarily closely by the near optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$ computed according to (14) for the finite horizon N. In particular, for any $\delta > 0$, there exists a finite horizon $\bar{N}$ such that we have the following.

i) $\sup_{x^{(p)} \in \mathcal{X}^{(p)}} |\gamma_{\bar{N}-1}^{(p)}(x^{(p)}) - \gamma_{\bar{N}}^{(p)}(x^{(p)})| \le \delta$.

ii) For this $\bar{N}$, $\sup_{x^{(p)} \in \mathcal{X}^{(p)}} |\gamma_{\bar{N}-1}^{(p)}(x^{(p)}) - \gamma^{(p)}(x^{(p)})| \le (2\beta\delta/(1-\beta))$.

D. Finite-Dimensional Characterization of Gittins Index

In light of Section II-C, the Gittins index for each project $p \in \{1, 2, \ldots, P\}$ can be computed arbitrarily accurately, providing we can solve the value iteration dynamic programming recursion (13). However, the dynamic programming (13) does not directly translate into practical solution methodologies. The fundamental problem with (13) is that at each iteration k, one needs to compute $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$ over an uncountably infinite set $\mathcal{X}^{(p)}$. The main contribution of this section is to construct a finite-dimensional characterization for the value function $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$, $k = 1, 2, \ldots, N$ and, hence, the near optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$. This finite-dimensional characterization of $\gamma_N^{(p)}(x^{(p)})$ is the main result of this paper. Constructive algorithms based on this finite characterization will be

[4] It is also possible to use the approximate policy iteration algorithm of Sondik [26]. However, there is no available software for implementing this algorithm and its performance is largely untested; see [20, p. 58]

[5] We use k here to denote the iteration number. Indeed, $N-k$ denotes the actual time index of the backward dynamic programming recursion. However, this is unimportant since we are interested in the final term $V_N^{(p)}(x^{(p)}, x^{(p)})$ only.

given in Section II-E to compute the Gittins index for the information states of the original bandit process.

In order to construct a finite characterization of $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$, our first step is to re-express $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$ in terms of a new information state π as a value function $\bar{V}_k^{(p)}$ (which will be defined later). Under this new coordinate basis, we will show that $\bar{V}_k^{(p)}(\pi^{(p)})$ satisfies the dynamic programming recursion of a standard POMDP. It is well known that the value function of a standard POMDP has a finite-dimensional characterization—it is piecewise linear and convex. Thus, we have a finite-dimensional characterization for $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$ and, therefore, for the near-optimal Gittins index $\gamma_k^{(p)}(x^{(p)})$.

Define the following new coordinate basis and parameters, where $\otimes$ denotes tensor (Kronecker) product (here, I denotes the $\mathcal{N}_p \times \mathcal{N}_p$ identity matrix)

$$\begin{aligned} \pi^{(p)} &= x^{(p)} \otimes \bar{x}^{(p)}, \quad \bar{R}_1(p) = R(p) \otimes \mathbf{1} \\ \bar{A}_1^{(p)} &= A^{(p)} \otimes I, \quad \bar{B}_1^{(p)}(m) = B^{(p)}(m) \otimes I \\ \bar{R}_2(p) &= \mathbf{1} \otimes R(p), \quad \bar{A}_2^{(p)} = I \otimes A^{(p)} \\ \bar{B}_2^{(p)}(m) &= I \otimes B^{(p)}(m). \end{aligned} \tag{15}$$

It is easily shown that $\bar{A}_1^{(p)}$, $\bar{A}_2^{(p)}$ are transition probability matrices (their rows add to one and each element is positive), and $\bar{B}_1^{(p)}(m)$, $\bar{B}_2^{(p)}(m)$ are observation probability matrices. In addition, the $\mathcal{N}_p^2$-dimensional vector $\pi^{(p)}$ is an information state since it satisfies

$$\mathbf{1}'\pi^{(p)} = 1, \quad \pi^{(p)}(i) \geq 0, \quad i = 1, 2, \ldots, \mathcal{N}_p^2.$$

Finally, define the control variable $\nu_k \in \{1, 2\}$ at each time k, where ν_k maps π_k to $\{1, 2\}$ at each time k. In terms of the return-to-state formulation of Section II-B, $\nu_k = 1$ means continue, and $\nu_k = 2$ means restart. Define the policy sequence[6] $\nu = (\nu_1, \ldots, \nu_k)$.

The following representation theorem relates to standard POMDPs and is straightforward to establish. Hence, the proof is omitted; see, for example, [4ch. 5.4].

Theorem 1: $\bar{A}_1^{(p)}, \bar{B}_1^{(p)}, \bar{R}(1)$ and $\bar{A}_2^{(p)}, \bar{B}_2^{(p)}, \bar{R}(2)$ defined in (15) form the transition probabilities, observation probabilities, and cost vectors of a two-valued control ($\nu_k \in \{1, 2\}$) POMDP problem with objective

$$\min_{\nu} \mathbf{E}\left[\sum_{k=0}^{N} \beta^k \bar{R}'_{\nu_k}(p)\pi_k\right].$$

Here, the vector $\pi^{(p)}$ is an information state for this POMDP and evolves according to

$$\pi_{k+1}^{(p)} = \frac{\bar{B}_{\nu_k}^{(p)}\left(\bar{A}^{(p)}\right)'_{\nu_k}\pi_k^{(p)}}{\mathbf{1}'\bar{B}_{\nu_k}^{(p)}\left(\bar{A}^{(p)}\right)'_{\nu_k}\pi_k^{(p)}}, \quad \nu_k \in \{1, 2\}$$

depending on the control ν_k chosen at each time instant. Finally, the dynamic programming recursion for optimizing this POMDP over the finite horizon N is given by

$$\begin{aligned} \bar{V}_{k+1}^{(p)}\left(\pi^{(p)}\right) = \min \Bigg[& \bar{R}'_1(p)\pi^{(p)} \\ & + \beta \sum_{m=1}^{\mathcal{M}_p} \bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}\right) \\ & \times \mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}, \bar{R}'_2(p)\pi^{(p)} \\ & + \beta \sum_{m=1}^{\mathcal{M}_p} \bar{V}_k^{(p)}\left(\frac{\bar{B}_2^{(p)}(m)\left(\bar{A}_2^{(p)}\right)'\pi^{(p)}}{\mathbf{1}'\bar{B}_2^{(p)}(m)\left(\bar{A}_2^{(p)}\right)'\pi^{(p)}}\right) \\ & \times \mathbf{1}'\bar{B}_2^{(p)}(m)\left(\bar{A}_2^{(p)}\right)'\pi^{(p)} \Bigg] \\ & \qquad k = 1, 2, \ldots, N \\ \bar{V}_0^{(p)}(\pi) = \min & \left[\bar{R}'_1(p)\pi^{(p)}, \bar{R}'_2(p)\pi^{(p)}\right]. \end{aligned} \tag{16}$$

Here, $\bar{V}_k^{(p)}(\pi^{(p)})$ denotes the value-function of the dynamic program

$$\bar{V}_k^{(p)}(\pi) \triangleq \min \mathbf{E}\left[\sum_{t=N-k}^{N} \beta^t \bar{R}'_{\nu_t}(p)\pi_t \,|\, \pi_{N-k} = \pi\right]. \tag{17}$$

The following is the main result of this paper.

Theorem 2: Under the coordinate basis defined in (15), the following three statements hold.

1) The value function $V_k^{(p)}(x^{(p)}, \bar{x}^{(p)})$ in (16) for computing the Gittins index is identically equal to the value function $\bar{V}_k^{(p)}(\pi^{(p)})$ of the standard POMDP (Lemma 1) at each iteration k.
2) At each iteration k, $k = 0, 1, \ldots, N$, the value function $\bar{V}_k^{(p)}(\pi^{(p)})$ is piecewise linear and convex and has the finite-dimensional representation
$$\bar{V}_k^{(p)}\left(\pi^{(p)}\right) = \min_{\lambda_{i,k} \in \Lambda_k^{(p)}} \lambda'_{i,k}\pi^{(p)}. \tag{18}$$
Here, all the $\mathcal{N}_p^2$-dimensional vectors $\lambda_{i,k}$ belong to a precomputable finite set of vectors $\Lambda_k^{(p)}$.
3) For any information state $x^{(p)} \in \mathcal{X}^{(p)}$ of project p, the near-optimal Gittins index $\gamma_N^{(p)}(x^{(p)})$ is given by the finite-dimensional representation
$$\gamma_N^{(p)}\left(x^{(p)}\right) = \min_{\lambda_{i,N} \in \Lambda_N^{(p)}} \lambda'_{i,N}\left(x^{(p)} \otimes x^{(p)}\right). \tag{19}$$

Remark: Statement 1 of the above theorem shows that the value iteration algorithm (13) for computing the Gittins index $\gamma_k^{(p)}(x^{(p)})$ is identical to the dynamic programming recursion (16) for optimizing a standard finite-horizon POMDP. Statement 2 says that the finite-horizon POMDP has a finite-dimensional piecewise linear solution characterized by

[6] Note that policy ν is used to compute the Gittins index of a given target p. It is not to be confused with the policy μ defined in Section I-A, which determines which target to observe

a precomputable finite set of vectors at each time instant. Statement 2 is not new—it is well known in the POMDP literature, e.g., see [20] and [25]. Therefore, we only present a short proof of Statement 2. Moreover, there are several linear programming-based algorithms available for computing the finite set of vectors $\Lambda_k^{(p)}$ at each iteration k; further details are given in Section II-E.

Finally, Statement 3 gives us the answer to computing the Gittins index for the HMM multiarm bandit problem. Recall that the information state $x^{(p)}$ is merely the filtered estimate computed by the pth HMM filter. Given that we can compute the set of vectors $\Lambda_N^{(p)}$, (19) gives us an explicit expression for the Gittins index $\gamma_N^{(p)}(x^{(p)})$. To our knowledge, this result has not appeared elsewhere in the operations research or stochastic control literature. ○

Proof: The proof of the first statement is by mathematical induction. At iteration $k = 0$

$$\begin{aligned}\bar{V}_0^{(p)}(\pi) &= \min\left[\bar{R}_1'(p)\pi^{(p)}, \bar{R}_2'(p)\pi^{(p)}\right]\\ &= \min\Big[(R(p)\otimes\mathbf{1})'\left(x^{(p)}\otimes\bar{x}^{(p)}\right)\\ &\qquad (\mathbf{1}\otimes R(p))'\left(x^{(p)}\otimes\bar{x}^{(p)}\right)\Big] \quad \text{(by (15))}\\ &= \min\Big[\left(R(p)'x^{(p)}\right)\otimes\left(\mathbf{1}'\bar{x}^{(p)}\right)\\ &\qquad \left(R(p)'\bar{x}^{(p)}\right)\otimes\left(\mathbf{1}'x^{(p)}\right)\Big]\\ &\qquad \text{(distributive property of Tensor products)}\\ &= \min\left[R(p)'x^{(p)}, R(p)'\bar{x}^{(p)}\right]\\ &\qquad \text{(since } x \text{ and } \bar{x} \text{ are information states)}\\ &= V_0^{(p)}(x,\bar{x}).\end{aligned}$$

Assume that at time k, $\bar{V}_k^{(p)}(\pi) = V_k^{(p)}(x,\bar{x})$, and consider (16). We have already shown earlier that $\bar{R}_1'(p)\pi^{(p)} = R(p)'x^{(p)}$, and $\bar{R}_2'(p)\pi^{(p)} = R(p)'\bar{x}^{(p)}$ Thus, to prove that $\bar{V}_{k+1}(\pi) = V_{k+1}^{(p)}(x,\bar{x})$, it only remains to be shown that

$$\begin{aligned}&\bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}\right)\mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}\\ &\quad = V_k^{(p)}\left(\frac{B^{(p)}(m)A^{(p)'}x^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}}, \bar{x}^{(p)}\right)\\ &\qquad \times \mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}.\end{aligned}$$

Consider the numerator and denominator terms on the left-hand side of the above equation

$$\begin{aligned}&\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}\\ &\quad = \left(B^{(p)}(m)\otimes I\right)\left(A^{(p)}\otimes I\right)'\left(x^{(p)}\otimes\bar{x}^{(p)}\right)\\ &\quad = \left(B^{(p)}(m)A^{(p)'}x^{(p)}\right)\otimes\bar{x}^{(p)}\\ &\mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}\\ &\quad = (\mathbf{1}\otimes\mathbf{1})'\left(B^{(p)}(m)A^{(p)'}x^{(p)}\right)\otimes\bar{x}^{(p)}\\ &\quad = \mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}.\end{aligned}$$

Thus

$$\begin{aligned}&\bar{V}_k^{(p)}\left(\frac{\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}{\mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}}\right)\\ &\quad \times \mathbf{1}'\bar{B}_1^{(p)}(m)\left(\bar{A}_1^{(p)}\right)'\pi^{(p)}\\ &\qquad = \bar{V}_k^{(p)}\left(\frac{B^{(p)}(m)A^{(p)'}x^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}}\otimes\bar{x}^{(p)}\right)\\ &\qquad\quad \times \mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}\\ &\qquad = V_k^{(p)}\left(\frac{B^{(p)}(m)A^{(p)'}x^{(p)}}{\mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}}, \bar{x}^{(p)}\right)\\ &\qquad\quad \times \mathbf{1}'B^{(p)}(m)A^{(p)'}x^{(p)}\end{aligned}$$

which proves that $\bar{V}_{k+1}^{(p)}(\pi) = V_{k+1}^{(p)}(x,\bar{x})$.

The second statement (finite-dimensional characterization) has an inductive proof (see [20] for details). At iteration $k = 0$, from (16), $\bar{V}_0(\pi) = \min[\bar{R}_1'(p)\pi^{(p)}, \bar{R}_2'(p)\pi^{(p)}]$ and is of the form (18). Assume at iteration k that $\bar{V}_k(\pi) = \min_{\lambda_{i,k}\in\Lambda_k^{(p)}}\lambda_{i,k}'\pi$. Then, substituting this in (16) we have (20), shown at the bottom of the page, where the vector $\lambda_{i^*(j,m,\pi),k}$ is defined as

$$\lambda_{i^*(j,m,\pi),k} \triangleq \arg\min_{\lambda_{i,k}\in\Lambda_k}\lambda_{i,k}'\bar{B}_j^{(p)}(m)\left(\bar{A}_j^{(p)}\right)'\pi. \tag{21}$$

Clearly, $\bar{V}_{k+1}(\pi)$ of (20) is of the form (18). Finally, (19) follows directly from (14) and (18). □

E. Optimal Algorithm

Given the finite-dimensional representation of the Gittins index in (19) of Theorem 2, there are several linear programming based algorithms in the POMDP literature such as

$$\begin{aligned}\bar{V}_{k+1}(\pi) &= \min_{j\in\{1,2\}}\left[\bar{R}_j'(p)\pi + \beta\sum_{m=1}^{\mathcal{M}_p}\min_{\lambda_{i,k}\in\Lambda_k}\lambda_{i,k}'\bar{B}_j^{(p)}(m)\left(\bar{A}_j^{(p)}\right)'\pi\right]\\ &= \min_{j\in\{1,2\}}\left[\bar{R}_j'(p) + \beta\sum_{m=1}^{\mathcal{M}_p}\lambda_{i^*(j,m,\pi),k}'\bar{B}_j^{(p)}(m)\left(\bar{A}_j^{(p)}\right)'\right]\pi\end{aligned} \tag{20}$$

Sondik's algorithm [25], Monahan's algorithm [21], Cheng's algorithm [20], and the Witness algorithm [8] that can be used to compute the finite set of vectors $\Lambda_N^{(p)}$ depicted in (18). See the website in [9] for an excellent tutorial exposition with graphics of these various algorithms. Any of these algorithms will equally well, although not with the same computational efficiency, produce the desired solution set of vectors $\Lambda_N^{(p)}$. In the numerical examples presented in Section IV, we used the "incremental-prune" algorithm that was recently developed in the artificial intelligence community by Cassandra *et al.* in 1997 [10].

1) Computational Complexity: Computing the Gittins index for each target p of the HMM multiarm bandit problem involves off-line computation of $\Lambda_N^{(p)}$. It is shown in [22] that solving a standard POMDP is PSPACE complete (which includes the class of NP hard problems), i.e., involves exponential (worst-case) complexity in $\mathcal{M}_p$. It has recently been demonstrated in [8] that for general POMDP problems with $\mathcal{M}_p$ up to 10, $\Lambda_N^{(p)}$ can be computed in several hundreds of minutes. In the numerical examples of Section IV, $\Lambda_{100}^{(p)}$ was computed for $\mathcal{N}_p = \mathcal{M}_p$ up to 4 on a Pentium II personal computer in less than 1 min. One of the advantages of the multiarm bandit formulation is that computational complexity only increases linearly with the number of targets. Thus, a problem with $P = 100$ targets where $\mathcal{N}_p = \mathcal{M}_p = 2$ for each target can easily be solved in the multiarm bandit formulation—whereas without the multiarm bandit assumption, the POMDP would have 2^{100} states and is clearly impossibly difficult to solve. See [19], where a suboptimal algorithm is proposed that approximates the piecewise linear value function; moreover, computable bounds on the approximation are presented.

2) Finitely Transient Policies: What happens to the set of vectors $\Lambda_N^{(p)}$ as $N \to \infty$ (i.e., if the value iteration algorithm is run for a large number of iterations N)? Depending on the parameters $A^{(p)}, B^{(p)}, R^{(p)}, \beta$, there are two possibilities.

1) The number of vectors in $\Lambda_N^{(p)}$ can grow exponentially to infinity as N increases to infinity so that the piecewise linear $V_N(\pi)$ converges to some continuous function $V(\pi)$, which is no longer piecewise linear. In such cases, computing $\Lambda_N^{(p)}$ for large N can be prohibitively expensive.
2) The number of vectors in $\Lambda_N^{(p)}$ remain finite as N increases to infinity. In particular, in some cases, there exists a horizon N_0, after which, the number of vectors in Λ_N, $N \geq N_0$ remain a constant. Such an optimal policy is called a *finitely transient* policy and was first introduced by Sondik [26]. Such cases are of great interest because the optimal policy can be computed arbitrarily accurately without significant computational cost. In the numerical examples of Section IV, we will present examples of multiarm bandit problems with finitely transient policies.

See [26] and [8] for details, precise definitions, and extensions of finitely transient policies. Although Sondik [26] presents an algebraic procedure to determine if a given policy is finitely transient, to date, there is no systematic procedure that says if the *optimal* policy of an arbitrary infinite horizon POMDP problem is finitely transient.

F. Suboptimal Algorithms

For large size state spaces, the optimal algorithm can have prohibitively large computational and memory requirements. In such cases, it is necessary to develop suboptimal algorithms that compute the approximate Gittins index of a HMM. An obvious brute-force suboptimal solution is to form a finite fixed grid discretization of the continuous information state $\mathcal{X}^{(p)}$ and solve the resulting finite finite-dimensional dynamic programming recursion. However, to obtain even moderately good scheduling policies, very large grids that involve enormous computational and memory requirements are required. We present three suboptimal algorithms to compute the Gittins index.

1) Coarse Resolution Algorithm: A numerical implementation of the optimal procedure outlined above requires specification of the accuracy to which the vectors in $\Lambda_k^{(p)}$ are computed for each iteration k. This means that any two vectors λ_1, λ_2 in $\Lambda_k^{(p)}$ are such that $\|\lambda_1 - \lambda_2\|_2 > \epsilon$, where $\epsilon > 0$ denotes the resolution tolerance. A natural way of significantly reducing the computational complexity of the optimal algorithm is to increase ϵ and, hence, significantly decrease the number of vectors in $\Lambda_k^{(p)}$. We call the resulting suboptimal algorithm the "coarse resolution algorithm." The coarse resolution algorithm is available as an option in the POMDP software program, which can be downloaded from [9]. Numerical examples presented in Section IV show that choosing $\epsilon = \min_{s,u} R(s,u)/5$ still gives sensor schedules that are very close to the optimal solution.

The second and third suboptimal algorithms presented below compute approximate Gittins index of the HMM in terms of the Gittins index of the underlying finite state Markov chain $s_k^{(p)}$. They are based on the principle of "certainty equivalence," which is widely used in stochastic control to design suboptimal controllers; see [4, ch. 6] for a general exposition. Certainty equivalence assumes that the state estimate $x_k^{(p)}$ is perfect at each time instant, i.e., exactly equal to underlying Markov chain state $s_k^{(p)}$. The scheduler is then designed for the Markov chains as follows.

Let $\bar{\gamma}^{(p)}(i)$ denote the Gittins index of the ith state of the underlying finite state Markov decision process. There are a several fast algorithms that can be used to compute the Gittins index $\bar{\gamma}^{(p)}(i)$ of the finite state Markov decision process, for example, the return-to-state-x_0 [16], Varaiya *et al.* algorithm [29], and Bertsimas and Nino-Mora polymatroid algorithm [5]. In terms of $\bar{\gamma}^{(p)}(i)$, we can compute the following two approximate Gittins indices for the HMM multiarm bandit problem.

2) Conditional Mean (CM) Gittins Index: Given the information state (HMM filtered density) $x_k^{(p)}$ of target p at time k, the CM Gittins index of the HMM is defined as

$$\begin{aligned}\gamma_{\mathrm{CM}}^{(p)}(x^{(p)}) &\triangleq \mathbf{E}\left[\gamma^{(p)}\left(x_k^{(p)}\right) \mid Y_k, U_{k-1}\right] \\ &= \sum_i \bar{\gamma}^{(p)}(i) x_k^{(p)}(i). \qquad (22)\end{aligned}$$

Note that $\gamma_{\mathrm{CM}}^{(p)}(x^{(p)})$ is merely the conditional expectation of the underlying finite-state Markov chain's Gittins index, given the observation and scheduling history.

3) Maximum a Posteriori (MAP) Gittins Index: Given the information state $x_k^{(p)}$ at time k, the MAP Gittins index

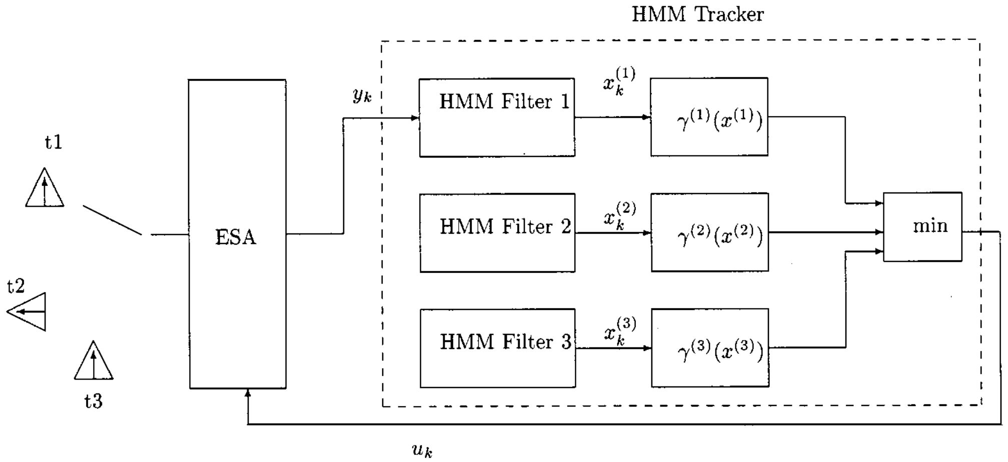


Fig. 2. Optimal solution to beam scheduling for multitarget tracking.

denoted as $\gamma^{(p)}_{\mathrm{MAP}}(x^{(p)})$ is the Gittins index of the underlying Markov chain state that is closest to $x_k^{(p)}$. That is, $\gamma^{(p)}_{\mathrm{MAP}}(x^{(p)}) = \bar{\gamma}^{(p)}(i^*)$, where $i^* = \max_i x^{(p)}(i)$ denotes the MAP state estimate for the pth target.

4) Computational Complexity: The Gittins index $\bar{\gamma}^{(p)}(i)$ of the underlying Markov decision process can be computed with $O(P\mathcal{N}_p^3)$ computational complexity [16], where P is the number of targets, and $\mathcal{N}_p$ is the number of states per target. Varaiya *et al.* and the Bertsimas and Nino-Mora polymatroid algorithm both run in $O(P\mathcal{N}_p^3)$ time [5], [16].

III. Beam Scheduling for ESA Tracking Systems

Given the complete solution of the HMM multiarm bandit problem in the previous section, we now return to the agile beam multitarget tracking problem discussed in Section I-C.

A. Summary of Optimal Beam Scheduling Algorithm

Fig. 2 shows the setup and optimal solution for the case where there are three targets. The HMM tracker consists of three HMM state filters. Suppose that target 1 is the optimal target to direct the beam toward at time $k-1$, i.e., $u_{k-1} = 1$. The HMM filter 1 receives noisy measurements of the state of target 1 and computes the filtered density (information state) $x_k^{(1)}$ of the target state according to (5). The corresponding Gittins index of this state is computed using (19). For the other targets, no observations are received, and their information states remain unchanged $(x_k^{(2)} = x_{k-1}^{(2)}, x_k^{(3)} = x_{k-1}^{(3)})$; hence, the Gittins indices $\gamma^{(2)}(x_k^{(2)}), \gamma^{(2)}(x_k^{(3)})$ remain unchanged. The Gittins indices of the states at time k of the three targets are then compared. The multiarm bandit theory then specifies that the optimal choice u_k at time k is to direct the beam toward the target with the smallest Gittins index, as shown in Fig. 2.

B. Beam Scheduling With Hybrid Sensor

Before detailing the optimal beam scheduling algorithm, we consider beam scheduling with a more sophisticated hybrid tracking sensor. The hybrid sensor consists of two entities: a fast-switching-tracker (FST), which can only track one target at a time, and a slow-position-identifier (SPI), which computes the "best" estimate of the states of all targets at periodic intervals. The periodic times at which the SPI produces the state estimates are called the *reinitialization times*.

The FST and the SPI communicate in the following way. After some reinitialization time, the SPI gives the FST the best estimates of the position of all the targets, and the FST reinitializes its values of the coordinates of all the targets and resumes tracking. This way, tracking accuracy is improved; see also the discussion in Section I-C. Let $T > 0$ denote the periodic reinitialization interval. At the reinitialization times $k = nT$, $n = 1, 2, \ldots$, the target-dependent costs $W(p), p = 1, \ldots, P$ [which are defined in (10)] for each target can be adjusted to reflect the most recent information obtained by the SPI and thereby improve tracking accuracy. Let $W_{nT}(p)$ denote the updated target dependent costs at the reinitialization times nT.

This update in the costs changes the Gittins index for each target. However, because the updated costs are assumed state independent, there is a simple way of updating the Gittins index as follows.

Let $\gamma_N^{(p)}(x^{(p)})$ denote the Gittins index for target p when the cost vector is $R(p)$. Let $R_{\mathrm{updated}}(p)$ denote the updated cost, i.e., $R_{\mathrm{updated}}(p) = R(p) + W_{nT}(p)\mathbf{1}$. Finally, let $\gamma^{(p)}_{N,\mathrm{updated}}(x^{(p)})$ denote the Gittins index corresponding to the updated cost $R_{\mathrm{updated}}(p)$. Then, we have the following result.

Theorem 3: The updated cost Gittins index $\gamma^{(p)}_{N,\mathrm{updated}}(x^{(p)})$ for target p can be computed as

$$\gamma^{(p)}_{N,\mathrm{updated}}(x^{(p)}) = \gamma_N^{(p)}\left(x^{(p)}\right) + W_{nT}(p)\frac{1-\beta^{N+1}}{1-\beta}. \quad (23)$$

Proof: The proof follows from the definition of the value-function $\bar{V}_N^{(p)}(\pi)$ in (17). From (17), the value function for the dynamic program with updated cost is

$$\begin{aligned}\bar{V}^{(p)}_{N,\mathrm{updated}}(\pi) &= \min_{\nu} \mathbf{E}\left[\sum_{t=0}^{N} \beta^t \bar{R}'_{\nu_t,\mathrm{updated}}(p)\pi_t \mid \pi_0 = \pi\right]\\ &= \sum_{t=0}^{\infty} \beta^t W_{nT}(p)\\ &\quad + \min_{\nu} \mathbf{E}\left[\sum_{t=0}^{N} \beta^t \bar{R}'_{\nu_t}(p)\pi_t \mid \pi_0 = \pi\right]\\ &= W_{nT}(p)\frac{1-\beta^{N+1}}{1-\beta} + \bar{V}_N^{(p)}(\pi). \qquad \square\end{aligned}$$

This is a useful result because it allows freedom in changing the constant target reward (every reinitialization time) without adding a large computational burden.

The complete optimal beam scheduling algorithm is given in Algorithm 1.

```
Algorithm 1: Algorithm for Real-Time Beam Scheduling
Input for each target $p = 1, \ldots, P$:
    $A^{(p)}$ {Transition probability matrix}, $B^{(p)}$ {Observation matrix}, $R(p)$ {Cost vector},
    $x_0^{(p)}$ {A priori state estimate at time 0}, $N$ {Horizon size (large)}, $\beta$ {discount factor}
Off-line Computation of Gittins indices: Compute finite set of vectors $\Lambda_N^{(p)}$
for $p = 1, \ldots, P$ do
    compute $\Lambda_N^{(p)}$ according to Section II-E
end
{Initialization at time $k = 0$}
    compute $\gamma_N^{(p)}(x_0^{(p)}) = \min_{\lambda_{i,N} \in \Lambda_N^{(p)}} \lambda'_{i,N}(x_0^{(p)} \otimes x_0^{(p)})$ according to (19).
Real Time Beam Scheduling over tracking horizon $T_{\max}$
while time $k < T_{\max}$ do
    {Track the target $q$ with largest Gittins index using HMM filter}
        Steer beam toward target $q = \min_{p \in \{1,\ldots,P\}} \{\gamma^{(p)}(x_k^{(p)})\}$ (see (11))
        Obtain noisy measurement $y_{k+1}^{(q)}$
        Update estimate of $q$th target's coordinates using the HMM filter (5)
            $$x_{k+1}^{(q)} = \frac{B^{(q)}(m) A^{(q)\prime} x_k^{(q)}}{\mathbf{1}' B^{(q)}(m) A^{(q)\prime} x_k^{(q)}}$$
        $\gamma_N^{(q)}(x_{k+1}^{(q)}) = \min_{\lambda_{i,N} \in \Lambda_N^{(q)}} \lambda'_{i,N}(x_{k+1}^{(q)} \otimes x_{k+1}^{(q)})$ according to (19)
    {For other $P-1$ targets $p = 1, \ldots, P, p \neq q$, state estimates remain unchanged}
        $\gamma_N^{p}(x_{k+1}^{(p)}) = \gamma_N^{(p)}(x_k^{(p)})$
    if $k = nT$ (re-initialization time) then begin
        {Compute the estimated positions of all the targets}
        run SPI
        {Optional: Modify the constant target reward $W_{nT}(p)$ and update Gittins indices}
        for $p = 1, \ldots, P$ do
            $\gamma_N^{(p)}(x_k^{(p)}) = (W_{nT}(p))/(1-\beta) + \gamma_N^{(p)}(x_k^{(p)})$ according to (23).
        end
        $n = n + 1$
    end
    $k = k + 1$
end.
```

1) Real-Time Computation Complexity: Given that the vector set $\Lambda_N^{(p)}$ is computed off-line, the real-time computations required in the above algorithm at each time k are

i) computing the HMM estimate $x_{k+1}^{(q}$) (5), which involves $O(\mathcal{N}_p^2)$ computations;

ii) computing $\gamma_N^{(q)}(x_{k+1}^{(q)})$, where (19) requires $O(|\Lambda_N^{(p)}|\mathcal{N}_p^2)$ computations.

If the suboptimal CM and MAP algorithms of Section II-F are used, then i) remains unchanged, whereas ii) requires $O(\mathcal{N}_p^2)$ computations.

IV. Numerical Examples

In this section, we present numerical examples that illustrate the performance of the optimal and suboptimal beam scheduling algorithms presented in Sections II-E and F. When each target evolves according to a two-state or three-state Markov chain, the Gittins index of each target can be graphically illustrated, meaning that a complete discussion of the algorithm behavior can be given. For this reason, Section IV-A deals with two-state Markov chains, and Section IV-B deals with three-state Markov chains. For higher state examples, while the optimal and suboptimal beam schedules can be computed, it is not possible to visualize the Gittins indices.

A. Two-State Example

1) Parameters: The scenario involves three $(P = 3)$) moving targets (e.g., aircraft). Each target $s_k^{(p)}$, $p \in \{1, 2, 3\}$ is modeled as a two-state Markov chain, i.e., $\mathcal{N}_p = 2$. For example, in the modal estimation case, these states could denote the orientation of the target. Alternatively, in the beam scheduling case, the states could be the distance of the target from the base station quantized into $s_k^{(p)} \in \{\texttt{near, far}\}$.

The following parameters were chosen for the three targets: $\beta = 0.9$

$$\text{Target 1: } A^{(1)} = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix} \quad B^{(1)} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}$$
$$R(1) = \begin{bmatrix} -14 \\ -3 \end{bmatrix} \quad x_0^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\text{Target 2: } A^{(2)} = \begin{bmatrix} 0.7 & 0.3 \\ 0.6 & 0.4 \end{bmatrix} \quad B^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$R(2) = \begin{bmatrix} -4 \\ -15 \end{bmatrix} \quad x_0^{(2)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\text{Target 3: } A^{(3)} = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix} \quad B^{(3)} = \begin{bmatrix} \alpha & 1-\alpha \\ 0 & 1 \end{bmatrix}$$
$$R(3) = \begin{bmatrix} 0 \\ -18 \end{bmatrix} \quad x_0^{(3)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{24}$$

Here, $0 \leq \alpha \leq 1$ is a parameter that we will vary to illustrate the performance of the various algorithms. Notice that target 1 is an HMM with both states observed in noise and target 2 is a fully observed Markov chain (i.e., its state is perfectly observed), whereas target 3 has one state perfectly observed and the other state observed in noise with probability of detection α.

Procedure: The various steps of the optimal beam scheduling Algorithm 1 in Section III-B were implemented as follows.

1) *Off-Line Computation of Gittins Index*: We computed the optimal Gittins indices of the three targets as follows: We

TABLE I
COMPARISON OF OPTIMAL ALGORITHM (TOLERANCE $\epsilon = 10^{-10}$) WITH COARSE RESOLUTION SUBOPTIMAL ALGORITHMS (TOLERANCES $\epsilon = 1$ AND 10). THE NUMBER OF VECTORS IN $\Lambda_N^{(3)}$ FOR EACH ϵ ARE ALSO SHOWN

α	Optimal $\epsilon = 10^{-10}$		Coarse $\epsilon = 1$		Coarse $\epsilon = 10$	
	Cost	# vectors	Cost	# vectors	cost	#vectors
0	-98.5	2	-98.5	2	-84.1	2
0.2	-98.5	58	-98.5	5	-83.9	2
0.4	-96.5	53	-96.0	8	-84	2
0.6	-105.4	45	-105.1	8	-84.04	2
0.8	-93.2	51	-93.2	6	-83.7	2
1.0	-83	4	-83	4	-82.9	2

used the POMDP program downloaded from the website [9] to compute the set of vectors $\Lambda_N^{(p)}$, $p = 1, 2, 3$. All our simulations were run on a Pentium 2 400-MHz personal computer. The POMDP program allows the user to choose from several available algorithms. We used the "Incremental Pruning" algorithm developed by Cassandra *et al.* in 1997 [10]. This is currently one of the fastest known algorithms for solving POMDPs; see [8] for details. For each of the three targets, the POMDP program (value iteration algorithm) was run on a horizon N such that $\sup_x |\gamma_N^{(p)}(x) - \gamma^{(p)}(x)| < 10^{-10}$. (Clearly, this is far more accurate than required; see the suboptimal algorithm below. However, the computation time required to compute $\Lambda_N^{(p)}$ was less than a minute.)

2) *Real-Time Beam Scheduling*: After computing $\Lambda_N^{(p)}$ as described above, the HMM tracker was implemented as outlined in Algorithm 1 of Section III-B. The *a priori* estimates of the various targets $x_0^{(p)}$ were chosen as in (24). The tracking horizon T was chosen as 50.

For comparison, the three suboptimal scheduling algorithms outlined in Section II-F were simulated, i.e., the conditional mean Gittins index, MAP Gittins index, and coarse resolution algorithms. In addition, a fourth periodic suboptimal scheduling algorithm was implemented where the beam was periodically switched between the three targets, i.e., $(u_1, u_2, u_3, u_4, \ldots) = (1, 2, 3, 1, \ldots)$.

Results: The POMDP program output reveals that the value function for target 1 $\Lambda_N^{(1)}$ is comprised of 12 vectors as $N \rightarrow \infty$, meaning that $V_N^{(1)}(\pi)$ is finitely transient (see Section II-E). Since target 2 is completely observed, its Gittins index is straightforwardly computed. Finally, for target 3, $\bar{V}_k^{(3)}(\pi)$ is finitely transient with $\lim_{N\rightarrow\infty} \Lambda_N^{(3)}$ comprising of a finite number of vectors, depending on the probability of detection α; see Table I.

The performance of the optimal algorithm for each $\alpha \in [0, 1]$ was measured in terms of the average cost incurred in tracking the three targets over the tracking horizon $T = 50$. For each value of α, the average cost was computed by carrying out 1000 independent simulations of the HMM beam scheduling algorithm.

Table I compares the performance of the optimal beam scheduling algorithm with the coarse resolution algorithm of Section II-F. The optimal algorithm uses a resolution of $\epsilon = 10^{-10}$ for computing $\Lambda^{(p)}$. (This is a default setting of the parameter `epsilon` in the POMDP program). The coarse resolution algorithms use $\epsilon = 1$ and $\epsilon = 10$, respectively. It can be seen from Table I that the performance when choosing $\epsilon = 1$ is almost identical to the optimal algorithm, despite the fact that the number of vectors in $\Lambda_N^{(3)}$ are significantly reduced.

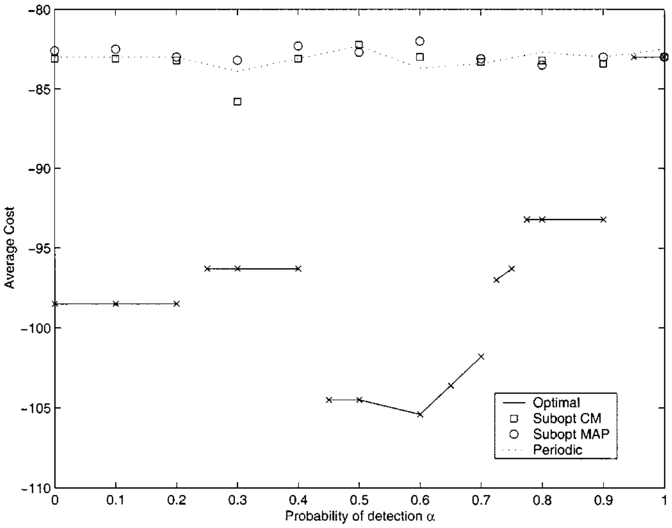


Fig. 3. Performance comparison of optimal and suboptimal beam scheduling algorithms. The optimal algorithm's performance consists of sixsegments, depending on the probability of detection α.

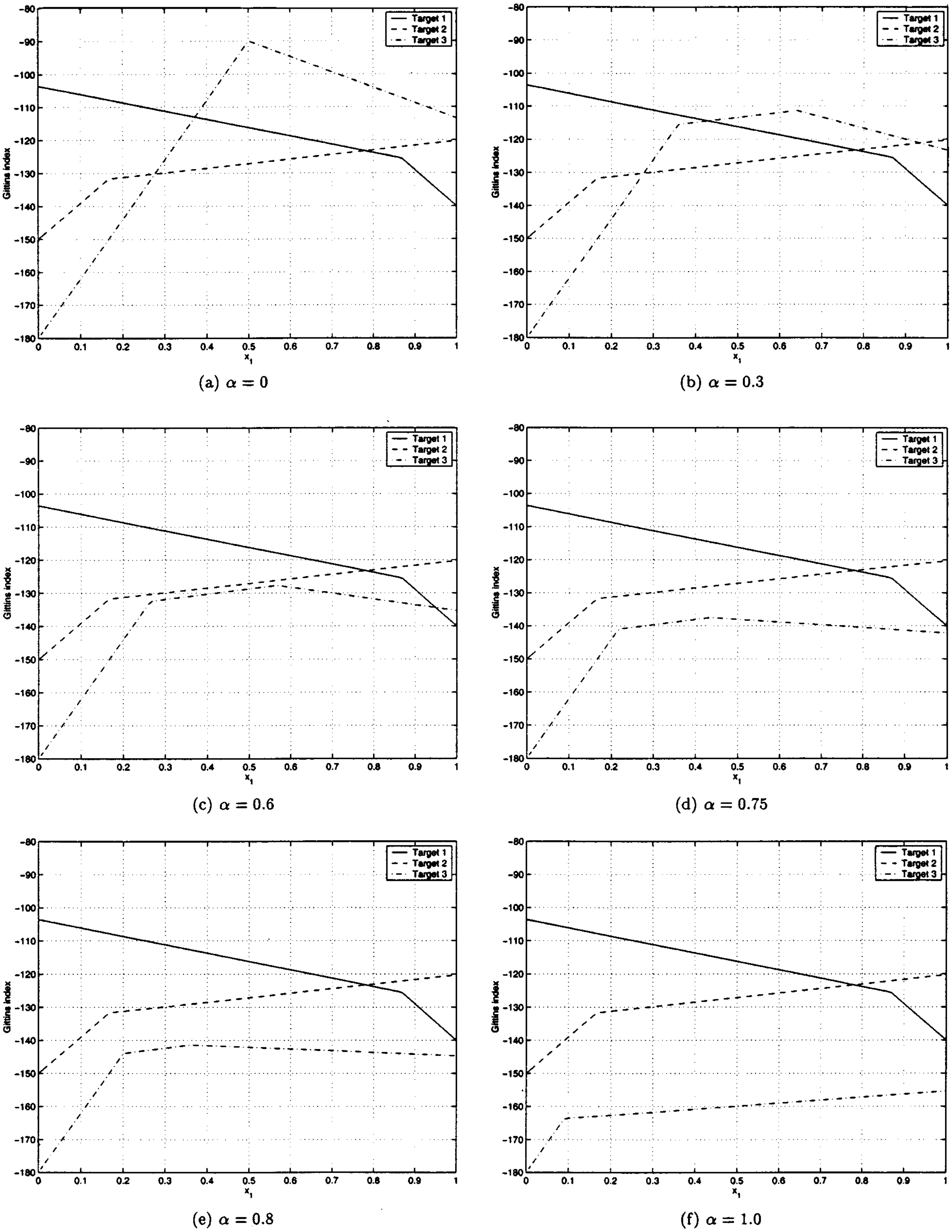
Fig. 3 compares the the performance of the optimal beam scheduling algorithm with three other sub-optimal algorithms.

1) Conditional mean Gittins index denoted by "Subopt CM;"
2) MAP Gittins index denoted by "Subopt MAP;"
3) periodic scheduling algorithm denoted by "Periodic."

The periodic schedule is a data-independent deterministic sequence and serves as a sensible upper bound to the minimum cost.

Discussion: As can be seen in Fig. 3, the performance of the optimal algorithm consists of six regions. This behavior can be explained in terms of the Gittins indices of the three targets. In Fig. 4, we plot the Gittins indices $\gamma^{(p)}(x)$ of the three targets versus the information state x for values of α in the six regions of the interval $[0, 1]$. Because x consists of two components that add to one, it suffices to plot the Gittins index versus the first component of x, which is denoted as x_1 in Figs. 3 and 4.

1) As shown in Fig. 4(a), for $\alpha \in [0,\ 0.2]$, $\gamma^{(3)}(x)$ consists of two line segments. In this region, the average beam scheduling cost is 98.5; see Fig. 3. In addition, due to the choice of *a priori* estimates $x_0^{(p)}$ in (24), it is clear from Fig. 4(a) that the optimal target to track at time $k = 0$ is target 2, i.e., $u_0 = 2$. Because target 2 is fully observed, its information state $x_k^{(2)}$ takes on only two possible values: $[1\ \ 0]'$ and $[0\ \ 1]'$. As soon as the Markov chain $s_k^{(2)}$ jumps state so that $x_k^{(2)} = [1\ \ 0]'$, target 1 becomes the optimal target to track.
2) As shown in Fig. 4(b), for $\alpha \in (0.2, 0.4]$, $\gamma^{(3)}(x)$ consists of three line segments. In this region, the average beam scheduling cost is 96.3



(a) $\alpha = 0$

(b) $\alpha = 0.3$

(c) $\alpha = 0.6$

(d) $\alpha = 0.75$

(e) $\alpha = 0.8$

(f) $\alpha = 1.0$

Fig. 4. Gittins indices $\gamma^{(p)}(x)$ of the three targets specified in (24) plotted versus first component of information state x_1. The six subfigures show the variation of $\gamma^{(3)}(x)$ with probability of detection α.

3) As shown in Fig. 4(c), for $\alpha \in (0.4, 0.7]$, $\gamma^{(3)}(x)$ no longer intersects $\gamma^{(2)}(x)$, meaning that if targets 2 and 3 have similar information states (estimates) at a given time, it is optimal to track target 3.

4) As shown in Fig. 4(d), for $\alpha \in (0.7, 0.775)$, $\gamma^{(3)}(x) < \gamma^{(1)}(x)$ at $x = [1\ \ 0]$. Given the initial *a priori* estimates in (24), this means that as soon as target 2 jumps state, the optimal target to track is target 3 (cf. Case 1).

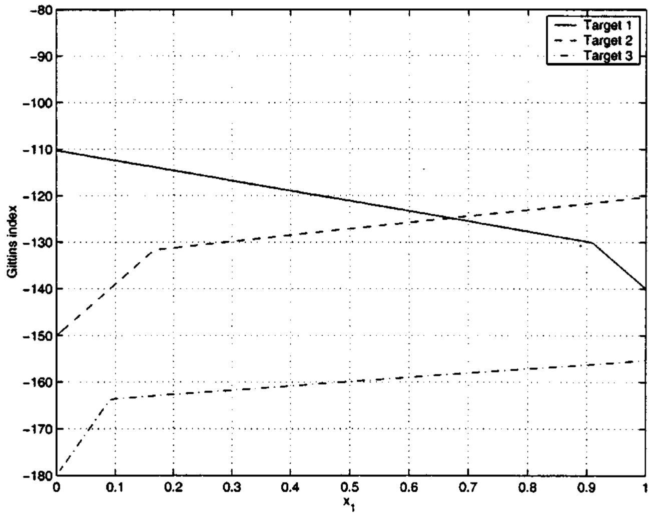


Fig. 5. Gittins index of underlying finite-state Markov chain. These are used to compute the suboptimal beam scheduling policy using the CM and MAP algorithms of Section II-F.

TABLE II
COMPARISON OF OPTIMAL ALGORITHM (TOLERANCE $\epsilon = 10^{-5}$) WITH THE FOLLOWING SUBOPTIMAL ALGORITHMS: COARSE RESOLUTION (TOLERANCE 10^{-1}), CM, MAP, AND PERIODIC SCHEDULER

α	Optimal $\epsilon = 10^{-5}$		Coarse $\epsilon = 10^{-1}$		CM	MAP	Periodic
	Cost	# vectors	Cost	# vectors			
0	13.89	53	13.92	3	14.42	14.58	15.86
0.1	13.90	147	13.98	3	14.42	14.93	15.85
0.3	13.91	213	14.03	3	14.20	14.60	15.86
0.5	13.94	213	14.10	3	14.10	14.65	15.86
0.7	13.83	202	13.99	3	14.09	14.42	15.83
0.9	13.89	90	14.04	3	14.30	14.50	15.83
1.0	13.89	4	13.91	3	13.89	13.9	15.83

5) As shown in Fig. 4(e), for $\alpha \in [0.775, 0.92)$, $\gamma^{(3)}(x) < \gamma^{(2)}(x)$ for all x. This means that target 1 will never be tracked since it is no longer profitable.

6) As shown in Fig. 4(f), for $\alpha \in [0.92, 1]$, $\gamma^{(3)}(x)$ is smaller than $\gamma^{(2)}(x)$ and $\gamma^{(1)}(x)$ for all x. This means that the optimal policy is only to track target 3 and completely ignore targets 1 and 2.

Finally, we comment on why the suboptimal algorithms perform comparably with the optimal algorithm when the probability of detection $\alpha \to 1$. Fig. 5 shows the Gittins index of the underlying finite-state Markov chains of the three targets (i.e., if the Markov chain state was exactly observed), which is used in the suboptimal CM and MAP algorithm. In this fully observed case, x can only be either $[1\ 0]'$ or $[0\ 1]$. It is clear that $\gamma^{(3)}(x)$ is greater than $\gamma^{(2)}(x)$ and $\gamma^{(1)}(x)$ for all x. This means that for the parameters specified in (24), the suboptimal MAP and CM algorithms will only track target 3, completely ignoring targets 1 and 2. This is identical to Case vi) in the HMM multiarm bandit problem, where $\alpha > 0.95$.

B. Multiple Target Tracking With Single Intelligent Sensor

1) Parameters: We consider three targets. The state space for each target comprises of how far it is currently from the base station discretized into 3 distinct distances $d_1 = 0.1$, $d_2 = 0.2$, $d_3 = 1$. As described in Section I-C, the aim is to determine which target to track at each time instant.

The transition probabilities of the three targets are modeled such that in one unit of time, it is impossible for the distance of the target to increase or to decrease by more than one discrete location. The transition probabilities and initial *a priori* state probabilities chosen are

$$A^{(1)} = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.4 & 0.6 \end{bmatrix}, \quad A^{(2)} = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$
$$A^{(3)} = \begin{bmatrix} 0.7 & 0.3 & 0 \\ 0.3 & 0.4 & 0.3 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \quad x_0^{(1)} = [0.3 \quad 0.5 \quad 0.2]'$$
$$x_0^{(2)} = [0.5 \quad 0 \quad 0.5]', \quad x_0^{(3)} = [0.3 \quad 0.4 \quad 0.3]'. \tag{25}$$

If the beam is directed toward target p at time k, the observation $y_k^{(p)} \in \{d_1, d_2, d_3\}$ is the measured distance of target p from the base station. The observation probability matrices chosen are

$$B^{(1)} = \begin{bmatrix} 0.95 & 0.05 & 0 \\ 0.025 & 0.95 & 0.025 \\ 0 & 0.05 & 0.95 \end{bmatrix}, \quad B^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$B^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1-\alpha & \alpha \end{bmatrix} \tag{26}$$

where $\alpha \in [0, 1]$ is a parameter we will vary to illustrate the performance of the various algorithms.

The costs assigned to tracking each target are proportional to the distance d_i of the target from the base station. This reflects the fact that when a target is close to the base station, the threat is greater, and priority must be given to tracking this target. For each target $p = 1, 2, 3$, the costs assigned were

$$R\left(s_k^{(p)} = d_i, p\right) = \rho^{(p)} d_i + r^{(p)}, \quad (d_1, d_2, d_3) = (0.1, 0.2, 1)$$
$$\rho^{(1)} = 2, \quad r^{(1)} = 5; \quad \rho^{(2)} = 5$$
$$r^{(2)} = 4; \quad \rho^{(3)} = 2, \quad r^{(3)} = 5.5. \tag{27}$$

The weights $\rho^{(p)}$ and $r^{(p)}$ reflect the relative importance of the distance and target type in the cost function. The discount factor was chosen as $\beta = 0.6$.

Results: Target 2 is fully observed 3 state Markov decision process, and its Gittins index is straightforwardly computed. Similar to Section IV-A, for targets 1 and 3, the POMDP program was run to compute $\Lambda_N^{(1)}$ and $\Lambda_N^{(3)}$ with a resolution accuracy $\epsilon = 10^{-5}$ and horizon length $N = 100$. $\Lambda_N^{(1)}$ was found to comprise 1220 vectors. The number of vectors in $\Lambda_N^{(3)}$ versus probability of detection α for various resolution accuracies ϵ is given in Table II.

As in Section IV-A, the performance of the optimal and suboptimal scheduling algorithms were evaluated in terms of the averaged cost incurred in tracking the three targets over the

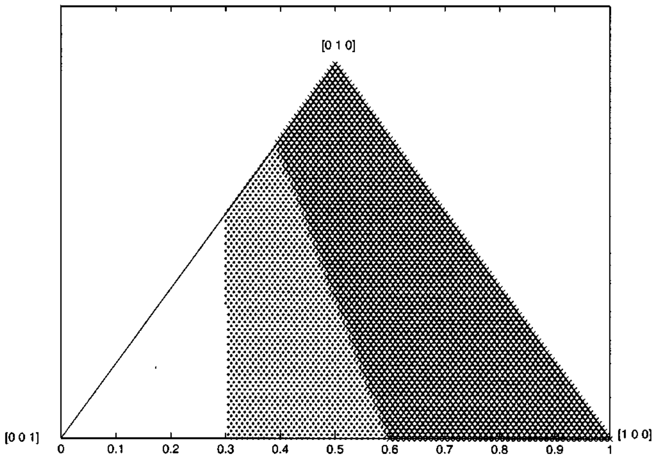


Fig. 6. This shows which target $p^* = \max_{p\in\{1,2,3\}} \gamma^{(p)}(x_k)$ to track at time k if all targets have the same information state x_k. $p^* = 1$ (light), 2 (dark), and 3 (white). Probability of detection $\alpha = 0.7$.

tracking horizon $T = 50$. For each value of α, the average cost was computed by carrying out 1000 independent simulations of the HMM beam scheduling algorithm.

Table II compares the performance of the optimal beam scheduling algorithm with the suboptimal algorithms of Section II-F. The coarse resolution algorithm uses $\epsilon = 10^{-1}$.

It can be seen from Table II that the coarse resolution algorithm with $\epsilon = 10^{-1}$ performs quite similarly to the optimal algorithm. The CM algorithm performs better than the MAP algorithm. Finally, we note that for $\alpha = 0$, the optimal solution is finitely transient with Λ, which is comprised of 53 vectors.

Discussion: To show the Gittins index of all three targets would require three intersecting 3-D plots—and is difficult to visualize. Instead, to get a feel for the performance of the optimal beam scheduling algorithm, we will graphically illustrate the Gittins index for the three targets as follows: Recall that the information state x lives in the space $\mathcal{X}$ defined in (7). For the three-state case, $\mathcal{X}$ is merely an equilateral triangle depicted in Fig. 6, where the three corners of the triangle denote the three Markov chain states. In Fig. 6, we plot $p^*(x) \overset{\triangle}{=} \min_{p\in\{1,2,3\}} \gamma^{(p)}(x)$ for $x \in \mathcal{X}$. Thus, p^* tells us which target to track if the three targets have the same information state x at a given time instant. The lightly shaded region denotes the area where $p^* = 1$, the dark region denotes $p^* = 2$, and the white region inside the triangle denotes the region $p^* = 3$.

It is seen from Fig. 6 that when the three targets are close to the base station [i.e., $x_k^{(p)}$ is close to $[0\ 0\ 1]'$ or equivalently $s_k^{(p)} = d_3$], it is optimal to track target 3 (white region). On the other hand, when there is uncertainty in the state estimates of all three targets (i.e., $x_k^{(p)}$ is in the center of the triangle), it is optimal to track target 1 (light region). This is not surprising since target 1 has the most degree of uncertainty associated with it because it moves the fastest [see $A^{(1)}$ in (25)], and all its states are measured in noise [see $B^{(1)}$ in (26)]. Finally, when the targets are close to state $s_k^{(p)} = d_2$ or $s_k^{(p)} = d_1$ [i.e., $x_k^{(p)}$ is close to $[0\ 1\ 0]'$ or $[1\ 0\ 0]'$], it is optimal to track target 2 (dark region). This is not surprising since target 2 is fully observed.

V. Conclusion

We have presented optimal and suboptimal algorithms for computing Gittins indices of a HMM multiarm bandit problem. These algorithms were then applied to the problem of beam scheduling for multitarget tracking. Numerical examples were presented to illustrate the performance of the algorithms.

The multiarm bandit problem formulation relies on correct specification of the parameters of the different targets. In current work, we are examining the use of simulation based neuro-dynamic programming methods [3] that do not require the parameters to be known. Finally, to fit the multitarget tracking problem within the multiarm bandit framework, we required that the $P-1$ HMM filters that do not receive observations at any given time instant do not update their state estimates $x_k^{(p)}$. This assumption is violated if, for example, an HMM state predictor (which predicts the state of the Markov chain without requiring any observations) is used when no observation are available. Unfortunately, if this assumption is violated, then the problem is no longer a multiarm bandit problem and does not have an indexable (de-coupled) solution.

Another worthwhile extension is the multiarm bandit problem for jump Markov linear systems. Such systems are widely used to model maneuvering targets. The multiarm bandit problem for such systems can be interpreted as computing optimal maneuvers for the targets to avoid being detected.

Acknowledgment

The authors acknowledge Dr. A. Cassandra of Microelectronics and Computer Technology Corporation for allowing us to use his software on Partially Observed Markov Decision Processes for our simulations. This software is freely available from the website [9].

References

[1] Y. Bar-Shalom, Ed., *Multitarget Multisensor Tracking: Applications and Adcances*. Norwell, MA: Artech House, 1993, vol. 2, ch. 8. Multitarget Tracking with an Agile Beam Radar.

[2] D. Berstimas and J. Nino-Mora, “Restless bandits, linear programming relaxations, and a primal dual index heuristic,” *Oper. Res.*, vol. 48, no. 1, pp. 80–90, 2000.

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

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995, vol. 1 and 2.

[5] D. Bertsimas and J. Nino-Mora, “Conservation laws, extended polymatroids and multiarmed bandit problems; A polyhedral approach to indexable systems,” *Math. Oper. Res.*, vol. 21, no. 2, pp. 257–305, May 1996.

[6] D. R. Billetter, *Multifunction Array Radar*. Norwell, MA: Artech House, 1989.

[7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwell, MA: Artech House, 1999.

[8] A. R. Cassandra, “Exact and approximate algorithms for partially observed Markov decision process,” Ph.D. dissertation, Brown Univ., Providence, RI, 1998.

[9] ——, [Online]. Available: http://www.cs.brown.edu/research/ai/pomdp/index.html.

[10] A. R. Cassandra, M. L. Littman, and N. L. Zhang, “Incremental pruning: A simple fast exact method for partially observed Markov decision processes,” in *Proc. 13th Annu. Conf. Uncertainty Artif. Intell.*, Providence, RI, 1997.

[11] D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research*. New York: McGraw-Hill, 1984, vol. 2.

[12] F. Dufour and P. Bertrand, “An image based filter for discrete-time Markovian jump linear systems,” *Automatica*, vol. 32, pp. 241–247, 1996.

[13] Y. Faihe and J. Muller, “Behaviors coordination using restless bandits allocation indexes,” in *Proc. Fifth Int. Conf. Simulation Adapt. Beh.*, Zurich, Switzerland, Aug. 1998.

[14] J. Hardwick and Q. F. Stout, “Flexible algorithms for creating and analysis adaptive designs,” *New Develop. Applicat. Experimental Des.*, vol. 34, pp. 91–105, 1998.

[15] J. C. Gittins, *Multi—Armed Bandit Allocation Indices*. New York: Wiley, 1989.

[16] M. N. Katehakis and A. F. Veinott Jr., “The multiarmed bandit problem: Decomposition and computation.,” *Math. Oper. Res.*, vol. 12, no. 2, pp. 262–268, 1987.

[17] V. Krishnamurthy and R. J. Elliott, “Filters for estimating Markov modulated poisson processes and image based tracking,” *Automatica*, vol. 33, no. 5, pp. 821–833, May 1997.

[18] P. R. Kumar and P. Varaiya, *Stochastic Systems—Estimation, Identification and Adaptive Control*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[19] W. S. Lovejoy, “Computationally feasible bounds for partially observed Markov decision processes,” *Oper. Res.*, vol. 39, no. 1, pp. 162–175, Jan.–Feb. 1991.

[20] ——, “A survey of algorithmic methods for partially observed Markov decision processes,” *Ann. Oper. Res.*, vol. 28, pp. 47–66, 1991.

[21] G. E. Monahan, “A survey of partially observable Markov decision processes: Theory, models and algorithms,” *Manage. Sci.*, vol. 28, no. 1, Jan. 1982.

[22] C. H. Papadimitrou and J. N. Tsitsiklis, “The compexity of Markov decision processes,” *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, 1987.

[23] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, pp. 257–285, 1989.

[24] K. H. Schlag, “A bounded rational approach to the multiarmed bandits,” Univ. Bonn, Dept. Econ. III, Bonn, Germany, Tech. Rep. Discussion Paper no. B-361, Feb. 1996.

[25] R. D. Smallwood and E. J. Sondik, “Optimal control of partially observable Markov processes over a finite horizon,” *Oper. Res.*, vol. 21, pp. 1071–1088, 1973.

[26] E. J. Sondik, “The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs,” *Oper. Res.*, vol. 26, no. 2, pp. 282–304, Mar.–Apr. 1978.

[27] R. L. Streit, *Studies in probabilistic multi-hypothesis tracking and related topics*. Newport, RI: Naval Undersea Warfare Cent. Div., Feb. 1998, vol. SES-98-01.

[28] D. D. Sworder and R. G. Hutchins, “Image-enhanced tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, pp. 701–709, Sept. 1989.

[29] P. P. Varaiya, J. C. Walrand, and C. Buyukkoc, “Extensions of the multiarmed bandit problem: The discounted case,” *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 426–439, May 1985.

[30] P. Whittle, “Multiarmed bandits and the Gittins index,” *J. R. Statist. Soc. B*, vol. 42, no. 2, pp. 143–149, 1980.

Vikram Krishnamurthy (SM’99) was born in 1966. He received the B.E. degree in electrical engineering from the University of Auckland, Auckland, New Zealand, in 1988 and the Ph.D. degree from the Australian National University, Canberra, in 1992.

He is currently a Professor with the Department of Electrical Engineering, University of Melbourne, Parkville, Victoria, Australia, and serves as Deputy Head of department. His research interests span several areas including stochastic scheduling and network optimization, time-series analysis, and statistical signal processing. He is Associate Editor for *Systems and Control Letters*.

Dr. Krishnamurthy is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and has served on the technical program committee of several conferences including the 37th IEEE Conference on Decision and Control, Tampa, FL, in 1998 and the IFAC Symposium on System Identification in 2000, Santa Barbara, CA.

Robin J. Evans was born in Melbourne, Australia, in 1947. He received the B.E. degree in electrical engineering from the University of Melbourne, Parkville, Victoria, Australia, in 1969 and the Ph.D. degree in 1975 from the University of Newcastle, Callaghan, Australia.

He then did postdoctoral studies at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, and the Control and Management Department, Cambridge University, Cambridge, U.K. He spent five years as an engineering officer with the Royal Australian Airforce, working in the area of radar systems. In 1977, he joined the University of Newcastle, where he was Head of the Department of Electrical and Computer Engineering from 1986 to 1991, and Co-Director of an ARC Special Research Centre on Industrial Control Systems between 1988 and 1991. In 1992 he moved to the University of Melbourne, where he was Head of the Department of Electrical and Electronic Engineering until 1996. He is currently Research Leader for the Cooperative Centre for Sensor Signal and Information Processing. His research has ranged across many areas including control theory, radar systems, signal processing, and computer systems.