43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Efficient Methods of Non-myopic Sensor Management for Multitarget Tracking

Chris Kreucher[a,b], Alfred O. Hero III[a], Keith Kastella[b], and Dan Chang[b]

Abstract— **This paper develops two efficient methods of long-term sensor management and investigates the benefit in the setting of multitarget tracking. The underlying tracking methodology is based on recursive estimation of a Joint Multitarget Probability Density (JMPD), implemented via particle filtering methods. The myopic sensor management scheme is based on maximizing the *expected* Rényi Divergence between the JMPD and the JMPD after a new measurement is made. Since a full non-myopic solution is computationally intractable when looking more than a small number of time steps ahead, two approximate strategies are investigated. First, we develop an information-directed search which focusses Monte Carlo evaluations on action sequences that are most informative. Second, we give an approximate method of solving Bellman's equation which replaces the value-to-go with an easily computed function that approximates the long term value of the action. The performance of these methods is compared in terms of tracking performance and computational requirements.**

I. Introduction

The problem of sensor management is to determine the best way to task a sensor or group of sensors when each sensor may have many modes and search patterns. Typically, sensors are used to gain information about the kinematic state (e.g. position and velocity) and identification of a group of targets. Applications of sensor management are often military in nature [11], but also include things such as robot path planning [9]. There are many objectives that the sensor manager may be tuned to meet, e.g. probability of target detection, minimization of track error/covariance, and identification accuracy. Each of these different objectives taken alone may lead to a different allocation strategy [11].

Sensor management schemes may be myopic (short term) or non-myopic (long-term). Long-term sensor scheduling will out perform short-term methods in situations where the dynamics of the scenario are predictably changing and where there are large gaps in sensor coverage. For example, when targets and/or sensor platforms are moving the visibility of a target from a sensor changes with time which may make long-term planning advantageous.

In this paper, we detail a multi-target tracking situation in which non-myopic scheduling outperforms myopic scheduling. This scenario involves a moving sensor which, due to terrain elevation, results in part of the surveillance region being obscured at each time step. We contrast the sensor scheduling decisions made by a myopic scheduler with that of a non-myopic scheduler in terms of resulting track error. As the full non-myopic solution requires computational time exponential in the number of time steps forward that the algorithm plans, we present two alternative schemes. First, we give an information-directed path searching scheme which reduces the complexity of the full Monte Carlo (MC) search and yields similar results. Second, we present an approximate method which replaces the value-to-go by a function which captures the long-term benefit of an action in terms of an "opportunity cost" or "regret".

The paper proceeds as follows. First, Section II is an overview of Bayesian multiple target tracking and our particle filter implementation. Second Section III gives details of our information-based method of myopic sensor management. Third, Section IV provides a motivating example of a scenario in which non-myopic sensor management provides benefit. Fourth, in Section V, we detail the full MC approach to non-myopic sensor management, and note the intractability for long time-scale problems. We develop therein two approximate methods of long term scheduling, including a technique which replaces the value-to-go with a function that approximates the long-term value of an action. Finally, in Section VI we provide simulation results comparing the myopic, non-myopic, and approximate techniques in terms of track error and computational burden.

This work was supported by the USAF Contract No. F33615-02-C-1199, AFRL contract SPO900-96-D-0080, and ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

[a]The University of Michigan Department of EECS, Ann Arbor, MI ckreuche@umich.edu, hero@eecs.umich.edu

[b]General Dynamics Advanced Information Systems, Ann Arbor, MI Keith.Kastella@gd-ais.com

II. Bayesian Multi-target Tracking

Estimating the joint multitarget probability density (JMPD) [8] provides a means for tracking an unknown number of targets in a Bayesian setting. The statistical model uses the joint multitarget conditional probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ as the probability density for exactly T targets with state vectors $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{T-1}, \mathbf{x}_T$ at time k based on a set of observations $\mathbf{Z}^k$. The number of targets T is a variable to be estimated simultaneously with the states. The observation set $\mathbf{Z}^k$ refers to the collection of measurements up to and including time k, i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, where each of the $\mathbf{z}^i$ may be a single measurement or a vector of measurements made at time i. We denote the multitarget state vector by $\mathbf{X}$, i.e. $\mathbf{X} = [\mathbf{x}_1,\ \mathbf{x}_2,\ ...,\ \mathbf{x}_T]$, where $\mathbf{X}$ is defined for $T = 1...\infty$.

Simulations in this paper treat the case where the number of targets is known and fixed, and target states are one-dimensional. More general situations are considered in [8].

0-7803-8682-5/04/$20.00 ©2004 IEEE

The temporal update proceeds according to the usual rules of Bayesian filtering. Given a kinematic model $p(\mathbf{X}^{k+1}|\mathbf{X}^k)$, the prediction density is

$$p(\mathbf{X}^{k+1}|\mathbf{Z}^k) = \int d\mathbf{X}^k p(\mathbf{X}^{k+1}|\mathbf{X}^k)p(\mathbf{X}^k|\mathbf{Z}^k) \ . \quad (1)$$

Given a sensor model, $p(\mathbf{z}^k|\mathbf{X}^k)$, Bayes' rule is used as new measurements $\mathbf{z}^{k+1}$ arrive yielding

$$p(\mathbf{X}^{k+1}|\mathbf{Z}^{k+1}) = \frac{p(\mathbf{z}^{k+1}|\mathbf{X}^{k+1})p(\mathbf{X}^{k+1}|\mathbf{Z}^k)}{p(\mathbf{z}|\mathbf{Z}^k)} \quad (2)$$

$\mathbf{X}$ has a very large sample space. It contains all configurations of state vectors for all values of T. Discretization on a grid has computational burden exponential in the number of targets and grid cells allotted to each state. A particle filter (PF) implementation allows for computational tractability [8]. To implement JMPD via a PF, we approximate the joint multitarget probability density by a set of N_p weighted samples (particles), i.e. $p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_p} w_p \delta(\mathbf{X} - \mathbf{X}_p)$.

III. Information Based Myopic Sensor Management

In this section, we detail our information-based myopic sensor management algorithm. This lays the foundation for extensions for non-myopic sensor management techniques discussed later. At each instance when a sensor is available, we use an information-based method to compute the best sensing action to take. This is done by first enumerating all possible sensing actions. A sensing action may consist of choosing a particular mode (e.g. SAR mode or GMTI mode), a particular dwell point/pointing angle, or a combination of the two. Next, the *expected* information gain is calculated for each possible action, and the action that yields the maximum expected information gain is taken.

Calculation of information gain between two densities f_1 and f_0 is via the Rényi information divergence [12][5]:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha - 1} ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \quad (3)$$

We compute the divergence between the predicted and the updated density after a measurement is made. Our PF approximation of the density simplifies eq. (3) to

$$D_\alpha \left(p(\cdot|\mathbf{Z}^{k+1})||p(\cdot|\mathbf{Z}^k)\right) \propto ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_p} w_p p(\mathbf{z}|\mathbf{X_p})^\alpha \quad (4)$$

The sensor model $p(\mathbf{z}|\mathbf{X}_p)$ incorporates everything about the sensor, including signal to noise ratio, detection probabilities, and whether the locations are visible.

We wish to perform the measurement that makes the divergence between the current density and the density after a new measurement largest. This indicates the action has maximally increased information content of the measurement updated density with respect to the density before a measurement was made. To this end, we calculate the expected value of eq. (4) for each of the N possible sensing actions and choose the action that maximizes the expectation. Let a_i, $i = 1...N$ to refer to the possible sensing actions under consideration, including but not limited to sensor mode selection and sensor beam positioning.

The expected value of eq. (4) is an integral over all possible outcomes z_{a_i} when performing action a_i:

$$||D_\alpha||_{a_i} = \int d\mathbf{z}_{a_i} p(\mathbf{z}_{a_i}|\mathbf{Z}^k) D_\alpha \left(p(\cdot|\mathbf{Z}^k, z_{a_i})||p(\cdot|\mathbf{Z}^k)\right) \quad (5)$$

In the special case of thresholded measurements, we have

$$||D_\alpha||_{a_i} \propto \sum_{z_{a_i}=0}^{1} p(z_{a_i}) ln \frac{1}{p(z_{a_i})^\alpha} \sum_{p=1}^{N_p} w_p p(z_{a_i}|\mathbf{X}_p)^\alpha \quad (6)$$

IV. Non-Myopic Sensor Management : Motivating Example

In many situations, a non-myopic management strategy will provide better decisions than the myopic strategy. In this section, we consider the problem where at each time step an airborne sensor is able to image a portion of a ground surveillance area to determine the location of a set of moving ground targets.

At each time step, the sensor position causes portions of the ground to be unobservable due to terrain elevation between the sensor and the ground. Given the sensor position and the terrain elevation, we can compute a visibility mask which determines how well a particular spot on the ground can be seen by the sensor. As an example, in Fig. IV, we give the visibility masks that are computed from a sensor positioned below and to the left of the surveillance area.

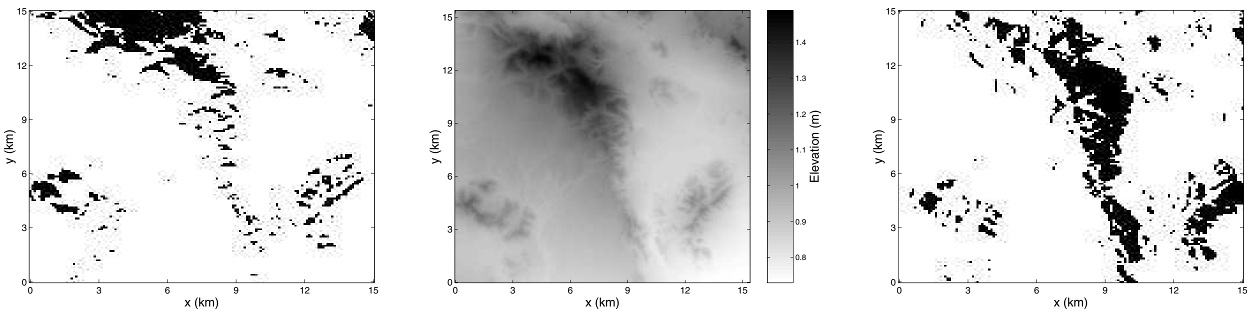
Visibility constraints enter into the sensor management formulation through $p(\mathbf{z})$. The (myopic) sensor manager will calculate small expected gain for a cell that is obscured (if completely obscured, 0 expected gain). Hence, the sensor will not be used to interrogate obscured areas.

A situation where non-myopic management aids tracking is when a target becomes invisible to the sensor for a brief amount of time and then reemerges. Extra sensor dwells immediately before obscuration (at the expense of not interrogating other targets) will sharpen the estimate of target location. This sharpened estimate will allow prediction of where the target will emerge. We illustrate this graphically with a six time-step vignette in Fig. 2.

V. Non-Myopic Sensor Management : Computational Methods

In this section, we present three information-based methods of non-myopic sensor management.

The first method is a Monte Carlo (MC) technique that considers all two step sequences (a_i^k, a_j^{k+1}) and computes the expected information gain by repeatedly simulating its application and computing the average gain. While simple



(a) Visibility mask, sensor below region (b) Elevation map of surveillance region (c) Visibility mask, sensor left of the region

Fig. 1. Visibility masks for a sensor positioned below and left of the surveillance region. We show binary visibility masks (non-visible areas are black and visible areas are white). In general, visibility may be between 0 and 1 indicating areas of reduced visibility, e.g. partially obscured by foliage.

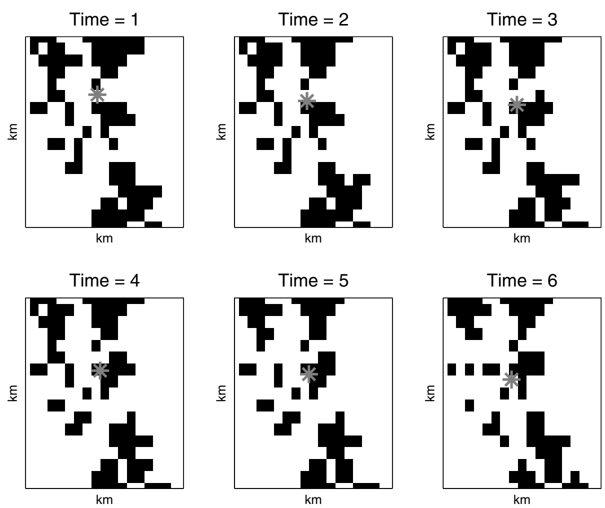


Fig. 2. A six time step vignette where the target moves through an obscured area. The target is depicted by an asterisk. Obscured areas are black and visible areas are white. Extra dwells just before becoming obscured (time = 1) aid in localization after the target emerges (time = 6).

to describe, this method has computational burden $O(N_p * N^H)$, where N is the number of actions at each time and H is the number of time steps the algorithm looks ahead.

The second method is a MC technique that adaptively decides which paths through action space to investigate. Given a compute budget, we use an information-directed algorithm to decide which paths deserve attention. This uses available computations in the best way minimizing the effect of the computational budget on solution quality.

The third method is a technique that approximates the long-term value of an action by the change in information gaining ability over time. This function makes actions that are rewarding due to future considerations more desirable to choose at the current time step, thus approximating the non-myopic decision. The algorithm is $O(N_p * N * H)$.

A. Notation and Preliminaries

Denote the value of state s at time k by $V_k(s)$. We will use $c(s,a)$ as shorthand for the myopic expected gain associated with an action a in state s, that is

$$c(s,a) \doteq ||D_\alpha \left(p(\cdot|\mathbf{Z}^{k+1})||p(\cdot|\mathbf{Z}^k)\right)||_a \tag{7}$$

where s is used as a surrogate for $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$.

Bellman's equation in the discounted reward scenario is

$$V_k(s) = \max_a\{c(s,a) + \gamma E_{s'}[V_{k+1}(s')]\} \tag{8}$$

where $E_{s'}[V_{k+1}(s')] = \sum_{j \in S} p(j|s,a)V_{k+1}(j)$.

The optimal non-myopic action $\hat{a}$ is then given by

$$\hat{a} = arg\max_a\{c(s,a) + \gamma E_{s'}[V_{k+1}(s')]\} \tag{9}$$

B. Monte Carlo Trials for Non-myopic Sensor Management

A straightforward but intractable way of solving eq. (8) is via Monte Carlo (MC) rollout techniques, i.e. repeatedly playing out a given position in order to calculate the expected reward from that position. The two-step rollout procedure is shown in Fig. 3. We first predict the target density at time $(k+1)$ by performing model update. The prediction density, $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$ is used to determine all possible actions at time $k+1$, $a_1^{k+1} \cdots a_N^{k+1}$.

For each action at time $k+1$, we perform the following two steps repeatedly to generate a MC average of the information gain. First, the action is simulated resulting in a measurement $\hat{z}^{k+1}$. The density of $\hat{z}^{k+1}$ is formed from $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$. The simulated measurement is used to update the density forming $p(\mathbf{X}^{k+1}|\mathbf{Z}^k, \hat{z}^{k+1})$. The realized gain in information from this measurement is calculated between $p(\mathbf{X}^{k+1}|\mathbf{Z}^k)$ and $p(\mathbf{X}^{k+1}|\mathbf{Z}^k, \hat{z}^{k+1})$ using eq. (3).

This predicted posterior is then model updated to form the prediction density at time $k+2$, $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1})$. At this point, the expected one-step (myopic) gains for each possible action at time $k+2$ is generated using eq. (6). The value of action a_i^{k+1} is then the actual realized gain from time step $k+1$ to time step $k+2$ plus the mean of the expected gain at time $k+2$. We call this 2-step procedure searching the path associated with the action a_i^{k+1}.

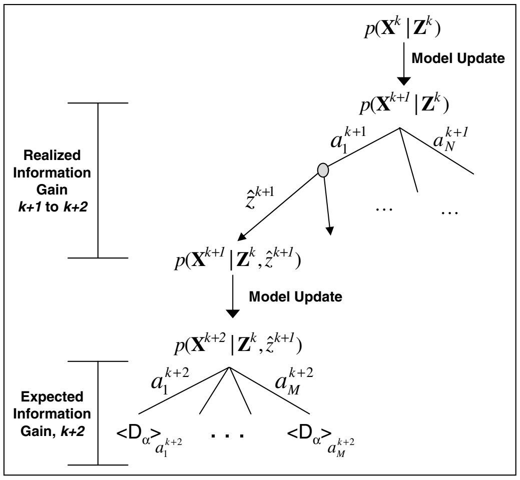


Fig. 3. The two-step non-myopic algorithm is rolled out for all possible actions at time $k+1$. The value of an action is the realized gain from the action plus the expected gain at the next step. This procedure is run many times to generate a MC average of the two-step gain for each action.

The extension to looking more than two time steps into the future is straightforward but computationally prohibitive – the algorithm is $O(N^H)$. For example, a three-step rollout would perform an additional simulation step using $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1})$ to simulate a measurement $\hat{z}^{k+2}$ at time $k+2$. This would generate a predicted posterior at time $k+2$, $p(\mathbf{X}^{k+2}|\mathbf{Z}^k, \hat{z}^{k+1}, \hat{z}^{k+2})$. A model update would form $p(\mathbf{X}^{k+3}|\mathbf{Z}^k, \hat{z}^{k+1}, \hat{z}^{k+2})$, and the expected myopic gain at time $k+3$ would be calculated. This procedure would be repeated for each action at time $k+1$ many times to generate a MC average of the expected gain.

C. Adaptive Trajectory Selection for Improved MC Rollout

In this section, we describe a method of performing MC rollout where we restrict ourselves to searching the tree a small number of times. Given this computational budget, we wish to determine the best trajectories to investigate.

At time $k+1$, there are N possible actions. Each action corresponds to the first step in a trajectory down the tree. Associated with each action is an expected (long-term) gain in information for executing that action, and we wish to determine this as precisely as possible. In section V-B we determined this gain by simply searching down each path many times and using the empirical average of information gain as a surrogate for the expected information gain. Here we wish to select the paths to search to best estimate the expected information gain with a fixed number of samples. We propose to select the best trajectory by computing the gain in information that making an additional simulation will garner. This will provide an automatic method to prune trajectories – i.e. decide which paths are not worth further investigation and which paths deserve greater attention.

We define $p_{a_i}(g|G_{a_i})$ to be a density on the expected long-term gain in information g if we were to actually take action a_i, conditioned on the long-term information gains simulated so far from searching down trajectories starting with action a_i, G_{a_i}. At beginning of each decision epoch, we will have not searched any trajectories yet and so $G_{a_i} = \emptyset$. Our goal is to determine $p_{a_i}(g|G_{a_i})$ for all actions a_i as accurately as possible using a fixed budget, so that when we actually task the sensor we are tasking it to make the action that maximizes the expected long-term gain in information.

At the onset, we have N actions and no idea which action is best. We propose to construct the initial density on the expected long-term information gain for actually taking action a_i by looking down the trajectory associated with action a_i a small number of times (M) to generate samples from the density $p_{a_i}(g|G_{a_i})$. These samples from $p_{a_i}(g|G_{a_i})$ will be used to approximate $p_{a_i}(g|G_{a_i})$ in a PF like manner, e.g. $p_{a_i}(g|G_{a_i}) = \frac{1}{M}\sum_{p=1}^{M}\delta(g-g_p)$.

We then simulate an additional K trajectories to improve our estimate of the expected long term information gain when taking action a_i, $p_{a_i}(g|G_{a_i})$. We use an information directed method to select the trajectory to investigate for each investigation. The method proceeds as follows. For each action a_i, we compute the expected gain in information with respect to $p_{a_i}(g|G_{a_i})$ that making one additional simulation of that action will garner. Then we investigate that path that generates the largest expected gain in information. We repeat this procedure for all K investigations.

Formally, we compute the expected gain in information for investigating action a_i as follows. Before investigating a new path, we have a density $p_{a_i}(g|G_{a_i})$. Assume that we have decided to investigate a particular action and this investigation has generated a new realization of the expected long-term gain $\hat{g}$. The updated density becomes

$$p_{a_i}(g|G_{a_i}, \hat{g}) = \frac{p_{a_i}(\hat{g}|g)p_{a_i}(g|G_{a_i})}{p_{a_i}(\hat{g}|G_{a_i})} \tag{10}$$

Using the Alpha-Divergence metric (eq. 3), and a method identical to that of section III, we can determine that the expected gain in information between $p_{a_i}(g|G_{a_i})$ and $p_{a_i}(g|G_{a_i}, \hat{g})$ for searching the trajectory starting with action a_i is proportional to the entropy of the distribution associated with that action, $\int_g p_{a_i}(g|G_{a_i}) ln\left(p_{a_i}(g|G_{a_i})\right) dg$, which is the intuitive result that the best trajectory to search is the trajectory associated with the highest uncertainty.

D. Approximating the Value-to-go

In this section, we detail another approximate method of determining the long-term value associated with an action. This method directly replaces the second term on the right hand side of eq. (8) and (9), the long term value factor. The strategy is predicated on the following observations. First, if by waiting to perform an action until a later time step, the ability to gain myopic information via an action decreases, the action should have high priority to perform now. Conversely, if the ability to gain myopic information is greater in the future, the action should be delayed.

The approximation we advocate is an information based method for computing the difference between the expected myopic information gaining capability at the current time with the expected myopic information gaining capability at a future time. Intuitively, this captures the "opportunity cost" or "regret" for not taking an action at the current time.

For a concrete example, consider the case of time varying visibility. If an area is predicted to be less visible in the future, the desire to interrogate now should be enhanced. Conversely, if an area is predicted to be more visible in the future, the desire to interrogate it now should be depressed. Of course, more than just visibility must be accounted for. The expected future occupancy and expected future uncertainty is relevant as well. The proposed method accommodates all of these factors simultaneously.

The optimal method for choosing the action to make at the current time, $\hat{a}$ is by evaluating eq. (8). We approximate the value-to-go, $E_{s'}[V_{k+1}(s')]$, by a function $N(s,a)$ which captures the long term reward of action a in state s and is easily computable. Specifically, we approximate eq. (8) by

$$\hat{a} = arg\max_{a}\{c(s,a) + \gamma N(s,a)\} \tag{11}$$

We use as $N(s,a)$ the "gain in information for waiting". Specifically, let $\bar{g}_a^k$ denote the expected myopic gain when taking action a at time k. Furthermore, let $p_a^k(\cdot)$ denote the distribution of myopic gains when taking action a at time k. Then we approximate the long-term value of taking action a by the gain (loss) in information received by waiting until a future time step to take the action,

$$N(s,a) \approx \sum_{m=1}^{M} \gamma^m sgn\big(\bar{g}_a^k - \bar{g}_a^{k+m}\big) D_\alpha\big(p_a^k(\cdot)||p_a^{k+m}(\cdot)\big) \tag{12}$$

where M is the number of time steps in the future that are considered. Each term in the summand has two components. First, $sgn\big(\bar{g}_a^k - \bar{g}_a^{k+m}\big)$ signifies if the expected information gain in the future is more or less than the present. A negative value implies that the future is better and that the action ought to be discouraged at present. A positive value implies that the future is worse and that the action ought to be encouraged at present. The second term, $D_\alpha\big(p_a^k(\cdot)||p_a^{k+m}(\cdot)\big)$ measures the Renyi divergence between the density on myopic gains at the current time step and at a future time step. What results is a magnitude of the difference between the two densities. A small number implies the two are very similar and therefore the non-myopic term will have little impact on the decision making.

To completely specify the technique advocated here, we introduce a weighting w which gives relative precedence to the non-myopic and myopic terms in the approximation to Bellman's equation, i.e. we approximate eq. (8) with

$$\hat{a} = arg\max_a\{c(s,a) + w\sum_{m=1}^{M} \gamma^m sgn\big(\bar{g}_a^k - \bar{g}_a^{k+m}\big) D_\alpha\big(p_a^k(\cdot)||p_a^{k+m}(\cdot)\big)\} \tag{13}$$

As $w \rightarrow 0$ the technique schedules myopically, and as $w \rightarrow \infty$ the technique considers only the future. An appropriate choice for w balances the present and the future.

This technique applies to a variety of other scenarios. For example, consider the convoy-movement scenario. By using kinematic prediction, one may be able to determine that two targets are about to come close together (e.g. enter the same sensor detection cell). This signals reduced ability to gain information about those targets in the future and therefore the targets should be interrogated at the current time step.

VI. Simulation Results

We investigate the following model problem, inspired by the scenario of Section IV. There are two targets each described by a one-dimensional position. The sensor may measure any one of 16 cells, each 1 unit wide. Cell locations are fixed and centered at $.5, 1.5, \cdots, 15.5$ units. The sensor makes three (not necessarily distinct) dwells per time step, receiving binary returns independent from dwell to dwell. In occupied cells, a detection is received with probability P_d (set here at 0.9). In cells that are unoccupied a detection is received with probability P_f (set here at .01).

At the onset, positions of the targets are known only probabilistically to the filter. The filter is initialized with the probability of target 1 location uniformly distributed across sensor cells $\{2 \cdots 6\}$ and the probability of target 2 location uniformly distributed across sensor cells $\{11 \cdots 15\}$.

Visibility of the sensor cells is as follows. At time 1, all cells are visible to the sensor. At times $2, 3, 4$, cells $\{11 \cdots 15\}$ are invisible. At time 5 all cells are visible again. This model problem closely emulates the situation where a target is initially visible to the sensor, becomes obscured, and then reemerges from the obscuration.

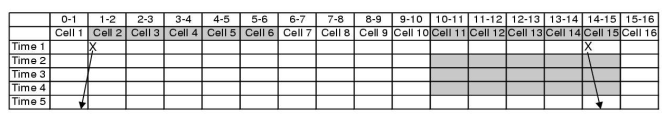


Fig. 4. The model problem. At the onset, the filter has estimates of target 1 and target 2 uniformly distributed across cells $\{2...6\}$ and $\{11...15\}$, respectively. At time 1 all cells are visible. At time 2, 3, and 4 cells $\{11...15\}$ are obscured. This emulates the the situation where one target is initially visible to the sensor, becomes obscured and then reemerges.

At time 1 the myopic strategy, having no information about the future visibility, will choose cells uniformly from the set $\{2...6\} \cup \{11...15\}$. As a result, target 1 and target 2 will on the average be given equal attention. A non-myopic strategy will preferentially choose cells from $\{11...15\}$ as they are to become invisible.

A. Results Using Information Directed Path Interrogation

Fig. 5 presents a comparison between uniform searching (described in Section V-B) with information-directed searching (Section V-C). Performance is compared in terms of median error versus number of paths searched (which measures algorithm complexity). As expected, uniform search requires more path interrogations to yield a desired error since it wastes investigations on paths of little value. The information directed method saves on the order of a factor of $2-4$ in compute time for a given error budget.

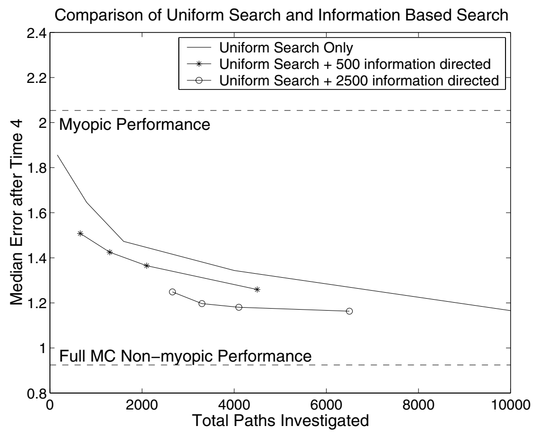


Fig. 5. A comparison between uniform MC and information-directed search. The top curve gives the results of searching each path equally (uniform search). The bottom two curves are each seeded with uniform search and followed by information-directed searches. A comparison is made in terms of the total number of paths searched between the algorithms, which is a measure of algorithm complexity. For a given number of paths searched, information-directed search yields better performance.

B. Results Using Approximation of Value-to-go

We illustrate here the performance of the approximation to Bellman's equation given in Section V-D. Fig. 6 shows the tracking performance in the model problem as a function of the weighting of the value-to-go function, w. As mentioned earlier, at $w = 0$ the algorithm acts myopically so the performance is that of the myopic scheduler of Section III. At large w, the algorithm takes actions based only on long term considerations (i.e. ignores the one-step value of an action). The resulting errors are slightly worse than being myopic. In between, the proper trade between short term and long term considerations is made and the performance nearly reaches that of the exact non-myopic scheduler.

Table I summarizes the performance of the algorithms in terms of compute time and tracking performance.

TABLE I

PERFORMANCE OF THE NON-MYOPIC SCHEDULING ALGORITHMS.

Method	Description (Section)	CPU Time (sec)	Median Error (cells)
Myopic	III	0.189	2.054
Monte Carlo[a]	V-B	10.24	1.473
Monte Carlo	V-B	53.030	0.949
Monte Carlo	V-B	157.32	0.925
Information-Directed	V-C	8.458	1.249
Approximate ($w = 0.05$)	V-D	0.258	0.932

[a]MC Non-myopic shown for 250, 2500, and 5000 searches/path

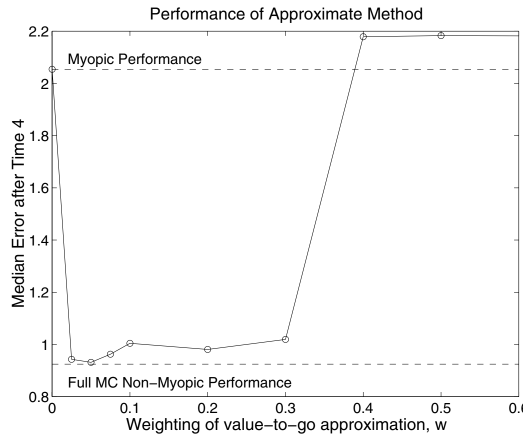


Fig. 6. Performance of the approximate non-myopic scheduler of Section V-D as a function of the weighting of the value-to-go approximation, w. w weights the influence of the one-step value of an action with the long-term value. When chose properly, the two considerations are balanced and the performance equals that of the exact non-myopic scheduler.

VII. Conclusion

This paper has investigated the benefit of long-term sensor scheduling. Since the non-myopic optimization problem is computationally intractable, approximate techniques must be developed. We have detailed two techniques that provide a computational speedup to this optimization problem and demonstrated the performance is equivalent to that of a full non-myopic optimization in a model problem involving time varying visibility due to sensor platform motion.

References

[1] D. P. Bertsekas and D. Castanon, "Rollout Algorithms for Stochastic Scheduling Problems", *Journal of Heuristics*, Vol. 5, pp. 89-108, 1999.

[2] D. Castanon, "Approximate Dynamic Programming for Sensor Management", *Proc. of 1997 Conference on Decision and Control*, 1997.

[3] A. S. Chhetri, D. Morrell, A. Papandreou-Suppappola, "The Use of Particle Filtering with the Unscented Transform to Schedule Sensors Multiple Steps Ahead", International Conference on Acoustics, Speech, and Signal Processing, Montreal, Cananda, May 17-21, 2004.

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

[5] A. O. Hero, B. Ma, O. Michel and J. Gorman, "Applications of entropic spanning graphs," IEEE Signal Processing Magazine (Special Issue on Mathematics in Imaging), Vol 19, No. 5, pp 85-95, Sept. 2002.

[6] K. J. Hintz, "A Measure of the Information Gain Attributable to Cueing", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, pp. 237-244, 1991.

[7] C. Kreucher, K. Kastella, and A. O. Hero III, "Multi-target Sensor Management Using Alpha-Divergence Measures", *Information Processing in Sensor Networks Conference*, April 2003.

[8] C. Kreucher, K. Kastella and A. O. Hero III, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", Proc. SPIE August 2003.

[9] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic Path Planning in Sensor-Based Terrain Acquisition, IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 462-472, August 1990.

[10] R. Mahler, "Global Optimal Sensor Allocation", *Proceedings of the Ninth National Symposium on Sensor Fusion*, vol. I, March 12-14, 1996, Naval Postgraduate School, Monterey CA, pp. 347-366.

[11] S. Musick and R. Malhotra, "Chasing the Elusive Sensor Manager", *Proceedings of NAECON*, Dayton, OH, May 1994, pp. 606-613.

[12] A. Rényi, "On measures of entropy and information", Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.