# Sensor management for multiple target tracking with heterogeneous sensor models

Jason L. Williams,* John W. Fisher III and Alan S. Willsky
Laboratory for Information and Decision Systems and
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, USA

## ABSTRACT

Modern sensors are able to rapidly change mode of operation and steer between physically separated objects. While control of such sensors over a rolling planning horizon can be formulated as a dynamic program, the optimal solution is inevitably intractable. In this paper, we consider the control problem under a restricted family of policies and show that the essential sensor control trade-offs are still captured. The advantage of this approach is that one can obtain the optimal policy within the restricted class in a tractable fashion, in this case by using the auction algorithm. The approach is well-suited for problems in which a single sensor (or group of sensors) is being used to track many targets using a heterogeneous sensor model, *i.e.*, where the quality of observations varies with object state, such as due to obscuration. Our algorithm efficiently weighs the rewards achievable by observing each target at each time to find the best sensor plan within the restricted set. We extend this approach using a roll-out algorithm, to handle additional cases such as when observations take different amounts of time to complete.

## 1. INTRODUCTION

Agile sensors such as phased array radars are able to rapidly share resources between tasks to support a large number of simultaneous estimation problems. The ability to exploit this potential is limited by the computational complexity of the stochastic control algorithms which result from attempting to optimize system performance.

In this paper, we consider a problem in which a single sensor is used to observe multiple processes which are evolving independently. At each time, only one process may be observed, hence the sensor resource manager task is to determine which process to observe at each time. Our method corresponds to the optimal solution of the optimization problem over a constrained set of policies. The structure of the constrained problem is such that the combinatorial optimization can be solved using efficient network flow optimization methods, allowing planning to be conducted over a long planning horizon at low computational cost.

In Section 2, we describe the basic formulation upon which we build our method. The method is described in Section 3, and compared briefly to existing strategies. The results of computational simulations are examined in Section 4.

## 2. BACKGROUND AND FORMULATION

In this section we describe a mathematical formulation for the problem of sensor management for tracking multiple independent objects. The model provides the basis for construction our algorithm, as described in Section 3.

* E-mail: jlwil@mit.edu, Telephone: +1 (617) 253-7220

Signal Processing, Sensor Fusion, and Target Recognition XV, edited by Ivan Kadar,
Proc. of SPIE Vol. 6235, 62350F, (2006) · 0277-786X/06/$15 · doi: 10.1117/12.666142

### 2.1. Estimation problem

We denote by $\boldsymbol{x}_k^i$ the state of process $i \in \{1, \ldots, N\}$ at time $k$. The goal of the system is to estimate each of these states with the best accuracy possible, according to some performance objective to be defined later. The state of each process may be:

1. continuous (*e.g.*, the position and velocity of an object which is being tracked),
2. discrete (*e.g.*, the class of an object which is being identified), or
3. mixed (*e.g.*, the combination of the two).

The state of each process evolves independently of other process according to the dynamics equation:

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{f}(\boldsymbol{x}_k^i, \boldsymbol{w}_k^i) \tag{1}$$

where $\boldsymbol{w}_k^i$ is independent of $\boldsymbol{w}_l^j$ for all $(k,i) \neq (l,j)$, and $\boldsymbol{w}_k^i \sim p_{\boldsymbol{w}_k^i}(\boldsymbol{w}_k^i)$. The process dynamics in Eq. (1) induces a transition Probability Density Function (PDF) $p_{\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i}(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i)$. Accordingly, if the joint PDF of $\{\boldsymbol{x}_k^1, \ldots, \boldsymbol{x}_k^N\}$ is:

$$p_{\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^N}(\boldsymbol{x}_k^1, \ldots, \boldsymbol{x}_k^N) = \prod_{j=1}^{N} p_{\boldsymbol{x}_k^j}(\boldsymbol{x}_k^j) \tag{2}$$

then the joint PDF at the following time will become:

$$p_{\boldsymbol{x}_{k+1}^1,\ldots,\boldsymbol{x}_{k+1}^N}(\boldsymbol{x}_{k+1}^1, \ldots, \boldsymbol{x}_{k+1}^N) = \prod_{j=1}^{N} p_{\boldsymbol{x}_{k+1}^j}(\boldsymbol{x}_{k+1}^j) \tag{3}$$

where

$$p_{\boldsymbol{x}_{k+1}^i}(\boldsymbol{x}_{k+1}^i) = \int p_{\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i}(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i) p_{\boldsymbol{x}_k^i}(\boldsymbol{x}_k^i) \mathrm{d}\boldsymbol{x}_k^i \tag{4}$$

We assume that, if we choose to observe process $i$ at time $k$, the resulting observation $\boldsymbol{z}_k$ will depend only on the state of process $i$:

$$p_{\boldsymbol{z}_k|\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^N}(\boldsymbol{z}_k|\boldsymbol{x}_k^1, \ldots, \boldsymbol{x}_k^N) = p_{\boldsymbol{z}_k|\boldsymbol{x}_k^i}(\boldsymbol{z}_k|\boldsymbol{x}_k^i) \tag{5}$$

This effectively excludes joint observation processes (*e.g.* data association). Under these assumptions, the joint PDF of $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ conditioned on the observation of $\boldsymbol{x}_k^i$, $\boldsymbol{z}_k$, can be expressed as:

$$p_{\boldsymbol{x}_k^1,\ldots,\boldsymbol{x}_k^N|\boldsymbol{z}_k}(\boldsymbol{x}_k^1, \ldots, \boldsymbol{x}_k^N|\boldsymbol{z}_k) = p_{\boldsymbol{x}_k^i|\boldsymbol{z}_k}(\boldsymbol{x}_k^i|\boldsymbol{z}_k) \prod_{\substack{j=1 \\ j\neq i}}^{N} p_{\boldsymbol{x}_k^j}(\boldsymbol{x}_k^j) \tag{6}$$

where $p_{\boldsymbol{x}_k^i|\boldsymbol{z}_k}(\boldsymbol{x}_k^i|\boldsymbol{z}_k)$ is calculated from $p_{\boldsymbol{x}_k^i}(\boldsymbol{x}_k^i)$ and $p_{\boldsymbol{z}_k|\boldsymbol{x}_k^i}(\boldsymbol{z}_k|\boldsymbol{x}_k^i)$ using Bayes' rule.

### 2.2. Stochastic control problem

The sensor resource management problem which we address in this paper is that of selecting at each time $(k)$ which process $u_k \in \{1, \ldots, N\}$ to observe. This is a stochastic control problem,[1] because the value of the measurement resulting from a particular choice of control is non-deterministic. The decision state of the dynamic program is the joint PDF of the process states conditioned on previously received observations; the PDF can be represented as a product of the marginal process state PDFs as discussed in Section 2.1. We define the shorthand $\mathbb{X}_k \triangleq p_{\boldsymbol{x}_k^{1:N}|\boldsymbol{z}_{0:k-1}}(\boldsymbol{x}_k^{1:N}|\boldsymbol{z}_{0:k-1})$ to denote this decision state.

We select as the per-stage reward the mutual information between the state of the process which we choose to observe, $\boldsymbol{x}_k^{u_k}$ and the resulting observation, $\boldsymbol{z}_k$, defined as the expected reduction of entropy in the state of the observed process due to the new observation:[2]

$$\begin{aligned} g(\mathbb{X}_k, u_k) &= I(\boldsymbol{x}_k^{1:N}; \boldsymbol{z}_k | \boldsymbol{z}_{0:k-1}) = I(\boldsymbol{x}_k^{u_k}; \boldsymbol{z}_k | \boldsymbol{z}_{0:k-1}) \\ &\triangleq H(\boldsymbol{x}_k^{u_k} | \boldsymbol{z}_{0:k-1}) - H(\boldsymbol{x}_k^{u_k} | \boldsymbol{z}_{0:k-1}, \boldsymbol{z}_k) \end{aligned} \tag{7}$$

Although not explicit in our notation in Eq. (7), we condition on the *value* of the past measurements, $\boldsymbol{z}_{0:k-1}$ (which have already been realized), and on the *random variable* corresponding to the new measurement $\boldsymbol{z}_k$ (which has not yet been realized). Note that, when conditioning on a random variable, we must take an expectation over the possible values that the measurement may ultimately assume.

We denote by $\mu_k(\cdot)$ a control policy for time $k$, *i.e.*, a mapping from decision state to control value, such that $u_k = \mu_k(\mathbb{X}_k)$ is the control which we would apply at time $k$ if the decision state was $\mathbb{X}_k$. At time $k$, we seek to find the series of control policies $(\mu_k, \ldots, \mu_{k+M-1})$ which will maximize the reward over the next $M$ time steps:

$$(\mu_k^*, \ldots, \mu_{k+M-1}^*) = \underset{\mu_k, \ldots, \mu_{k+M-1}}{\arg\max} \; \mathrm{E}\left[\sum_{l=k}^{k+M-1} g(\mathbb{X}_l, \mu_l(\mathbb{X}_l))\right] \tag{8}$$

Conceptually, the dynamic program in Eq. (8) can be solved using $M$ steps of value iteration.[1] However, this requires us to store and evaluate reward-to-go functions for every value of the decision state $\mathbb{X}_k$, *i.e.*, every possible PDF of joint object state. In this problem, no finite parameterization of the reward-to-go function is known, hence the method cannot be applied. The high dimensionality of the space of decision states also precludes approximate methods involving discretization (as one would need to discretize the space of PDFs of joint object state).

#### 2.2.1. Open Loop Feedback Control

Open Loop Feedback Control (OLFC) is a commonly used suboptimal control method in which the controller designs at time $k$ an open loop plan for the next $M$ steps, $(u_k, \ldots, u_{k+M-1})$, assuming that no further information (*i.e.*, observations) will become available during execution of the plan. After $m < M$ steps of the plan have been executed and observations have become available, they are incorporated and an updated plan is generated. Typically $m = 1$, in which case the plan is regenerated after each new observation is received.

The equation which the OLFC must solve at time $k$ is:

$$(u_k^*, \ldots, u_{k+M-1}^*) = \underset{u_k, \ldots, u_{k+M-1}}{\arg\max} \; \mathrm{E}\left[\sum_{l=k}^{k+M-1} g(\mathbb{X}_l, u_l)\right] \tag{9}$$

While simpler than Eq. (8), this equation is still a hard combinatorial optimization problem because the decision states $\mathbb{X}_l$ at later times depend upon the choice of controls at earlier times (*i.e.*, the decisions between different times are coupled). Since each control choice $u_l \in \{1, \ldots, N\}$ (corresponding to observing one of the $N$ processes), the number of possible combinations which must be considered is $N^M$. Furthermore, for most dynamics and observation models (linear Gaussian models are one of the few exceptions), the computational cost of evaluating the expected value of the reward for a particular choice of $(u_k, \ldots, u_{k+M-1})$ is also exponential in $M$.

## 3. PROPOSED APPROACH

In this paper, we propose solving the OLFC in Eq. (9) within a restricted class of open loop plans, in which each process *can only be observed once* during the $M$-step planning horizon. We expect that this will not be overly restrictive if the planning horizon $M$ is small compared to the number of processes $N$ (*i.e.*, $N \gg M$). For example, if we are using a single sensor to track many objects, and we can only observe one object at any one time, the optimal solution is likely to involve observing each object once, and then waiting for the uncertainty in the estimate of that object to grow again before observing it a second time, rather than continually observing the same object.

Illustration of reward trajectories and resulting assignment

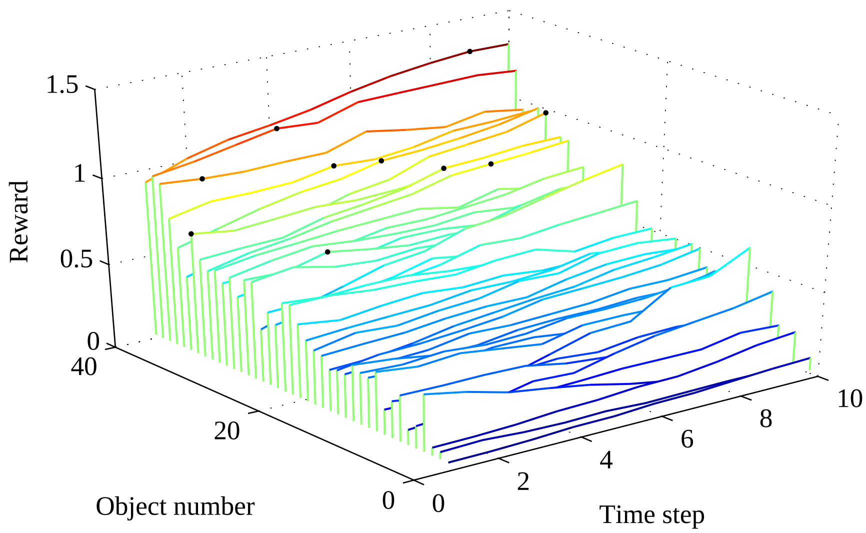


**Figure 1.** Example of operation of auction-based assignment algorithm from simulation in Section 4. Each "strip" in the diagram corresponds to the reward for observing a particular object at different times over the 10-step planning horizon (assuming that it is only observed once within the horizon). The role of the auction algorithm is to pick one unique object to observe at each time in the planning horizon in order to maximize the sum of the rewards gained. The solution is shown as black dots.

This approach averts the complexity of both the combinatorial optimization (which can be solved using the efficient auction algorithm[3]) and the estimation of rewards (since objects are independent and each object is only observed once, the reward of an $M$-measurement sequence can be decomposed into the sum of rewards of $M$ single measurements).

### 3.1. Auction-based formulation

The restricted OLFC we seek to solve is a form for which the combinatorial complexity can be avoided using a well-known efficient solution. One can gain an intuition for the resulting algorithm from the diagram in Fig. 1. Each "strip" in the diagram corresponds to the reward for measuring a particular object at different times over the 10-step planning horizon (assuming that the object is only observed once in the horizon). The role of the auction algorithm is to pick one unique object to observe at each time in the planning horizon in order to maximize the sum of the rewards gained. The solution is shown as black dots. The rewards shown in the diagram correspond to a sample taken from the simulation discussed in Section 4.

We seek to solve the restricted OLFC:

$$(u_k^*, \dots, u_{k+M-1}^*) = \underset{u_k,\dots,u_{k+M-1}}{\arg\max}\ \mathrm{E}\left[\sum_{l=k}^{k+M-1} g(\mathbb{X}_l, u_l)\right] \quad \text{s.t. } u_l \neq u_{l'}\ \forall\, l \neq l' \tag{10}$$

Since the PDF of each process state evolves according to Eq. (4) unless it is observed, the $l$-th term in the sum of Eq. (10) is:

$$g(\mathbb{X}_l, u_l) = I(\boldsymbol{x}_l^{u_l}; \boldsymbol{z}_l | \boldsymbol{z}_{0:l-1}) = I(\boldsymbol{x}_l^{u_l}; \boldsymbol{z}_l | \boldsymbol{z}_{0:k-1}) \tag{11}$$

where the second equality is due to the fact that no information has been received about process $u_l$ since the start of the planning horizon ($k$). Thus we can define $g_l^{u_l} = I(\boldsymbol{x}_l^{u_l}; \boldsymbol{z}_l | \boldsymbol{z}_{0:k-1})$, and Eq. (10) can be written equivalently as:

$$(u_k^*, \ldots, u_{k+M-1}^*) = \underset{u_k, \ldots, u_{k+M-1}}{\arg\max} \left[ \sum_{l=k}^{k+M-1} g_l^{u_l} \right]$$
$$\text{s.t. } u_l \neq u_{l'} \; \forall \; l \neq l' \tag{12}$$

The Linear Programming (LP) relaxation of the optimization problem in Eq. (12) can be written by converting the control variables $u_l$ to flags $u_l^i$ ($u_l^i = 1$ if $u_l = i$ and zero otherwise):

$$\begin{aligned} \max & \sum_{l=1}^{M} \sum_{i=1}^{N} u_l^i g_l^i \\ \text{s.t. } & u_l^i \geq 0 \; \forall \; i, l \\ & \sum_{i=1}^{N} u_l^i \leq 1 \; \forall \; l \\ & \sum_{l=1}^{M} u_l^i \leq 1 \; \forall \; i \end{aligned} \tag{13}$$

Since the constraints in this problem are unimodular, one can always find an optimal solution with integer components, thus one can always find an optimal solution of Eq. (13) corresponding to the optimal solution of Eq. (12). An $\epsilon$-optimal integer solution to the LP in Eq. (13) can be found efficiently using the asymmetric auction algorithm.[3]

#### 3.1.1. Multiple sensors

An extension of this basic problem structure is one involving multiple sensors. In this case, we break each time step into sub-steps, each of which corresponds to a different sensor. The structure of the problem is then identical to that in Eq. (13): each process can be observed a total of once in the planning horizon, and each sensor can observe a single process at each time.

#### 3.1.2. Multiple sensing modes

Another extension of the structure involves a sensor with multiple modes, each of which measures the selected process through a different observation model. In this case, the problem is no longer an assignment problem, although it remains a network flow problem, and hence it remains solvable using efficient methods.

### 3.2. Roll-out

Roll-out methods use a one-step (or longer) lookahead in combination with the reward-to-go corresponding to a heuristic "base" policy. Denoting by $J_l^\pi(\mathbb{X}_l)$ the reward-to-go of the base policy $\pi$ starting from state $\mathbb{X}_l$ at time $l$, the roll-out policy $\mu_k^r$ would select at time $k$ and state $\mathbb{X}_k$ the action

$$\mu_k^r(\mathbb{X}_k) = \underset{u_k}{\arg\max} \left\{ g(\mathbb{X}_k, u_k) + \underset{\mathbb{X}_{k+1}|\mathbb{X}_k, u_k}{\mathrm{E}} J_{k+1}^\pi(\mathbb{X}_{k+1}) \right\} \tag{14}$$

The algorithm described in Section 3.1 can be used as a base policy in a roll-out in order to obtain an improved policy. The structure of the resulting algorithm is:

- For each choice of control at time $k$ ($u_k$):
  - Simulate a number of measurements which could result from applying $u_k$
  - For each measurement value:
    - Calculate the probabilistic state which follows $\mathbb{X}_k$ after incorporating the new measurement

      * Run the auction algorithm commencing from this updated state

  – Evaluate the reward of the action $u_k$ as the average over the rewards resulting from each of the simulated measurement values

- Select the action with the highest reward

An alternative version of algorithm uses the auction method a single time for each choice of control, using reward values conditioned on the control choice (avoiding the process of simulating measurement values). The former corresponds to a closed loop stochastic control roll-out based on the open loop base policy produced by the auction method, while the latter corresponds to an open loop roll-out of the same base policy (resulting in a computational saving).

This algorithm has several advantages over using the auction-based method in Section 3.1 directly. Firstly, the limitation of only allowing each process to be observed once in the planning horizon is mitigated to some extent. The roll-out algorithm considers observing each process at the current time step, and then uses the auction method to construct a plan to use following that observation. In this way, the auction is able to observe, another time, the process being considered for observation in the first lookahead step. Since this procedure is repeated at each time step, the actual control applied by the roll-out method may be to observe the same process sequentially for an arbitrarily long period (in the unlikely case in which this appears to be the best action).

Another advantage of this method is that it could be used to evaluate the benefit of observations that take different durations to complete. For example, one could use a roll-out to compare the benefit of taking observations of different durations at the current time step by evaluating the reward of an $m$-step duration observation, and adding it to the reward obtained by the auction method over the following $(M - m)$ steps (all tasks must have fixed duration in the auction).

### 3.3. Comparison to greedy method

Sensor management with information theoretic criteria is often performed using a greedy heuristic method, in which the action taken at time $k$ is the one which yields the largest instantaneous reward at time $k$. While such a method is able to yield action sequences which observe the same object multiple times in the planning horizon, its selection of actions is ignorant of upcoming observations, hence it cannot capture trade-offs such as the desire to observe an object with a lesser observation at the current time if it will be unobservable in the future.

Interestingly, however, the following analysis shows that the measurement sequence produced by the greedy method is guaranteed to be within a factor of two of the optimal sequence. The analysis is related to recent work by Krause and Guestrin,[4] which deals with the problem of selecting the best $K$-element subset of observations, rather than the problem of concern for this paper, *i.e.*, selecting the best observation at each time, where at each stage we choose from a different set of observations.

To commence, note that, if measurements are conditionally independent conditioned on the state $x$ (which we take as being the joint state of all processes over all time), then the mutual information between a given measurement and the state is reduced when conditioning on additional measurements is introduced; this property is referred to as submodularity. Denoting by $z^\alpha$ and $z^\beta$ the sets of measurements corresponding to action choice sets $\alpha$ and $\beta$, this can be formalized as:

$$\begin{aligned} I(x; z^a | z^\alpha, z^\beta) &= H(z^a | z^\alpha, z^\beta) - H(z^a | x) && (15) \\ &\leq H(z^a | z^\alpha) - H(z^a | x) && (16) \\ &= I(x; z^a | z^\alpha) && (17) \end{aligned}$$

where the inequality in Eq. (16) is the well-known fact that conditioning (on a random variable) reduces entropy.[2] Subsequently, we define the greedy algorithm as choosing at stage $i$ the observation:

$$g_i = \arg\max_g I(x; z_i^g | z_1^{g_1}, \ldots, z_{i-1}^{g_{i-1}}) \tag{18}$$

Consider the optimal observation sequence $\{z_1^{o_1}, \dots, z_M^{o_M}\}$ (or, for that matter, any other observation sequence). Since our reward function is increasing:

$$I(x; z_1^{o_1}, \dots, z_M^{o_M}) \leq I(x; z_1^{o_1}, \dots, z_M^{o_M}, z_1^{g_1}, \dots, z_M^{g_M}) \tag{19}$$

Using the mutual information chain rule:[2]

$$= \sum_{i=1}^{M} \Big[ I(x; z_i^{g_i} | z_1^{g_1}, \dots, z_{i-1}^{g_{i-1}}, z_1^{o_1}, \dots, z_{i-1}^{o_{i-1}}) + I(x; z_i^{o_i} | z_1^{g_1}, \dots, z_{i-1}^{g_i}, z_1^{o_1}, \dots, z_{i-1}^{o_{i-1}}) \Big] \tag{20}$$

By submodularity, we can remove any subset of the conditionings that we desire:

$$\leq \sum_{i=1}^{M} \Big[ I(x; z_i^{g_i} | z_1^{g_1}, \dots, z_{i-1}^{g_{i-1}}) + I(x; z_i^{o_i} | z_1^{g_1}, \dots, z_{i-1}^{g_{i-1}}) \Big] \tag{21}$$

Finally, by definition of $g_i$:

$$\leq 2 \sum_{i=1}^{M} I(x; z_i^{g_i} | z_1^{g_1}, \dots, z_{i-1}^{g_{i-1}}) \tag{22}$$

$$= 2I(x; z_1^{g_1}, \dots, z_M^{g_M}) \tag{23}$$

This result is surprising, especially since it is applicable to time-varying models. To our knowledge, this is the first presentation of this bound.

Thus we compare the performance of our auction approach to the greedy method, bearing in mind that the optimal open loop performance can be no better than twice that of the greedy algorithm.

### 3.4. Comparison to existing methods

Krishnamurthy and Evans[5,6] studied a similar problem to that examined here, finding an optimal solution to the problem of tracking multiple independent objects using a Hidden Markov model. The key assumption which we seek to avoid here is that the information state of unobserved objects does not change between time steps. In our model, the dynamic evolution of unobserved objects can change the observation characteristics dramatically, hence such an assumption is inappropriate. The other advantage of our approach is that it does not rely on conventional POMDP solution methods, for which computational complexity severely limits the number of states in the underlying estimation problem.

Castañon[7] also studied a similar problem, involving classification of a large number of independent objects. This work formulates the resource management problem as a constrained dynamic program, and solves the Lagrangian relaxation optimally. The formulation naturally allows for tasks which require different durations to complete. While our method provides a heuristic method for addressing circumstances in which observations require different durations to complete, the major difference in our work is the ability to address large state spaces (not requiring solution of POMDPs) and time varying observation models.

The problem in question is also similar to that studied by Kreutcher, *et al*,[8] although our method has included a reward-to-go corresponding to the solution of an optimization problem, rather than a heuristic approach.

## 4. SIMULATION RESULTS

The approach described in Section 3 was tested on a tracking scenario in which a single sensor is used to simultaneously track $N$ objects, where $N$ is set to 20 and 40 in different tests. The state object $i$ at time $k$, $\boldsymbol{x}_k^i$, consists of position and velocity in two dimensions. The state evolves according to a linear Gaussian model:

$$\boldsymbol{x}_{k+1}^i = \mathbf{F}\boldsymbol{x}_k^i + \boldsymbol{w}_k^i \tag{24}$$

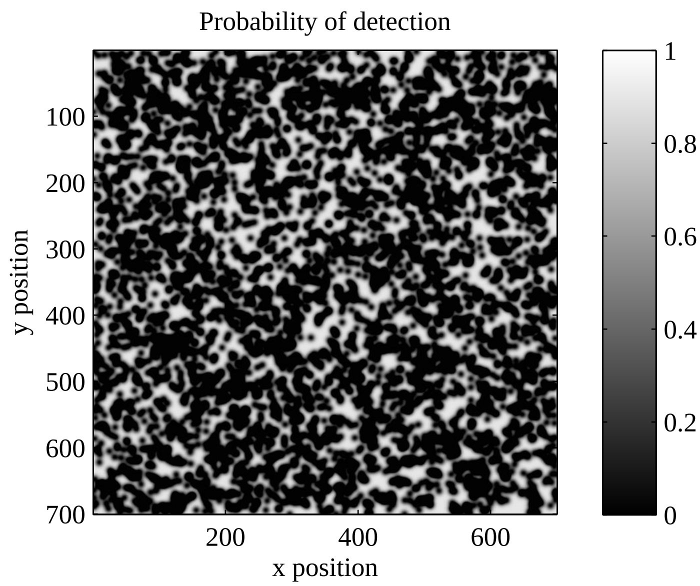


**Figure 2.** Example of randomly generated detection map. The color intensity indicates the probability of detection at each $x$ and $y$ position in the region.

where $\boldsymbol{w}_k^i \sim \mathcal{N}\{\boldsymbol{w}_k^i; \mathbf{0}, \mathbf{Q}\}$ is a white Gaussian noise process. $\mathbf{F}$ and $\mathbf{Q}$ are set as:

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}; \ \mathbf{Q} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \tag{25}$$

The diffusion strength $q$ is set to 0.01. The sensor can be used to observe any one of the $N$ objects in each time step. The measurement obtained from observing object $u_k$ with the sensor consists of a detection flag $d_k \in \{0, 1\}$ and, if $d_k = 1$, a linear Gaussian measurement of the position, $\boldsymbol{z}_k$:

$$\boldsymbol{z}_k = \mathbf{H}\boldsymbol{x}_k^{u_k} + \boldsymbol{v}_k \tag{26}$$

where $\boldsymbol{v}_k \sim \mathcal{N}\{\boldsymbol{v}_k; \mathbf{0}, \mathbf{R}\}$ is a white Gaussian noise process. $\mathbf{H}$ and $\mathbf{R}$ are set as:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \ \mathbf{R} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \tag{27}$$

The probability of detection $P_{d_k|\boldsymbol{x}_k}(1|\boldsymbol{x}_k)$ is a function of object position. The function is randomly generated for each Monte Carlo simulation; an example of the function is illustrated in Fig. 2. The function may be viewed as an obscuration map, *e.g.* due to foliage.

The performance over 200 Monte Carlo runs is illustrated in Fig. 3 for $N = 20$ objects, and in Fig. 4 for $N = 40$ objects. The point with a planning horizon of zero corresponds to a raster, in which objects are observed sequentially. With a planning horizon of one, the auction-based algorithm corresponds to greedy selection. The performance is measured as the average (over the 200 simulations) total change in entropy due to incorporating chosen measurements over all time.

The diagrams demonstrate that, with the right choice of planning horizon, the auction method is able to improve performance over the greedy method. While the improvement is only nominal, this is not unexpected

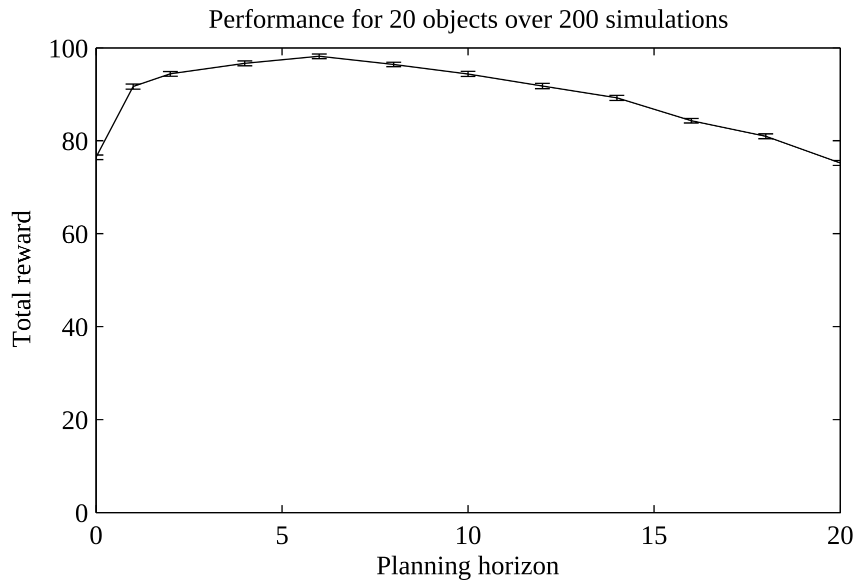


**Figure 3.** Performance tracking $N = 20$ objects. Performance is measured as the average (over the 200 simulations) total change in entropy due to incorporating chosen measurements over all time. The point with a planning horizon of zero corresponds to observing objects sequentially; with a planning horizon of one the auction-based method is equivalent to greedy selection. Error bars indicate 1-$\sigma$ confidence bounds for the estimate of average total reward.

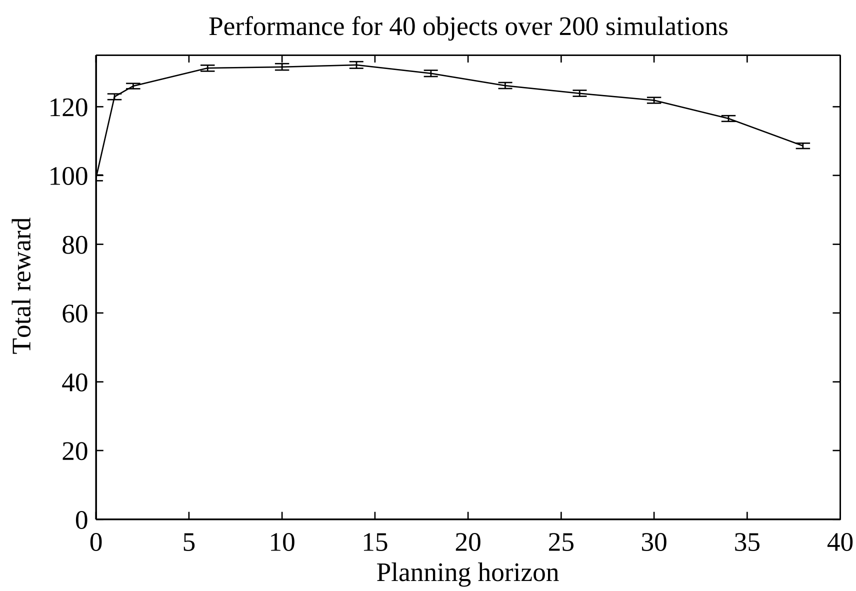


**Figure 4.** Performance tracking $N = 40$ objects.

considering the analysis in Section 3.3, which establishes a bound for the performance of the greedy algorithm versus the optimal method.

The reduction in performance for larger planning horizons is a consequence of the restriction to observe each object at most once in the horizon. If the planning horizon is on the order of the number of objects, we are then, in effect, enforcing that each object *must* be observed once. As illustrated in Fig. 1, in this scenario, there will often be objects receiving low reward values throughout the planning interval, hence by forcing the controller to observe each object, we are forcing it to (at some stage) take observations of little value. Using this insight, we would expect that the optimal choice of planning horizon is related to the number of objects we expect to be clearly observable at some stage in the horizon. Fig. 4 demonstrates that the sensitivity to the planning horizon length is reduced in scenarios involving more objects.

Results utilizing the roll-out method described in Section 3.2 are pending at the time of publication.

## 5. CONCLUSION

The formulation in Section 3 demonstrates that the sensor resource management problem involving many objects can be solved suboptimally using an efficient auction algorithm. The simulation results demonstrate that the resulting strategy can improve performance over the greedy method, which itself possesses an open loop performance guarantee. The method can be extended to accommodate additional problem structure including multiple sensors, multiple sensor modes and sensing actions requiring different numbers of time steps to complete.

While the model is based around assumptions of independent process evolution and independent observations (precluding data association), it may be possible to apply the algorithm to scenarios in which the assumptions are not met; this is a topic of future study.

The greatest limitation of the auction-based heuristic method is the restriction that each process can only be observed once during the planning horizon. We are currently examining extensions which combine the strengths of the greedy selection and the auction method in a modified auction algorithm which is able to select multiple observations of each process in the planning horizon in a greedy manner while still capturing the trade-off between processes that enables the auction-based method to outperform the purely greedy approach.

## ACKNOWLEDGMENT

This work was supported in part by ODDR&E MURI through ARO grant DAAD19-00-0466 and MIT Lincoln Laboratory through ACC PO#3019934.

## REFERENCES

1. D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA, second ed., 2000.
2. T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, NY, 1991.
3. D. Bertsekas, "Auction algorithms for network flow problems: A tutorial introduction," *Computational Optimization and Applications* **1**, pp. 7–66, 1992.
4. A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *UAI 2005*, July 2005.
5. V. Krishnamurthy and R. Evans, "Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking," *Signal Processing, IEEE Transactions on* **49**, pp. 2893–2908, December 2001.
6. V. Krishnamurthy and R. Evans, "Correction to 'Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking'," *Signal Processing, IEEE Transactions on* **51**, pp. 1662–1663, June 2003.
7. D. A. Castañon, "Approximate dynamic programming for sensor management," in *Proc 36th Conference on Decision and Control*, pp. 1202–1207, IEEE, December 1997.
8. C. M. Kreucher, A. O. Hero III, K. Kastella, and D. Chang, "Efficient methods of non-myopic sensor management for multitarget tracking," in *43rd IEEE Conference on Decision and Control*, December 2004.