

A CONSTRAINT GENERATION INTEGER PROGRAMMING APPROACH TO INFORMATION THEORETIC SENSOR RESOURCE MANAGEMENT

Jason L. Williams,^{1,2} John W. Fisher III,² Alan S. Willsky^{1,2}

¹Laboratory for Information and Decision Systems and

²Computer Science and Artificial Intelligence Laboratory

Massachusetts Institute of Technology, Cambridge MA 02139

jlwilliams@alum.mit.edu fisher@csail.mit.edu willsky@mit.edu

ABSTRACT

Many estimation problems involve sensors which can be actively controlled to alter the information received and utilized in the underlying inference task. In this paper, we discuss a novel integer programming method which exploits the submodularity of information theoretic estimation criterion to find an efficient solution to constructing an open loop plan for sensor resource management problems involving many independent objects. The integer programming formulation solves a sequence of simplified problems; the solution of each forms an upper bound to the full complexity problem. The updates performed between iterations may be viewed as steps in a constraint generation process, ensuring that the bound is successively tightened. An auxiliary problem also provides a lower bound to the optimal solution, and a solution attaining that bound, enabling early termination with a guaranteed near-optimal solution. Computational experiments demonstrate the benefit that the algorithm can provide in various planning problems.

Index Terms— Sequential decision procedures, sequential estimation, tracking

1. INTRODUCTION

Active sensing is motivated by modern sensors which can be controlled to observe different aspects of an underlying probabilistic process. For example, if we use cameras to track people in buildings, we can steer the camera to focus or zoom on different people or places; in a sensor network, we can choose to activate and deactivate different nodes and different sensing modalities within a particular node; or in a medical diagnosis problem we can choose which tests to administer to a patient. In each of these cases, our control choices impact the information that we receive in our observation, and thus the performance achieved in the underlying inference task.

A commonly used performance objective in active sensing is mutual information (MI) (*e.g.*, [1]). Denoting the quantity that we aim to infer as X and the observation resulting

from control choice u as z^u , the MI between X and z^u is defined as the expected reduction in the entropy produced by the observation [2], *i.e.*, $I(X; z^u) = H(X) - H(X|z^u) = H(z^u) - H(z^u|X)$.¹ Since $H(X)$ is invariant to the control choice u , choosing u to maximize $I(X; z^u)$ is equivalent to minimizing the uncertainty in X as measured by the conditional entropy $H(X|z^u)$.

In this paper we propose a novel method for addressing a problem structure which commonly arises in multiple object tracking, similar to that examined in [3]. Suppose that we have a number of objects, numbered $\{1, \dots, M\}$, each of which can be observed using any sensor time slot. We seek to construct an open loop plan of which object to observe with our sensor (or sensors) in each time slot. This entire plan could be executed, or the first few steps could be executed and an updated plan constructed (so-called Open Loop Feedback Control, [4]). To motivate this structure, consider a problem in which we use an airborne sensor to track objects moving on the ground beneath foliage. In some positions, objects will be in clear view and observations will yield accurate position information; in other positions, objects will be obscured by foliage and observations will be essentially uninformative. Within the time scale of a planning horizon, objects will move in and out of obscurity, and it will be preferable to observe objects during the portion of time in which they are expected to be in clear view.

Our algorithm exploits submodularity, the same property used to obtain performance guarantees for greedy heuristics in [5, 4]. Submodularity captures the notion that as we select more observations, the value of the remaining unselected observations decreases, *i.e.*, the notion of diminishing returns.

Definition 1. A set function f is submodular if $f(C \cup \mathcal{A}) - f(\mathcal{A}) \geq f(C \cup \mathcal{B}) - f(\mathcal{B}) \forall \mathcal{B} \supseteq \mathcal{A}$.

It was established in [5] that, assuming that observations are independent conditioned on the quantity to be estimated, MI is a submodular function of the observation selection set.

¹Note that when we condition on a random variable (such as a yet unrealized observation) the conditional entropy involves an expectation over the distribution of that random variable.

This work was supported by MIT Lincoln Laboratory through ACC PO#3019934.

The simple result that we will utilize from submodularity is that $I(x; z^C | z^A) \geq I(x; z^C | z^B) \forall B \supseteq A$.

2. INTEGER PROGRAMMING FORMULATION

The emphasis of our formulation is to exploit the structure which results in sensor management problems involving observation of multiple independent objects. In addition to the above assumption that observations should be independent conditioned on the state, three new assumptions regarding the objects' states must be met for this structure to arise: (1) the prior distribution over object states must be independent; (2) the objects must evolve according to independent dynamical processes; and (3) the objects must be observed through independent observation processes. The first two assumptions are not overly limiting in multi-object tracking problems. The third, which is often violated (*e.g.*, due to data association), is made as an approximation for planning purposes. In circumstances involving strong dependency between small numbers of objects (out of a large total number), dependent objects may be considered as a collective *hyper-object*, and independence of the hyper-objects remains.

When the three assumptions are met, the mutual information reward of observations of different objects becomes the sum of the individual observation rewards. Denoting by $X^i = \{x_1^i, \dots, x_N^i\}$ the joint state (over the N -step planning horizon) of object i , we define the reward of observation set $\mathcal{A}^i \subseteq \{1, \dots, N\}$ of object i (*i.e.*, \mathcal{A}^i represents the subset of time slots in which we observe object i) to be:

$$r_{\mathcal{A}^i}^i = I(X^i; z_{\mathcal{A}^i}^i) \quad (1)$$

where $z_{\mathcal{A}^i}^i$ are the random variables corresponding to the observations of object i in the time slots in \mathcal{A}^i . Under the preceding assumptions, we can write the reward of choosing observation set \mathcal{A}^i for object $i \in \{1, \dots, M\}$ as:

$$\begin{aligned} & I(X^1, \dots, X^M; z_{\mathcal{A}^1}^1, \dots, z_{\mathcal{A}^M}^M) \\ &= \sum_{i=1}^M I(X^i; z_{\mathcal{A}^i}^i) = \sum_{i=1}^M r_{\mathcal{A}^i}^i \end{aligned} \quad (2)$$

As a slight generalization, let \mathcal{R} denote the set of sensing resources that are available (assumed finite). The elements of \mathcal{R} may correspond to different time slots of the same sensor, the same time slot of different sensors, or combinations of both. We assume that each element of \mathcal{R} may be assigned at most one task (although this can be easily generalized, as shown in [4]). As another slight generalization, let $\mathcal{U}^i = \{u_1^i, \dots, u_{L_i}^i\}$ be the set of elemental observation actions (assumed finite) that may be used for object i , where each elemental observation u_j^i corresponds to observing object i using a particular mode of a particular sensor within a particular period of time. An elemental action may occupy multiple resources; let $t(u_j^i) \subseteq \mathcal{R}$ be the subset of resources consumed by the elemental observation action u_j^i . Let

$\mathcal{S}^i \subseteq 2^{\mathcal{U}^i}$ be the collection of observation subsets which we allow for object i . This is assumed to take the form of Eq. (3), consisting simply of all subsets of \mathcal{U}^i for which no two elements consume the same resource:

$$\mathcal{S}^i = \{\mathcal{A} \subseteq \mathcal{U}^i | t(u_1) \cap t(u_2) = \emptyset \forall u_1, u_2 \in \mathcal{A}\} \quad (3)$$

We denote by $t(\mathcal{A}) \subseteq \mathcal{R}$ the set of resources consumed by the actions in set \mathcal{A} , *i.e.*, $t(\mathcal{A}) = \bigcup_{u \in \mathcal{A}} t(u)$.

The problem that we (conceptually) seek to solve is that of selecting the set of observation actions for each object such that the total reward is maximized subject to the constraint that each resource can be used at most once:

$$\max_{\omega_{\mathcal{A}^i}^i} \sum_{i=1}^M \sum_{\mathcal{A}^i \in \mathcal{S}^i} r_{\mathcal{A}^i}^i \omega_{\mathcal{A}^i}^i \quad (4a)$$

$$\text{s.t.} \sum_{i=1}^M \sum_{\substack{\mathcal{A}^i \in \mathcal{S}^i \\ t \in t(\mathcal{A}^i)}} \omega_{\mathcal{A}^i}^i \leq 1 \quad \forall t \in \mathcal{R} \quad (4b)$$

$$\sum_{\mathcal{A}^i \in \mathcal{S}^i} \omega_{\mathcal{A}^i}^i = 1 \quad \forall i \in \{1, \dots, M\} \quad (4c)$$

$$\omega_{\mathcal{A}^i}^i \in \{0, 1\} \quad \forall i, \mathcal{A}^i \in \mathcal{S}^i \quad (4d)$$

The binary indicator variables $\omega_{\mathcal{A}^i}^i$ are 1 if the observation set \mathcal{A}^i is chosen for object i and 0 otherwise. The constraints in Eq. (4b) ensure that each resource (*e.g.*, sensor time slot) is used at most once. The constraints in Eq. (4c) ensure that exactly one observation set is chosen for any given object; this is necessary to ensure that the additive objective is the exact reward of corresponding selection (since, in general, $r_{\mathcal{A} \cup \mathcal{B}}^i \neq r_{\mathcal{A}}^i + r_{\mathcal{B}}^i$).

The problem is not a pure assignment problem, as the observation subsets $\mathcal{A}^i \in \mathcal{S}^i$ consume multiple resources and hence appear in more than one of the constraints defined by Eq. (4b). The problem is actually a bundle assignment problem, and conceptually could be addressed using combinatorial auction methods (*e.g.*, [6]). However, generally this would require computation of $r_{\mathcal{A}^i}^i$ for every subset $\mathcal{A}^i \in \mathcal{S}^i$. If the collections of observation sets $\mathcal{S}^i, i \in \{1, \dots, M\}$ allow for several observations to be taken of the same object, the number of subsets may be combinatorially large.

3. CONSTRAINT GENERATION APPROACH

This section outlines the approach we propose, which in many practical situations can provide an efficient solution of the integer program in Eq. (4). The algorithm, which is described in detail in [4], proceeds by sequentially solving a series of integer programs with progressively greater complexity. In the limit, we arrive at the full complexity of the integer program in Eq. (4), but in many practical situations it is possible to terminate much sooner with an optimal solution. By simultaneously lowering an upper bound on the optimal solution, and

raising a lower bound on the optimal solution (which comes alongside a solution attaining the lower bound), we can also terminate early with a solution that is guaranteed to be within a given fraction of optimality.

The formulation may be conceptually understood as dividing the collection of subsets for each object (\mathcal{S}^i) at iteration l into two collections: $\mathcal{T}_l^i \subseteq \mathcal{S}^i$ and the remainder $\mathcal{S}^i \setminus \mathcal{T}_l^i$. The subsets in \mathcal{T}_l^i are those for which the exact reward has been evaluated; we will refer to these as *candidate subsets*. The reward of each of the remaining subsets (*i.e.*, those in $\mathcal{S}^i \setminus \mathcal{T}_l^i$) has not been evaluated, but an upper bound to each reward is available. In practice, we will not explicitly enumerate the elements in $\mathcal{S}^i \setminus \mathcal{T}_l^i$; rather we use a compact representation which implicitly considers all elements on the basis of upper bounds obtained using submodularity. The compact representation of $\mathcal{S}^i \setminus \mathcal{T}_l^i$ associates with each candidate subset, $\mathcal{A}^i \in \mathcal{T}_l^i$, a subset of observation actions, $\mathcal{B}_{l,\mathcal{A}^i}^i$; \mathcal{A}^i may be augmented with any subset of $\mathcal{B}_{l,\mathcal{A}^i}^i$ to generate new subsets that are not in \mathcal{T}_l^i (but that are in \mathcal{S}^i). We refer to $\mathcal{B}_{l,\mathcal{A}^i}^i$ as an *exploration subset*, since it provides a mechanism for discovering promising new subsets that should be incorporated into \mathcal{T}_{l+1}^i . The additional reward for selecting an exploration subset element $u \in \mathcal{B}_{l,\mathcal{A}^i}^i$ when the candidate subset \mathcal{A}^i is already selected is $r_{u|\mathcal{A}^i}^i \triangleq r_{\mathcal{A}^i \cup \{u\}}^i - r_{\mathcal{A}^i}^i$. By submodularity, $\forall C \subseteq \mathcal{B}_{l,\mathcal{A}^i}^i$,

$$r_{\mathcal{A}^i \cup C}^i \leq r_{\mathcal{A}^i}^i + \sum_{u \in C} r_{u|\mathcal{A}^i}^i$$

Equality will hold if $|C| \leq 1$.²

To initialize the problem, we select $\mathcal{T}_0^i = \{\emptyset\}$, and $\mathcal{B}_{0,\emptyset}^i = \mathcal{U}^i$ for all i . The integer program that we solve at each stage is:

$$\max_{\omega_{\mathcal{A}^i}^i, \omega_{u|\mathcal{A}^i}^i} \sum_{i=1}^M \sum_{\mathcal{A}^i \in \mathcal{T}_l^i} \left[r_{\mathcal{A}^i}^i \omega_{\mathcal{A}^i}^i + \sum_{u \in \mathcal{B}_{l,\mathcal{A}^i}^i} r_{u|\mathcal{A}^i}^i \omega_{u|\mathcal{A}^i}^i \right] \quad (5a)$$

$$\text{s.t.} \sum_{i=1}^M \sum_{\substack{\mathcal{A}^i \in \mathcal{T}_l^i \\ t \in t(\mathcal{A}^i)}} \omega_{\mathcal{A}^i}^i + \sum_{i=1}^M \sum_{\substack{\mathcal{A}^i \in \mathcal{T}_l^i \\ t \in t(u)}} \sum_{u \in \mathcal{B}_{l,\mathcal{A}^i}^i} \omega_{u|\mathcal{A}^i}^i \leq 1 \quad \forall t \in \mathcal{R} \quad (5b)$$

$$\sum_{\mathcal{A}^i \in \mathcal{T}_l^i} \omega_{\mathcal{A}^i}^i = 1 \quad \forall i \in \{1, \dots, M\} \quad (5c)$$

$$\sum_{u \in \mathcal{B}_{l,\mathcal{A}^i}^i} \omega_{u|\mathcal{A}^i}^i - |\mathcal{B}_{l,\mathcal{A}^i}^i| \omega_{\mathcal{A}^i}^i \leq 0 \quad \forall i, \mathcal{A}^i \in \mathcal{T}_l^i \quad (5d)$$

²We assume throughout that $\mathcal{B}_{l,\mathcal{A}^i}^i \cap \mathcal{A}^i = \emptyset$; our algorithm for constructing $\mathcal{B}_{l,\mathcal{A}^i}^i$ will ensure that this is the case.

$$\omega_{\mathcal{A}^i}^i \in \{0, 1\} \quad \forall i, \mathcal{A}^i \in \mathcal{T}^i \quad (5e)$$

$$\omega_{u|\mathcal{A}^i}^i \in \{0, 1\} \quad \forall i, \mathcal{A}^i \in \mathcal{T}^i, u \in \mathcal{B}_{l,\mathcal{A}^i}^i \quad (5f)$$

The solution of the integer program selects the subset for each object that maximizes the upper bound, ensuring that the resource constraints (*e.g.*, Eq. (5b)) are satisfied. The observation subset selected for object i is the set \mathcal{A}^i for which $\omega_{\mathcal{A}^i}^i = 1$, augmented by any additional observations u for which $\omega_{u|\mathcal{A}^i}^i = 1$. If the subset that the integer program selects for each object i is in \mathcal{T}_l^i —*i.e.*, it is a subset which had been generated and for which the exact reward had been evaluated in a previous iteration—then we have found an optimal solution to the original problem, *i.e.*, Eq. (4). This occurs when no more than one exploration subset element is chosen for each and every object. Conversely, if the integer program selects a subset in $\mathcal{S}^i \setminus \mathcal{T}_l^i$ for one or more objects, then we need to tighten the upper bounds on the rewards of those subsets, *e.g.*, by adding the newly selected subsets to \mathcal{T}_l^i in the next iteration and evaluating their exact rewards. This occurs when two or more exploration subset elements are chosen for any object. Each iteration of the optimization reconsiders all decision variables, allowing the solution from the previous iteration to be augmented or reversed in any way.

The algorithm used to update the candidate subsets \mathcal{T}_l^i and exploration subsets $\mathcal{B}_{l,\mathcal{A}^i}^i$ between iterations ensures that the upper bounds are tightened at each iteration. The algorithm and its theoretic characteristics are explored in detail in [4]. At each iteration l , a new candidate subset is introduced for each object for which more than two exploration subset elements were selected. The new candidate subset consists of the previously selected subset \mathcal{A}^i , augmented with the exploration subset element (among those selected) u^* with the greatest incremental reward. The exploration subset for \mathcal{A}^i candidate subset is updated such that u^* is removed, *i.e.*, $\mathcal{B}_{l+1,\mathcal{A}^i}^i = \mathcal{B}_{l,\mathcal{A}^i}^i \setminus \{u^*\}$. The exploration subset for the new candidate subset $\mathcal{A}^i \cup \{u^*\}$ is set to the same subset (*i.e.*, $\mathcal{B}_{l+1,\mathcal{A}^i \cup \{u^*\}}^i = \mathcal{B}_{l+1,\mathcal{A}^i}^i$).³

3.1. Augmented integer program

In the previous section we described a sequence of integer programs which form a progressively tighter upper bound to the solution of the full complexity integer program in Eq. (4). In each iteration, we also solve an augmented integer program, which provides the best solution amongst all solutions for which the exact reward has been evaluated (*i.e.*, a lower bound to the reward of the optimal solution); this is formed simply by adding to Eq. (5) constraints that prevent selection of more than one exploration subset element for any object. The reward of this augmented integer program is a non-decreasing function of iteration number. By combining this

³Actually, any elements which cannot be selected alongside u^* (*e.g.*, due to resource constraints) are removed from $\mathcal{B}_{l+1,\mathcal{A}^i \cup \{u^*\}}^i$.

best solution with the upper bound produced by the previously described constraint generation algorithm, we can terminate when we are within a desired tolerance of optimality.

3.2. Comments

On the surface, our algorithm bears some similarity to the recent work [7], which also solves a sensor resource management problem through an iterative solution of integer programs. However, the solution methodology in [7] is contingent on the cost criterion yielding a relaxation to a convex minimization. The obvious relaxation of information theoretic selection problems is a convex *maximization* [4] (a geometry for which few useful tools exist), hence it is unclear how to apply the approach in [7] when an information theoretic criterion is used.

4. EXPERIMENTAL RESULTS

The algorithm was implemented using C++, solving the integer programs using ILOG[®] CPLEX[®] 10.1 through the callable library interface. Termination occurs when the solution of the augmented integer program is guaranteed to be within 95% of optimality.

4.1. Multiple object tracking

Our first example models surveillance of multiple objects by a radar platform moving in a fixed racetrack pattern. Observation noise increases when objects become close to each other: this is a surrogate for the impact of data association, although we do not model the dependency between objects which generally results. We denote by \mathbf{y}_k the state (*i.e.*, position and velocity) of the sensing platform at time k . There are M objects under track, the states of which evolve according to a nominally constant velocity model:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k + \mathbf{w}_k \quad (6)$$

where \mathbf{w}_k is a discrete time zero-mean Gaussian white noise process with covariance

$$\mathbf{Q} = q \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ 0 & 0 & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \quad (7)$$

with $\Delta t = 0.01$ sec, and $q = 0.25$. The simulation runs for 100 time slots. The initial positions of the objects are distributed uniformly on the region $[10, 100] \times [10, 100]$; velocity magnitudes are drawn from a Gaussian distribution with mean 30 and standard deviation 0.5, while the velocity directions

are distributed uniformly on $[0, 2\pi]$. The initial estimates are set to the true state, corrupted by additive Gaussian noise with zero mean and standard deviation 0.02 (in position states) and 0.1 (in velocity states). In each time slot, the sensor may observe one of the M objects, obtaining either an azimuth and range observation, or an azimuth and range rate observation, each of which occupies a single time slot:

$$\mathbf{z}_k^{i,r} = \begin{bmatrix} \tan^{-1} \left(\frac{[\mathbf{x}_k^i - \mathbf{y}_k]_3}{[\mathbf{x}_k^i - \mathbf{y}_k]_1} \right) \\ \sqrt{([\mathbf{x}_k^i - \mathbf{y}_k]_1)^2 + ([\mathbf{x}_k^i - \mathbf{y}_k]_3)^2} \end{bmatrix} + d^i(\mathbf{x}_k^1, \dots, \mathbf{x}_k^M) \begin{bmatrix} b(\mathbf{x}_k^i, \mathbf{y}_k) & 0 \\ 0 & 1 \end{bmatrix} \mathbf{v}_k^{i,r} \quad (8)$$

$$\mathbf{z}_k^{i,d} = \begin{bmatrix} \tan^{-1} \left(\frac{[\mathbf{x}_k^i - \mathbf{y}_k]_3}{[\mathbf{x}_k^i - \mathbf{y}_k]_1} \right) \\ \frac{[\mathbf{x}_k^i - \mathbf{y}_k]_1 [\mathbf{x}_k^i - \mathbf{y}_k]_2 + [\mathbf{x}_k^i - \mathbf{y}_k]_3 [\mathbf{x}_k^i - \mathbf{y}_k]_4}{\sqrt{([\mathbf{x}_k^i - \mathbf{y}_k]_1)^2 + ([\mathbf{x}_k^i - \mathbf{y}_k]_3)^2}} \end{bmatrix} + d^i(\mathbf{x}_k^1, \dots, \mathbf{x}_k^M) \begin{bmatrix} b(\mathbf{x}_k^i, \mathbf{y}_k) & 0 \\ 0 & 1 \end{bmatrix} \mathbf{v}_k^{i,d} \quad (9)$$

where $\mathbf{z}_k^{i,r}$ denotes the azimuth/range observation for object i at time k , and $\mathbf{z}_k^{i,d}$ denotes the azimuth/range rate (*i.e.*, Doppler) observation. The notation $[\mathbf{a}]_l$ denotes the l -th element of the vector \mathbf{a} ; the first and third elements of the object state \mathbf{x}_k^i and the sensor state \mathbf{y}_k contain the position in the x -axis and y -axis respectively, while the second and fourth elements contain the velocity in the x -axis and y -axis respectively. The observation noises $\mathbf{v}_k^{i,r}$ and $\mathbf{v}_k^{i,d}$ are independent white Gaussian noise processes with zero mean and independent elements. The standard deviation of the noise on the azimuth observations (σ_ϕ) is 3° ; the multiplier function $b(\mathbf{x}_k^i, \mathbf{y}_k)$ varies from unity on the broadside (*i.e.*, when the sensor platform heading is perpendicular to the vector from the sensor to the object) to $\frac{1}{3}$ end-on. The standard deviation of the range observation (σ_r) is 0.1 units, while the standard deviation of the range rate observation (σ_d) is 0.075 units/sec. The function $d(\mathbf{x}_k^1, \dots, \mathbf{x}_k^M)$ captures the increase in observation noise when objects are close together:

$$d^i(\mathbf{x}_k^1, \dots, \mathbf{x}_k^M) = \sum_{j \neq i} \delta \left(\sqrt{([\mathbf{x}_k^i - \mathbf{x}_k^j]_1)^2 + ([\mathbf{x}_k^i - \mathbf{x}_k^j]_3)^2} \right)$$

where $\delta(x) = 10 - x$ for $0 \leq x \leq 10$ and $\delta(x) = 0$ otherwise. The state dependent noise is handled in a manner similar to the optimal linear estimator for bilinear systems, in which we estimate the variance of the observation noise, and then use this in a conventional linearized Kalman filter (for reward evaluations for planning) and extended Kalman filter (for estimation). In addition to the option of these two observations, the sensor can also choose a more accurate observation that takes three time slots to complete, and is not subject increased noise when objects become closely spaced. The azimuth noise for these observations in the broadside aspect has $\sigma_\phi = 0.6^\circ$, while the range noise has $\sigma_r = 0.02$ units, and the range rate noise has $\sigma_d = 0.015$ units/sec.

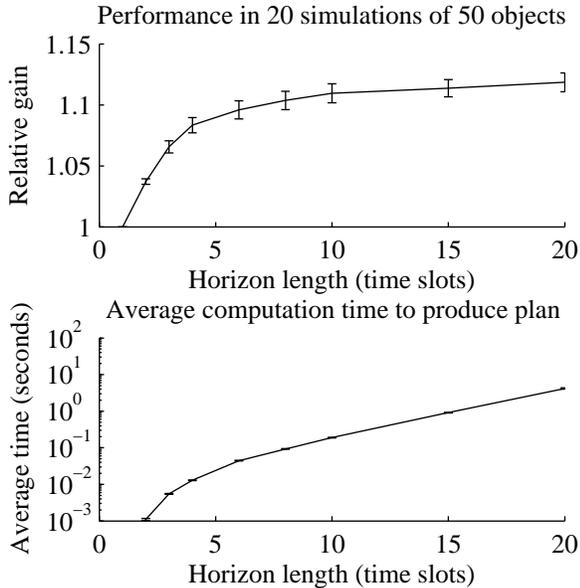


Fig. 1. Top diagram shows the total reward for each planning horizon length divided by the total reward for a single step planning horizon, averaged over 20 simulations. Error bars show the standard deviation of the mean performance estimate. Lower diagram shows the average time required to produce plan for different planning horizon lengths.

The results of the simulation are shown in Fig. 1. When the planning horizon is less than three time steps, the controller does not have the option of the three time step observation available to it. A moderate gain in performance is obtained by extending the planning horizon from one time step to three time steps to enable use of the longer observation. The increase is roughly doubled as the planning horizon is increased, allowing the controller to anticipate periods when observations for some objects are poor. As expected, the complexity increases exponentially with the planning horizon length. However, using the algorithm it is possible to produce a plan for 50 objects over 20 time slots using a few seconds in computation time. Performing the same planning through full enumeration would involve evaluation of the reward of more than 10^{40} different candidate sequences, a computation which is intractable on any foreseeable computational hardware.

4.2. Example of possible benefit

The scenario we discuss here demonstrates the increase in performance which is possible through long planning horizons when observations occupy different numbers of time slots. In such circumstances, algorithms utilizing short-term planning may make choices that preclude selection of later observations that may be arbitrarily more valuable. The scenario involves $M = 50$ objects observed

using a single sensor through a linear Gaussian observation model. The initial distribution of the objects is jointly Gaussian, where all objects are independent with covariance \mathbf{I} . In each time slot, a single object may be observed through either of two linear Gaussian observations (*i.e.*, of the form $\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$, where $\mathbf{v} \sim \mathcal{N}\{0, \mathbf{R}\}$). The first, which occupies a single time slot, has $\mathbf{H}_k^{i,1} = \mathbf{I}$, and $\mathbf{R}_k^{i,1} = 2\mathbf{I}$. The second, which occupies five time slots, has $\mathbf{H}_k^{i,2} = \mathbf{I}$, and $\mathbf{R}_k^{i,2} = r_k\mathbf{I}$. The noise variance of the longer observation, r_k , varies periodically with time (k), according to $r_k = 10^{-\text{mod}(k-1,5)-1}$ (the time index k commences at $k = 1$). Unless the planning horizon is sufficiently long to anticipate the availability of the observation with noise variance 10^{-5} several time steps later, the algorithm will select an observation with lower reward, which precludes selection of this later more accurate observation.

The performance is examined in detail in [4]. As the planning horizon increases from a single time slot (*i.e.*, myopic) to 50 time slots, the performance (reward) increases by a factor of $4.7\times$. The computation time required to produce a plan for 50 time slots is on the order of tens of milliseconds. While this is an extreme example, it illustrates an occasion when planning is highly beneficial (when there are observations that occupy several time slots with time varying rewards), and that the proposed algorithm is able to efficiently solve large planning problems in such a situation. An algorithm utilizing short-term planning in such circumstances may make choices that preclude selection of later observations which may be arbitrarily more valuable.

5. REFERENCES

- [1] K.J. Hintz and E.S. McVey, "Multi-process constrained estimation," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, no. 1, pp. 237–244, 1991.
- [2] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory*, John Wiley and Sons, New York, NY, 1991.
- [3] V. Krishnamurthy and R.J. Evans, "Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking," *Signal Processing, IEEE Transactions on*, vol. 49, no. 12, pp. 2893–2908, December 2001.
- [4] Jason L. Williams, *Information Theoretic Sensor Management*, Ph.D. thesis, Massachusetts Institute of Technology, February 2007, Available online at <http://ssg.mit.edu/group/jlwil/publications/Thesis.pdf>.
- [5] Andreas Krause and Carlos Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Uncertainty in Artificial Intelligence*, July 2005.
- [6] David C. Parkes and Lyle H. Ungar, "Iterative combinatorial auctions: Theory and practice," in *Proc 17th National Conference on Artificial Intelligence (AAAI)*, 2000, pp. 74–81.
- [7] Amit S. Chhetri, Darryl Morrell, and Antonia Papandreou-Suppappola, "Sensor resource allocation for tracking using outer approximation," *IEEE Signal Processing Letters*, vol. 14, no. 3, pp. 213–216, Mar. 2007.