Value-Based Tasking Controllers for Sensing Assets

William M. McEneaney, Ali Oran and Andrew Cavender[*][†]

Abstract

Sensing in support of combat teams will be increasingly performed by UAVs. With the accelerating pace of modern combat operations, and the developing networked-management of such, the problem of how to task sensor assets is entering the realm where automated decision-support tools should be applied. The objective by which the relative value of possible future sensor tasks should be compared is the value that they bring to the combat operation, with adjustment by the possible costs of these sensing operations. A mathematical theory allowing for the determination of the control-problem value as a function of information state (where this information state is described by a probability distribution) has recently been obtained. Possible future sensing tasks are mapped into the (stochastic) observation outcomes, and these are further mapped into potential a posteriori probability distributions. The value to the combat operation is the expectation of the minimax value as a function of these potential future probability distributions. This is used as an objective function, according to which optimal sensing-platform tasking is computed. Both open-loop and observation-feedback sensing-platform tasking controllers are developed. Importantly, a new, and quite general, representation of the form of such a value function is obtained. The value takes the form of a pointwise minimum of linear functionals. Further, this form is retained through dynamic programming backward propagation for

[*] Research partially supported by AFOSR grant FA9550-06-1-0238 and Northrop-Grumman.

[†] All authors are with the Dept. of Mechanical and Aerospace Engineering, University of California San Diego, San Diego, CA 92093-0411, USA. wmcenean@ucsd.edu, aoran@ucsd.edu, acavende@ucsd.edu

solution of the problem. This representation will allow for the solution of high-dimensional problems in tasking of sensing UAVs. A simple, illustrative example is shown.

1 Introduction

Sensing in support of combat teams will be increasingly performed by UAVs. With the accelerating pace of modern combat operations, and the developing networked-management of such, the problem of how to task sensor assets is entering the realm where automated decision-support tools should be applied.

This problem falls into the category that is sometimes referred to as observation control. Earlier, mathematically strong research on this general area may be found in Miller and Runggaldier [10]. In that case, a different measure of information value (covariance) was used, as well as a different dynamics model. Questions of observation control have arisen more recently in the context of control of groups of UAVs and UUVs, viewed as mobile sensor networks. Such an approach is considered in [4], where the authors look at this problem in the context of underwater vehicles. Similarly, in [5] a related problem with air vehicles is considered. In both of those cases, the goal of the sensing operation was very different from that considered here. Here, we are concerned with optimizing expected observation value, where that value is through the effect the information has on a related partial-information game or control problem, which involves a separate set of entities which are consuming the observation-induced information. In the application here, this related control problem consists of combat operations. The nature of this related problem figures very heavily into the solvability of the observation control problem. In particular, it implies a specific structure (convex, piecewise linear) for the value-of-information function, and this structure is exploited.

The correct measure of value for sensor UAV tasking consists of two components. The first is simply the expected cost of the sensing action (expected cost of loss and maintenance). The second component is much more important, and it is the expected payoff to the warfighters of the possible observational data returns. In [6, 7], the authors develop an object, $V_t(q)$ which describes the minimax expected payoff for a game between Blue and Red combat teams at time t as a function of the Blue knowledge of the system state, specified as probability distribution q. (It is assumed that Red has perfect state information.)

This object may be used to determine the value of sensing actions. For discussion purposes, suppose a simple, decomposed problem is given as follows. Suppose Blue will choose a sensing control action, ending at time, $t = T > 0$, immediately followed by a combat action. At time, $t = 0$, Blue knowledge is described by q_0. Given a series of sensing actions, $u^o_{[0,T-1]} \doteq \{u^o_t\}_{t=0}^{T-1}$, there is an associated set of possible observations, $\{y_t(u^o_t)\}_{t=0}^{T-1}$. Note that the y_t are random variables – the actual observation that will be obtained will be corrupted by noise. Given such a set of observations, one may update the distribution q_0 to q_T by an estimator such as Bayes rule. Note that q_T is a random variable. The obvious resulting payoff is $V_T(q_T)$.

One may use this payoff to formulate this sensing operation as an optimization problem. Select $u^o_{[0,T-1]}$ to maximize payoff

$$J(u^o_{[0,T-1]}) = \mathbf{E}\left\{V_T(q_T) - C(u^o_{[0,T-1]})\right\},$$

where $C(u^o_{[0,T-1]})$ is a random variable describing the possible UAV (or other sensing platform) losses due to the control choice.

Of course, the sensing actions will occur, not only before, but in parallel with the combat actions. Further, the optimal sensing action at the next time-step could (and almost certainly would) depend on the observations thus far obtained. Consequently, the above optimization-problem format generalizes to a control problem.

The key to reasonable computational speed with this approach is an ability to model the probability of kill against Blue, $\rho = \rho(q)$, as a function of Blue's information state (where the argument indicates that this depends on probability distribution q, i.e., the information state) for each micro-action of the future engagements. Micro-actions are essentially single steps on an abstract graph-based model of combat force movement. More specifically, Blue forces will move on the nodes of this graph, where each node represents a sub-region of the terrain. Let q^l be a probability distribution over the l^{th} sub-region of the terrain, representing the unknown Red force locations in that sub-region. We will demonstrate that the probability of kill for that sub-regoin, $\rho^l(q^l)$, takes the form of a concave piecewise-linear function. See left-hand image of Figure 1 for an example. These forms can be pre-computed as functions of the local geometry. The survival probability of a Blue entity over a larger set of actions is simply the product of the survival probabilities for the constituent micro-actions (each taking the form $1 - \rho^i(q^l)$). Thus, the cost criterion to be optimized in the UAV-tasking control problem is a function of products of these forms. The theory of dynamic programming is used to develop the general algorithm. We then demonstrate that a specific representation for the value as a product of convex (due to the use of survival probabilities rather than probabilities of kill) piecewise linear functions is preserved under the backward dynamic programming propagation. This form is exploited to generate a new type of algorithm, which will allow us to solve high-dimensional problems. A very simple example of an optimal path can be seen in right-hand image of Figure 1, where the blue path indicates movement of the Blue combat ground forces, and the green path indicates the optimal tasking path of a single supporting UAV sensor platform. The red dots indicate potential locations of Red forces.

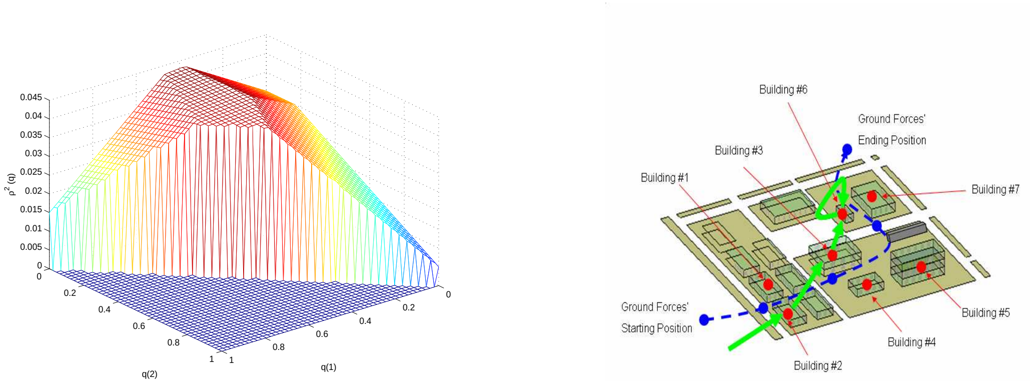

Figure 1: $\rho^l(q)$ for a micro-action and optimal sensing-support route

2 Modeling the Value of Information

Prior to solving the UAV tasking problem, we will determine the form of the function describing the value of information, where this information state will take the form of a probability distribution. This form for the value of information, which will be obtained in this section, will be exploited when we solve the UAV tasking control problem in the sections to follow.

The (unknown) state of the Red forces will be modeled as a set of positions on an abstract graph. As this is a discrete object, the corresponding probability distribution will be a vector of probabilities. The Blue combat force state will also be modeled as a discrete position or set of positions, moving on an abstract graph.

Each observation will propagate an a priori probability distribution representing Blue's knowledge into a resulting a posteriori distribution, via Bayes rule. In fact, this time-indexed sequence of probability distributions will represent the state of the system (of the observation-control problem). That is, the state-process we are interested in here consists of the probability distributions, q_t^l. Note that given a sequence of observation tasks, the resulting observations form a stochastic process (due to observation noise such as induced by false positives), and consequently, q_t^l is a stochastic process. In order to obtain the cost criterion for our observation control problem, we must look at the problem of determining the value of probability distributions, q_t^l.

We now begin to develop this value-of-distribution. *Keep in mind that this is not the value of the observation-control problem, but is the value of information, which will form the cost criterion of the observation-control problem to be discussed in the next section.* In Section 4 of [7], a minimax game value for Blue under partial information, modeled as q_t, was obtained. In that case, both Blue and Red were active players. However, for the problem under consideration here, we will not model them as such. The reason for this is that it is not generally military policy to allow a sensor-tasking controller to direct combat operations, and this latter task is (very reasonably) kept under human control. Consequently, we will assume that we are given a Blue COA (course-of-action) for combat operations, and that this will remain fixed, as far as our control problem is concerned.

For a first problem formulation, which is what we are studying here, we assume that this is a "Blue-attack mission" where Red is in some fixed, but unknown, set of positions. With such a model, and more specifically with a fixed Blue COA, it is not at first obvious how Blue would benefit from observational data. The key is to note that the Blue COA is a high-level controller, and that there exist finer actions which the local commander controls.

We use the following simple example to motivate the mathematical model, but the approach applies to a larger variety of Command and Control (C^2) problems. We suppose that the Blue forces are moving in urban terrain, see Figure 2. Red forces may lie in any of the buildings marked with a Red dot.

We break this problem into a simple three-step process in which Blue combat forces make three steps to reach their goal. In each step, they may come under attack from Red forces, see Figure 2. The three steps are indicated by the three ovals. We suppose that, at each step, Blue can be attacked only from the locations in the corresponding oval. (This is not essential to the theory, but simplifies the computations for us here.)

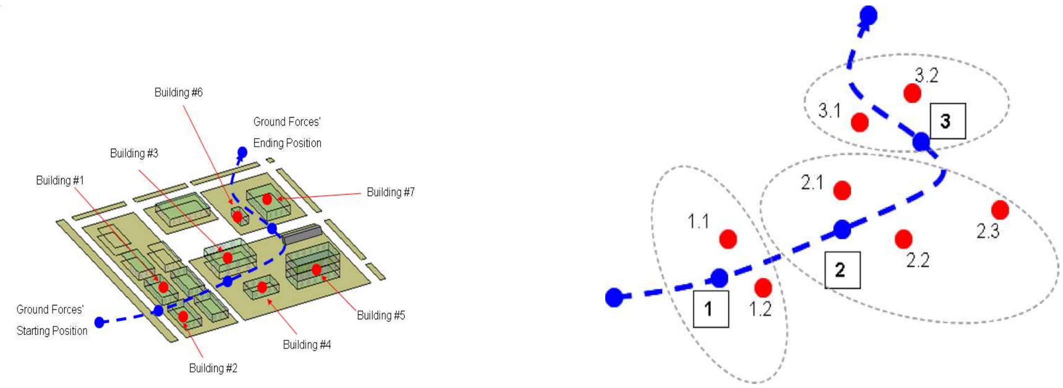

Figure 2: Blue COA and abstracted version

We continue to use this simple example problem in defining the underlying value-of-distribution/observation-control cost criterion. We suppose that in the k^{th} step, a single Blue combat entity (e.g., a fire-team) will be at $l_k \in \mathcal{L} = \{1, 2 \ldots L\}$ where $\mathcal{L}$ is the set of nodes of an abstract graph representing allowable movement in the battlespace, where as noted above, these nodes correspond to sub-regions of the terrain. (See, for example, [8] for a discussion of such a graph representation.) In the example depicted in Figure 2, $L = 3$. Let the possible Red locations in subregion $l \in \mathcal{L}$ be $\mathcal{N}_l = \{1, 2 \ldots N_l\}$. Suppose, for this study, that we know that there is exactly one Red unit in each subregion; otherwise the analysis is more complex, but not conceptually different. If Red is at location $n \in \mathcal{N}_l$, we say $x_l^R = n$. In the example depicted in Figure 2, $N_1 = 2$, $N_2 = 3$ and $N_3 = 2$.

The Blue information on the Red state in subregion l is described by probability distribution q^l. Note that distribution q^l lies in simplex S^{N_l}

where

$$S^n = \left\{ q \in I\!R^n \mid q_i \in [0,1] \, \forall i \leq n \text{ and } \sum_{i=1}^{n} q_i = 1 \right\}.$$

In particular, the n^{th} component of q^l, q_n^l, is the probability that there is a Red entity at location n of subregion l. The full Blue information state is

$$Q \doteq \{q^l\}_{l=1}^{L}. \tag{1}$$

Continuing the development using our motivational example, we suppose that the allowable *local, combat* Blue controls at node l are $U^l = \{0\} \cup \mathcal{N}_l$. In other words, at time-step k, Blue is at node l_k and may apply local control $u_k \in U^{l_k}$. In our example, $u_k = n \in \mathcal{N}_l$ implies that Blue is laying cover fire on potential Red location n, while proceeding with its Blue COA. Further, $u_k = 0$ implies that Blue is "tight" during this step, meaning that the local Blue entity does not fire unless fired upon during this step. We will make the assumption that Blue is well-protected if firing upon the correct potential Red location. Further, we will assume that Blue is somewhat better able to defend itself from an attack from location n' if in stance "tight" $(u = 0)$ rather than in stance $u = n \neq n'$ (with $n > 0$). The inclusion of control $u = 0$ is not necessary to the approach, but is included for additional realism.

The local Blue forces will have a health state $h \in \{0, 1\}$, where $h = 1$ represents healthy, and $h = 0$ represents destroyed. This simple model is sufficient for our purposes here; our goal is the development of the sensing-platform control. At each time-step, Blue wishes to maximize its probability of survival. Let $\rho^l : \mathcal{N}_l \times U^l \to [0,1]$, where specifically, $\rho_k^l(x_l^R, u)$ will be the probability that Blue is destroyed (h transitioning from 1 to 0) at node l given true Red position x_l^R and Blue local combat control u.

In order to obtain the value-of-information, we form a small control problem for each micro-action. The dynamics are those of the Blue entity health, given just above. The cost criterion will be

$$\begin{aligned} J^l(q,u) &\doteq \sum_{x \in \mathcal{N}_l} \rho^l(x,u) q_x^l \\ &= \mathbf{p}^l(u) \cdot q^l, \end{aligned} \tag{2}$$

where $\mathbf{p}^l(u)$ is the vector of length N_l with elements $\rho^l(x,u)$ for $x \in \mathcal{N}_l$. The value-of-information is the value of this small control problem, also referred to as a micro-action, and is given by

$$V^l(q) \doteq \min_{u \in U^l} \left\{ \mathbf{p}^l(u) \cdot q \right\}. \qquad (3)$$

We see that each V^l is a piecewise linear function of its argument, Blue's probability distribution of Red. Specifically, V^l maps S^{N_l} into $I\!R$.

The probability that Blue is destroyed at time-step k is $\widehat{P}_k(q) \doteq P_{l_k}(q) = V^{l_k}(q)$. Suppose there are K time-steps in the Blue COA. Then, the probability that the Blue entity survives to reach the goal is

$$\overline{P}_s(Q) \doteq \prod_{k=1}^{K} \left[1 - P_{l_k}(q^{l_k}) \right]. \qquad (4)$$

The goal of the sensing operation is to maximize the expected value (recalling that Q will be random) of $\overline{P}_s$ through the tasking control of the sensor platform(s), and this will be discussed in the next section. Prior to that we note that $P_l(q)$ may be computed offline and stored for use in the computations. Further beyond that, one may be able to simply store P_l functions indexed according to local geometry, rather than computing them for each geography encountered.

3 Observation-Control Problem

As noted above, we consider here the simplest case, where the Blue combat forces act *after* the completion of the sensing operations. For the more general problem, we consider the sensing actions operating in parallel with the combat operations, and as the speeds of the sensing and combat entities may be dissimilar, we need to use multiple time-scales which must be properly synchronized. In that case, we also include the information gained by the combat forces as they move through the physical space. These additional aspects (of parallel operations) induce considerable additional notation and technicalities which obscure the main point. Consequently, for this presentation, we assume the sensing actions take place first, and then the combat operations occur. With this simplified model, the observation-platform tasking control problem is a finite time-horizon, terminal-cost problem.

In order to simplify notation for the subsequent mathematical development, we now let information state at time t be q_t where $q_t : \mathcal{J} \to [0, 1]$

where $\mathcal{J}$ is the set of possible Red configurations, and $[q_t]_j \in [0,1]$ and $\sum_{j\in\mathcal{J}}[q_t]_j = 1$. (Recall that we are assuming that there is exactly one Red entity in each subregion. In the notation used for the combat operations subproblem of the last section, $\mathcal{J} = \{j_1, j_2, \dots j_L\}$ where each $j_l \in \{1, 2, \dots N_l\}$.) Then $q_t \in S^J$ where $J = \#\mathcal{J}$. Let the mapping from $q \in S^J$ to Q of (1) be denoted by $F : S^J \to S^{N_1} \times \dots \times S^{N_L}$, and define $\widetilde{P}_s(q) \doteq \overline{P}_s(F(q))$ for all $q \in S^J$. Let the initial information state (at sensing-platform control problem time $t = 0$) be q_0.

The expected costs due to maintenance and loss of sensing platforms, i.e. $\mathbf{E}\{C(u^o_\cdot)\}$, is easily modeled. We concentrate here on the, more interesting, payoff-component – the value for combat teams $\widetilde{P}_s(q)$, and ignore the $C(u^o_\cdot)$ component.

For simplicity, we suppose that the sensing platforms can move from any position to any other position in one time-step. With this freedom, any sequence of ordered pairs, (l, x) where $l \in \mathcal{L}$ and $x \in \mathcal{N}_l$, is an admissible sensor-platform control. Let $\mathcal{U}$ be the set of ordered pairs (l, x) where $l \in \mathcal{L}$ and $x \in \mathcal{N}_l$, i.e., $\mathcal{U} = \{(l, x) \,|\, l \in \mathcal{L}, x \in \mathcal{N}_l\}$.

Suppose there are T time-steps in the sensing-platform control problem. The payoff for information state q at time t with observation-platform control $u^o_\cdot$ is

$$J^{o,o}(t, q, u^o_\cdot) \doteq \mathbf{E}\left\{\widetilde{P}_s(q_T)\right\} \tag{5}$$

where

$$u^o_\cdot = u^o_{[t,T-1]} = \{u^o_t \in \mathcal{U} \,|\, t \in \{t, t+1, \dots T-1\}\,\}.$$

Note that $u^o_{[t,T-1]} \in \mathcal{U}^{T-t}$, where the superscript indicates outer product $T-t$ times. The propagation of the state from q to q_T is discussed further below.

We consider multiple approaches to the optimal control problem. The first is simply open-loop optimization. This would be appropriate for a concept-of-operations where incoming observational data could not be used to re-adjust the sensing-platform task plan. With such a model, the control problem reduces to an open-loop optimization problem. In particular, one solves for the value function

$$V^{o,o}(t, q) = \max_{u^o_\cdot \in \mathcal{U}^{T-t}} J^{o,o}(t, q, u^o_\cdot), \tag{6}$$

and the corresponding optimal task-plan.

The second approach is the true feedback case, in which the state at time t consists of the current sensor position and the current information state, q_t.

As the sensor can move from any location to any other in one time-step, we will suppress the sensor-position as state component. Let $\mathcal{A}_t \doteq \{\alpha : S^J \to \mathcal{U}\}$ where $J \doteq \#\mathcal{J}$. Let

$$\begin{aligned}\mathcal{A}^t \doteq \{ \quad & \alpha_{[t,T-1]} : [S^J]^{T-t} \to \mathcal{U}^{T-t} \,|\text{ if } q_r = \hat{q}_r \text{ for all} \\ & r \le \bar{t},\text{ then } \alpha_r[q.] = \alpha_r[\hat{q}.] \text{ for all } r \le \bar{t}\,\}.\end{aligned}$$

The payoff for information state $q_t = q$ and non-anticipative control $\alpha \in \mathcal{A}^t$ is

$$J^{o,f}(t,q,\alpha.) \doteq \mathbf{E}\left\{\widetilde{P}_s(q_T)\right\}, \tag{7}$$

and (again) the propagation of the state from q to q_T is discussed below. The corresponding value function is

$$V^{o,f}(t,q) = \sup_{\alpha.\in\mathcal{A}^t} J^{o,f}(t,q,\alpha.). \tag{8}$$

We have not yet indicated the dynamics of the state in these above definitions, and now do so. The dynamics of the information state are given by Bayes rule. More specifically, suppose the sensor is at (l,x) at time t. The observation $y = y_t$ will take a value in $\mathcal{Y} \doteq \{0,1\}$ where $y = 0$ indicates that no Red entity is observed at (l,x), and $y = 1$ indicates that a Red entity is observed there. (We recall that an "entity" may be a Red combat team.) Let $R_j^{y,u}$ be the probability of observation y given given the sensor is at $u \in \mathcal{U}$ and the Red force state is $j \in \mathcal{J}$. Let $\mathbf{R}^{y,u}$ be the vector of length J with components $R_j^{y,u}$, and let $D(\mathbf{R}^{y,u})$ be the $J \times J$ matrix with diagonal elements $[D(\mathbf{R}^{y,u})]_{j,j} = R_j^{y,u}$ and $[D(\mathbf{R}^{y,u})]_{i,j} = 0$ for $i \ne j$. Then, given any sensing control action $u_t \in \mathcal{U}$ and resulting (random-variable) observation y_t, one has

$$\begin{aligned} q_{t+1} \quad &= \frac{1}{\mathbf{R}^{y_t,u_t}\cdot q_t} D(\mathbf{R}^{y_t,u_t}) q_t, \\ &\doteq \beta^{y_t,u_t}(q_t) \end{aligned} \tag{9}$$

which defines the stochastic information state dynamics.

Using the above dynamics model, one may obtain computationally explicit forms for $J^{o,o}$ and $V^{o,o}$. Suppose one applies control sequence $u_{[0,T-1]}$, resulting in observation sequence $y_{[0,T-1]}$. The probability of any such sequence is

$$P(y_{[0,T-1]}) = \sum_{x\in\mathcal{J}} P(y_{[0,T-1]}|x) q_x. \tag{10}$$

Also, since the observation noises are (assumed) independent, one has

$$P(y_{[0,T-1]}|x) = \prod_{t=0}^{T-1} R_x^{y_t,u_t}. \tag{11}$$

Combining (10) and (11) yields

$$\begin{aligned} P(y_{[0,T-1]}) &= \sum_{x\in\mathcal{J}} \left[\prod_{t=0}^{T-1} R_x^{y_t,u_t}\right] q_x \\ &= \sum_{x\in\mathcal{J}} \left[\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q\right]_x . \end{aligned} \tag{12}$$

Also note that

$$q_T \quad = \beta^{y_{T-1},u_{T-1}} \circ \beta^{y_{T-2},u_{T-2}} \circ \cdots \circ \beta^{y_0,u_0}(q)$$

where $\circ$ indicates composition, and one can show that this is

$$= \frac{\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q}{\sum_{x\in\mathcal{J}} \left[\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q\right]_x}. \tag{13}$$

By (5), (12) and (13),

$$\begin{aligned} &J^{o,o}(t,q,u_\cdot^o) = \\ &\sum_{y_{[0,T-1]}\in\mathcal{Y}^T} \left[\widetilde{P}_s\left(\frac{\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q}{\sum_{x\in\mathcal{J}} \left[\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q\right]_x}\right)\right] \\ &\qquad \cdot \left[\sum_{x\in\mathcal{J}} \left[\left(\prod_{t=0}^{T-1} D(\mathbf{R}^{y_t,u_t})\right) q\right]_x\right]. \end{aligned} \tag{14}$$

One can easily check that, if $\widetilde{P}_s(q) = a^T q$ for some $a \in I\!R^J$, then

$$J^{o,o}(t,q,u_\cdot^o) = a^T q = \widetilde{P}_s(q),$$

in which case, the expected gain from any sensing would be zero. It is important to note that the fact that sensing has value is dependent on the nonlinearity of the dependence of the survival probability, $\widetilde{P}_s$ on q.

Now we return to the information state feedback case. The first thing we obtain is the dynamic programming principle (DPP):

Theorem 3.1 *For $t \in \{0, 1, \ldots T-1\}$,*

$$V^{o,f}(t,q) = \max_{u \in \mathcal{U}} \mathbf{E}_y \left\{ V^{o,f}(t+1, \beta^{y,u}(q)) \right\}$$

where the expectation is over the set of possible observations.

The proof is not included, but is of standard form.

We indicate how the backward DPP is mechanized. First, of course,

$$V^{o,f}(T,q) = \widetilde{P}_s(q).$$

Next, note that

$$V^{o,f}(t,q) \quad = \max_{u \in \mathcal{U}} \sum_{y \in \mathcal{Y}} P(y) V^{o,f}(t+1, \beta^{y,u}(q))$$

which, by the above exposition,

$$\begin{aligned} &= \max_{u \in \mathcal{U}} \sum_{y \in \mathcal{Y}} \Big\{ [\mathbf{R}^{y,u} \cdot q] \\ &\qquad \cdot V^{o,f} \left(t+1, \frac{1}{\mathbf{R}^{y,u} \cdot q} D(\mathbf{R}^{y,u}) q \right) \Big\}. \end{aligned} \tag{15}$$

4 Computation Complexity

The open-loop case is straightforward. The computation of $J^{o,o}(0, q, u.)$ is given by (14). Using this, given any q and any $u.$, one may compute $J^{o,o}$ directly. Of course, if $\#\mathcal{Y}^{T-1}$ is large, then this can become burdensome. Further, when optimizing over $u. \in \mathcal{U}^T$ to find the optimum, the calculations become yet more burdensome. Nonetheless, many terms are computed repeatedly, and so clever pre-computation of such terms outside the summation and optimization loops is greatly beneficial.

The feedback case is more demanding. Consider (15). In order to compute $V^{o,f}(t,q)$, one must have $V^{o,f}(t+1, \cdot)$ on S^J. If one began to think of this as a problem where one needed to consider propagation on the space of functions over the simplex $\mathcal{S}^N$ (an $N-1$-dimensional hyperplane), where $N = \#\bar{\mathcal{L}}\#\mathcal{H} = 2\#\bar{\mathcal{L}}$, then this propagation would certainly be computationally unfeasible. Fortunately, one may make use of the special form of $\widetilde{P}_s$ as a pointwise maximum of affine functions (implying the same for $V^{o,f}$, but we do not include the proof here). This allows us to operate on the value function parameterized as a maximum of relatively easily computed functions, thereby making the computations much more feasible.

Clearly $V^{o,f}(T,q)$ is of the form $V^{o,f}(t,q) = \max_{i\in\mathcal{I}_t} b^{i,t}\cdot q$. We need to show that this form is retained under the dynamic programming propagation. For any set, $\mathcal{I}$, and positive integer N, let $\mathcal{P}^N(\mathcal{I})$ be the set of all sequences of length N with elements from $\mathcal{I}$, and note that the cardinality of $\mathcal{P}^N(\mathcal{I})$ is $(\#\mathcal{I})^N$. For simplicity here, we relabel $\mathcal{Y}$ and $\mathcal{U}$ as $\mathcal{Y} = \{1,2,\dots N_y\}$ and $\mathcal{U} = \{1,2,\dots N_u\}$.

Theorem 4.1 *Suppose $V^{o,f}(t+1,q)$ takes the form*

$$V^{o,f}(t+1,q) = \max_{i\in\mathcal{I}_{t+1}} b^i_{t+1}\cdot q$$

where $\mathcal{I}_{t+1} = \{1,2,\dots I_{t+1}\}$. Then,

$$V^{o,f}(t,q) = \max_{i\in\mathcal{I}_t} b^i_t\cdot q$$

where $\mathcal{I}_t = \{1,2,\dots I_t\}$, $I_t = N_u(I_{t+1})^{N_y}$,

$$b^i_t = \sum_{y\in\mathcal{Y}} D(\mathbf{R}^{y,u}) b^{j_y}_{t+1} \tag{16}$$

where $(u,\{j_y\}) = \mathcal{M}^{-1}(i)$, and $\mathcal{M}$ is a one-to-one, onto mapping from $\mathcal{U}\times\mathcal{P}^{N_y}(\mathcal{I}_{t+1})\to\mathcal{I}_t$ (i.e., an indexing of $\mathcal{U}\times\mathcal{P}^{N_y}(\mathcal{I}_{t+1})$).

Proof. By (15) and the assumption, one has

$$\begin{aligned} V^{o,f} &= \max_{u\in\mathcal{U}} \sum_{y\in\mathcal{Y}} \left\{ [\mathbf{R}^{y,u}\cdot q] \max_{i\in\mathcal{I}_{t+1}} \left[b^i_{t+1}\cdot \frac{1}{\mathbf{R}^{y,u}\cdot q} D(\mathbf{R}^{y,u}) q \right] \right\} \\ &= \max_{u\in\mathcal{U}} \sum_{y\in\mathcal{Y}} \max_{i\in\mathcal{I}_{t+1}} \left\{ \left[D(\mathbf{R}^{y,u}) b^i_{t+1} \right]^T q \right\} \end{aligned}$$

which, upon defining $\bar{b}^{i,y,u,t} \doteq D(\mathbf{R}^{y,u}) b^i_{t+1}$, is

$$= \max_{u\in\mathcal{U}} \sum_{y\in\mathcal{Y}} \max_{i\in\mathcal{I}_{t+1}} \left\{ \left[\bar{b}^{i,y,u,t} \right]^T q \right\}.$$

The next step is most easily seen using the max-plus algebra notation, where we note that the max-plus algebra is the commutative semifield over

$I\!\!R \cup \{-\infty\}$ with operations $a \oplus b = \max\{a,b\}$ and $a \otimes b = a + b$ (c.f., [1], [2], [3], [9]). Using this notation, we have

$$V^{o,f}(t,q) \quad = \bigoplus_{u\in\mathcal{U}} \bigotimes_{y\in\mathcal{Y}} \bigoplus_{i\in\mathcal{I}_{t+1}} \left\{ \left[\bar{b}^{i,y,u,t}\right]^T q \right\},$$

which, using the max-plus distributive property,

$$= \bigoplus_{u\in\mathcal{U}} \bigoplus_{\{i_y\}\in\mathcal{P}^{N_y}(\mathcal{I}_{t+1})} \bigotimes_{y\in\mathcal{Y}} \left\{ \left[\bar{b}^{i_y,y,u,t}\right]^T q \right\}$$

where $\mathcal{P}^{N_y}(\mathcal{I}_{t+1}) = \{\, \{i_y\}_{y\in\mathcal{Y}} \,|\, i_y \in \mathcal{I}_{t+1}\, \forall y \in \mathcal{Y}\}$. Returning to our previous notation, this is

$$\begin{aligned} V^{o,f}(t,q) \quad &= \max_{u\in\mathcal{U}} \max_{\{i_y\}\in\mathcal{P}^{N_y}(\mathcal{I}_{t+1})} \left\{ \sum_{y\in\mathcal{Y}} \left[\bar{b}^{i_y,y,u,t}\right]^T q \right\} \\ &= \max_{(u,\{i_y\})\in\mathcal{U}\times\mathcal{P}^{N_y}(\mathcal{I}_{t+1})} \left\{ \sum_{y\in\mathcal{Y}} \left[\bar{b}^{i_y,y,u,t}\right]^T q \right\}. \qquad (17) \end{aligned}$$

Let $I_t \doteq N_u \left[\#\left(\mathcal{P}^{N_y}(\mathcal{I}_{t+1})\right)\right] = N_u \left[(\#\mathcal{I}_{t+1})^{N_y}\right]$, and define $\mathcal{I}_t \doteq \{1, 2, \dots I_t\}$. Then let $\mathcal{M} : \mathcal{U} \times \mathcal{P}^{N_y}(\mathcal{I}_{t+1}) \to \mathcal{I}_t$ be a one-to-one, onto map defining an indexing at time-step t, and let

$$b^{\mathcal{M}(u,\{i_y\}),t} \doteq \sum_{y\in\mathcal{Y}} \bar{b}^{i_y,y,u,t} = \sum_{y\in\mathcal{Y}} \left[D(\mathbf{R}^{y,u}) b_{t+1}^{i_y} \right].$$

Then, (17) is

$$V^{o,f}(t,q) = \max_{(u,\{i_y\})\in\mathcal{U}\times\mathcal{P}^{N_y}(\mathcal{I}_{t+1})} \left[b^{\mathcal{M}(u,\{i_y\}),t}\right]^T q = \max_{i\in\mathcal{I}_t} b^{i,t} \cdot q. \;\square$$

Corollary 4.2 *For all* $t \in \{0, 1, \dots T\}$,

$$V^{o,f}(t,q) = \max_{i\in\mathcal{I}_t} b^{i,t} \cdot q$$

for appropriate $\mathcal{I}_t$ *and set of* $b^{i,t}$*, obtained from the backward dynamic program.*

With this result, we see that backward propagation of $V^{o,f}$ reduces to backward propagation of the sets $\mathcal{B}_t = \{b_t^i : i \in \mathcal{I}_t\}$ using (16). This implies that one does not need to grid S^J, a technique which is subject to the curse-of-dimensionality. However, note that I_t will grow rapidly. This growth may be attenuated by judicious pruning of the $\mathcal{B}_t$, which may be effected by linear programs. (That is, the determination of the contribution of any particular b_t^i to the overall value function may be obtained from solving a simple linear program.) Thus, the use of Theorem 4.1 leads to much more computationally efficient schemes.

5 Example

We briefly present some results for the simple example indicated in the introduction.

In this example, the battlespace is divided into three regions as depicted abstractly in Figure 2. There is exactly one Red entity in each region. In each of Regions 1 and 3, there are exactly two buildings where Red may have entities. For simplicity, the parameters chosen for each of the two regions were identical. Computing the the probabilities of Blue entity loss in each of those regions according to the method of Section 2, results in a $P_l(q)$ depicted in Figure 3. Region 2 has three buildings, and the corresponding $P_l(q)$ was depicted above in the right-hand image of Figure 1. For the example studied here, the Blue sensing-UAV could visit two buildings prior to the start of combat operations. Thereafter, sensing actions and combat actions were interleaved, until the terminal time at the end of the third combat step (Region 3).

The value function for the open-loop case is depicted in Figure 3, as a function of the initial information in Regions 1 and 3. (Note that the information state is minimally stored as a vector in the four-dimensional unit hypercube, and so we only display it over two components – the probabilities that there is a Red entity in Building 1.1 and the probability there is a Red entity in Building 3.1 according to Figure 2 labeling.) We compare this with a heuristically generated sensing-platform task plan, which for any specific q, might be similar to what a commander would choose. The expected payoff for the heuristic task planner is depicted in Figure 4, and the percent improvement (in terms of reduced attrition) is depicted in Figure 4.

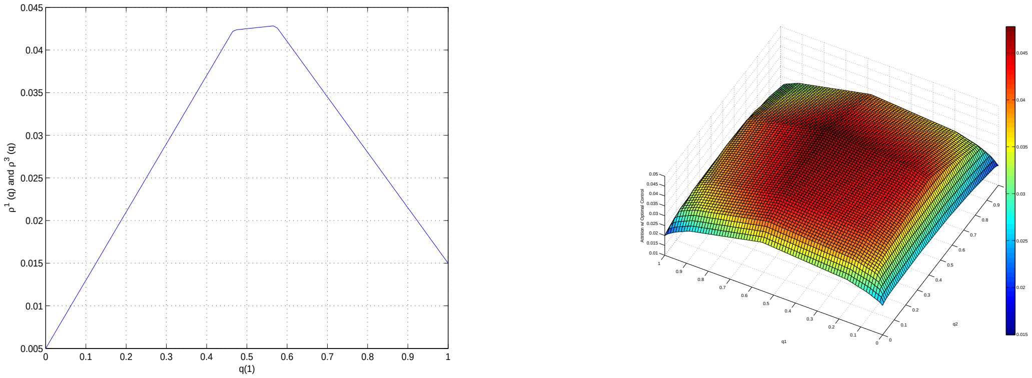

Figure 3: $\rho^1(q) = \rho^3(q)$ (for micro-actions in sub-regions 1 and 3) and open-loop value

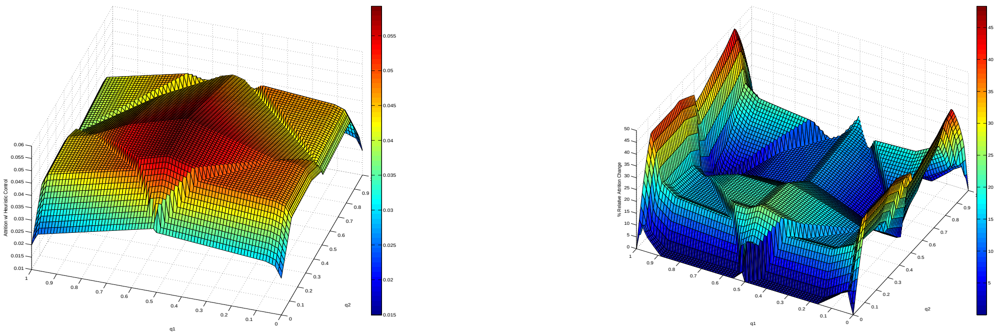

Figure 4: Payoff using heuristic task-plan and difference between optimal and heuristic

References

[1] F.L. Baccelli, G. Cohen, G.J. Olsder and J.-P. Quadrat, *Synchronization and Linearity,* John Wiley, New York, 1992.

[2] R.A. Cuninghame-Green, *Minimax Algebra*, Lecture Notes in Economics

and Mathematical Systems 166, Springer, New York, 1979.

[3] V.N. Kolokoltsov and V.P. Maslov, *Idempotent Analysis and Its Applications,* Kluwer, 1997.

[4] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni and R.E. Davis, "Collective motion, sensor networks, and ocean sampling", Proc. of IEEE, Vol. 95 (2007), 48–74.

[5] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking", Automatica, Vol. 42 (2006), 661–668.

[6] W.M. McEneaney and R. Singh, "Robustness Against Deception", *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind,* A. Kott and W.M. McEneaney (Eds.), Chapman and Hall/CRC Press (2007), 167–208.

[7] W.M. McEneaney, "Some Classes of Imperfect Information Finite State Space Stochastic Games with Finite-Dimensional Solutions", Applied Math. and Optim., Vol. 50 (2004), 87–118.

[8] W.M. McEneaney, B.G. Fitzpatrick and I.G. Lauko, "Stochastic Game Approach to Air Operations," IEEE Trans. Aerospace and Electronic Systems, Vol. 40 (2004), 1191–1216.

[9] W.M. McEneaney, *Max-Plus Methods for Nonlinear Control and Estimation*, Birkhauser, Boston, 2006.

[10] B.M. Miller and W.J. Runggaldier, "Optimization of observations: A stochastic control approach", SIAM J. Control and Optim., Vol. 35 (1997), 1030–1052,