Optimizing Assignment of Air-to-Ground Assets and BDA Sensors

LtCol Kirk A. Yost, USAF

JCS J-8/Forces Division
yostka@js.pentagon.mil

Dr. Alan R. Washburn

Operations Research Department
Naval Postgraduate School
awashburn@nps.navy.mil

ABSTRACT

Finding the correct balance between investment in weapons and information assets has proven to be a substantial challenge. Information alone cannot defeat the enemy on the battlefield, but modern long-range weaponry is useless without the information necessary to employ it. We present a new optimization methodology that combines attack assets and bomb-damage assessment sensors in a single allocation model. The resulting model allows analysis of the tradeoffs between information (sensor) and transformation (attack) assets. We discuss the computational difficulties of the problem and cover the motivating ideas behind the methodology. Then, we outline the mathematical techniques used, and present results for a notional data set.

THE INFORMATION CHALLENGE

General Howell Estes, shortly before he retired as the commander of USSPACECOM, made the following comments in an interview:

> ... combat systems, without timely relevant information, are useless. On the other hand, you can't take out an enemy tank with just information. We need to strike a balance between "shooters" and "information systems" if we're going to be successful in the future. (Scott 1998)

This quote succinctly sums up one of the biggest challenges in structuring modern combat forces. New long-range, precision weaponry requires accurate information to be of any use; furthermore, these new weapons tend to be scarce and expensive. The expense of these weapons is justified largely based on their effectiveness. The newest systems offer accuracy, lethality, and long range, increasing the risk to the enemy while decreasing risk to ourselves. Yet, most of our analyses of these weapons assume the availability of accurate information, which is a key prerequisite.

Information systems, on the other hand, have long been plagued with unsatisfactory metrics. We tend to use measures such as bandwidth and throughput when talking about information systems, but we have difficulty relating changes in these measures to changes in combat outcomes. Information theory, as formulated by Shannon (1949) offers a set of measures, but Shannon's work has nothing to say about the *accuracy* of information. Nonetheless, some valuable work has been done in this area; Sherrill and Barr (1995) offer one such attempt.

This brings us to the present situation, as summarized elegantly by General Estes. We must have both the weaponry and the information to be successful; one is ineffective without the other. Unfortunately, we have few methods available to assess the relationships between the two, much less divide scarce investment resources among the two classes of systems.

ERRONEOUS INFORMATION IN AIR-TO-GROUND WARFARE

We limit our discussion to conventional weapons delivered from fixed-wing aircraft, and sensors that perform bomb-damage assessment (BDA). While these are just subsets of the classes of weapons and sensors, the philosophy in this article has wide applicability.

In air-to-ground operations, the quality of information applies in two basic areas: target acquisition and target assessment. In target acquisition, we include not only the task of finding the target, but also identifying the best set of targets to attack to accomplish a campaign objective. In target assessment, we are concerned with determining the "state" of the target, that is, whether it is functional, destroyed, damaged, or in repair.

We cannot count on error-free information in either area. Consequently, we must use *probability*. We cannot say with certainty that we have located the correct targets, nor can we say with certainty what the affect of our attacks were. In most cases, our information is subject to error.

Air planners have two general approaches to erroneous BDA, neither of which is particularly appealing. The first is informally known as the "pummel factor," and refers to the idea of simply overwhelming uncertainty with firepower. In many operations, we plan to attack critical targets with an extraordinary number of weapons over extended periods, not because the targets are difficult, but because we are not confident that we can get good information quickly on the effects of the attacks.

APPLICATION AREA:
Operational Contribution of C4ISR; Power Projection, Planning, and Execution

OR METHODOLOGIES:
Linear programming; Markov processes; Stochastic processes; Simulation

The second approach is to "play weatherman." Here, the planner simply reports a probability to the decision maker, who presumably can allocate resources based on this information. However, this method has some risk. In his book *Crusade* (1993, p. 234), Rick Atkinson documents General Norman Schwartzkopf's response (expletives deleted) after being briefed that a particular Iraqi unit was "64% effective:"

> Not 65%? Not 63%? ****, you don't know what the **** you're talking about, do you?

Now, it is not clear whether General Schwartzkopf was disagreeing with the implied accuracy of the estimate or the way it was presented. Regardless, it *is* clear that merely reporting probabilities of target status to a commander faced with hundreds or thousands of allocation decisions is not particularly helpful. We recommend the article by Lewis (1994) for additional discussion of this issue.

Consider the situation shown in Figure 1, which shows two aircraft shelters attacked in DESERT STORM. The outward appearances of the shelters are completely contrary to the effects achieved. The functional shelter appears to be destroyed, while the opposite is true for the other shelter. Yet, a campaign planner would have to decide, on the basis of inconclusive imagery, whether or not to reattack these targets. In this situation, the information available makes the probability of making an error high.

CURRENT MODELING APPROACHES

What does the analytical community have to offer to air planners facing such decisions? The answer as implemented in many current campaign models is: not much.

Many models employ parameters that go by the name of "C4ISR degrades," or "perception factors," that attempt to capture information effects, but these parameters are largely not measurable. These parameters have come to be known as "interesting rheostat knobs" (IRKs). These models can only be used to determine the criticality of the IRKs, which cannot be tied to information resources. As an example, the 1995–97 Deep Attack Weapons Mix Study (DAWMS) employed 5 different IRKs to represent C4ISR effects in a model. One, the "dead or alive" factor, represented reductions in expected kills per sortie due to both "underestimate[s] and overestimate[s] of target damage as well as inefficiencies introduced by untimely reporting of target status" (IDA 1996). This fac-

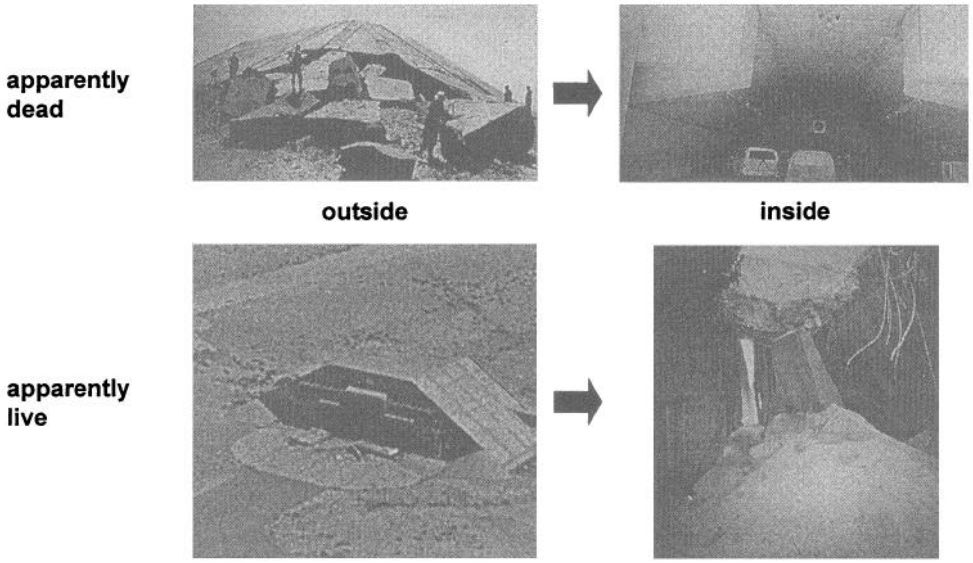

Figure 1. Two aircraft shelters attacked in DESERT STORM. The top shelter appears to be extensively damaged, but the inside is untouched. The bottom shelter has what appears to be an inconsequential hole in the roof, but the contents are completely destroyed (Cohen 1993).

tor varied between 0 and 1, but was not defined as anything measurable (or even derivable) by any other means than expert judgment. Not surprisingly, this IRK turned out to be crucial in the final results produced by DAWMS.

Other examples are the infamous SFINT (scaling factor—interdiction) and SFCAS (scaling factor—close air support) parameters in the TACWAR campaign model (TRAC 1994, pp. D-25–D-26). These factors are commonly used to reduce sortie effectiveness due to C4ISR problems, but they are not tied to any measurable resource or activity. Interestingly enough, the version of TACWAR currently running in the Joint Staff sets SFINT and SFCAS using the IRKs developed for DAWMS.

REQUIREMENTS FOR AN INTEGRATED MODEL

We do not intend to denigrate the modeling approaches described above. Models employing IRKs (even nested IRKs, in the TACWAR example above) are honest and useful attempts to evaluate C4ISR effects. Experienced modelers also know that IRKs provide very quick sensitivity analysis. If the IRK does not effect the results at its extremes, then it is unlikely that installing a large functional model to replace the IRK would be worthwhile.

But what happens when we attempt to replace sensor IRKs with measurable parameters? At the system level, we can estimate the probabilities that a sensor responds, the distribution of the timeliness of the response, and the probability of a correct assessment. Unfortunately, this still leaves us with an incomplete decision tool. As General Schwartzkopf's quote indicates, we cannot easily translate a large number of probabilities into attack decisions. Since we cannot currently do this translation, we settle for modeling approaches that reduce C4ISR to noise factors that degrade results.

Furthermore, there is very little available work on modeling the allocation of sensors. A notable exception is the Sensor-Platform Allocation Model (Rice 1997), which uses a linear program (LP) to match sensors to targets to provide coverages. While the coverage requirements are inputs into this model, it at least attempts to find a good allocation. Many models use fixed rules for assigning sensor resources and leave the user to discover a good set of assignments.

Consequently, we would like an integrated model that, based on the *probability* that a target is in a particular state, decides whether to attack (and with what), look (and with what), or to ignore the target. This avoids the problem noted by General Schwartzkopf, because the model recommends actions based on probabilities, rather than merely computing the probabilities. Furthermore, such a model could help investigate the relationships between transformation (attack) and sensor (information) resources.

Such a model requires performance data for the sensors. We characterize our BDA sensor systems using two different parameters. The first is throughput, or the number of targets the sensor can assess per unit time. This definition is important because it specifies the number of assessments available for campaign planning. We do not use bit rates, bandwidth, or even images produced; all we care about is the finished assessments. In this sense, we are describing the output of the entire sensor system (often called the promulgation, exploitation, and dissemination system, or PEDS). The second is the assessment probabilities of the sensor system. We assume that the sensor system assesses a target as being in a discrete state (e.g., functioning or destroyed), and that probabilities of correct assessments given the target state are available.

COMPUTATIONAL CHALLENGES

The introduction of target state probabilities also introduces tremendous computational difficulties. To illustrate these, consider the following small scenario: we have 1 target, with two possible states (functioning or destroyed). We use a 3-day planning horizon with 3 operating periods per day, so we have 9 possible "stages." In each stage, we can either attack the target with 1 of 2 weapon types, look at it with 1 of 2 sensor types, or pause until the next stage. The weapons have known probabilities of kill, and the sensors have known error probabilities (reporting a functioning target as destroyed, or vice versa).

We call a rule for taking an action based on the probability the target is destroyed and the stage a *policy*. One measure of the difficulty of solving this problem is the number of available policies. Even for this simple problem, the answer is surprisingly large: there are 8.3×10^{270} possible policies!

To see why this is, let N be an index for the stages and S_N be the number of possible policies. The recursive expression for the number of policies for this example is

$$S_1 = 5$$
$$S_N = 3S_{N-1} + 2S_{N-1}^2, N = 2, \ldots 9.$$

The sensor looks result in two possible outcomes, so any particular policy must specify an action for either possible target assessment, resulting in the squared term in the recursion.

Our current doctrine only considers a limited number of policies, such as "shoot-shoot," or "shoot-look-shoot." Yet, with many different types of aircraft, weapons, and sensors available, there are many more plausible policies. Consider the policy in Figure 2, which reflects many realities in modern air warfare. The policy begins with an attack using a plentiful asset (the F-16 with an unguided weapon). If UAV 1 (a plentiful but somewhat inaccurate sensor) assesses the target as destroyed, the policy stops. If it assesses it as alive, we look again. If the target still appears alive, we send in the more precise, but less plentiful, UAV 2. If it confirms the target is still alive, we assign a scarce, high-value F-117 and a laser-guided GBU-27 bomb to the target.

Given the attack probabilities of kill and the assessment probabilities of each sensor, we can compute the expected sensor looks and sorties consumed and the probability the target is destroyed by this policy. But now consider a real campaign problem, with hundreds of aircraft-weapon combinations, thousands of targets, and 10–20 sensor types. We cannot even begin to compute all the possible policies.

THE MASTER ALLOCATION PROBLEM

Assume for the moment that we can generate the entire collection of possible attack policies. Then, we could solve the allocation problem using a relatively simple model.

The DoD possesses many optimization models that allocate aircraft and weapons to targets, such as HEAVY ATTACK (Brown, Coulter, and Washburn 1994), the Conventional Target Evaluation Model (Cotsworth 1993), and the Conventional Forces Assessment Model (Yost 1996). These models all use "shoot" or "shoot-shoot" policies as input, but they could be easily extended to handle sensors. Consider the following formulation of such an LP:

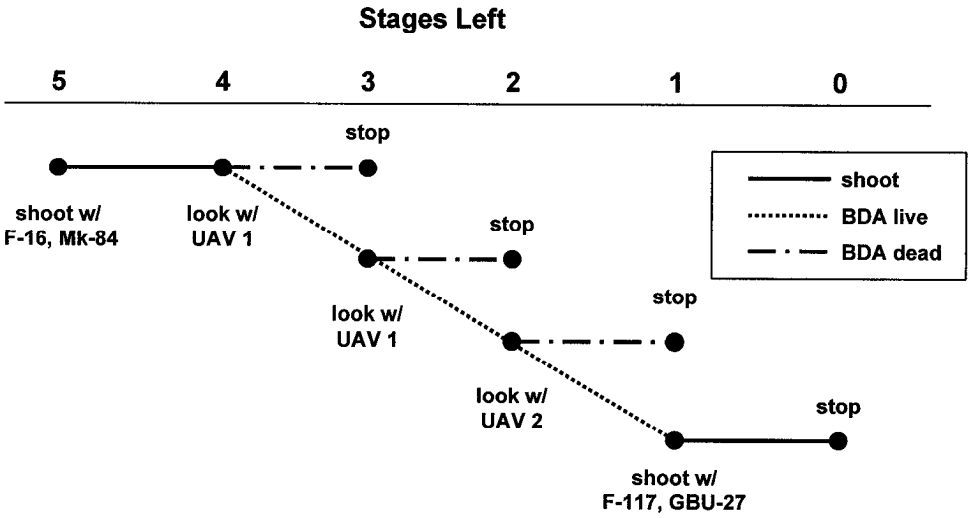

Figure 2. A complex policy. The planner may start by attacking the target with a plentiful asset (the F-16 using unguided Mk-84 bombs). If the BDA from UAV 1 indicates the target is destroyed, we stop; otherwise, we look again. If the target is still alive, we use the more precise, but less plentiful, UAV 2. If it still indicates the target is alive, we resort to using a precision attack with an F-117 and a laser-guided GBU-27 bomb.

- Sets And Set Indices

$i \in I$	aircraft types
$w \in W$	weapon types
$g \in G$	target types
$o \in O$	sensor types
$s \in S_g$	possible policies for target type g
t	time period; $t = 1, 2, \ldots T$

- Data

V_g	value of gth target type per target
PD_{gs}	probability of destroying target g using policy s
ES_{igst}	expected number of aircraft i sorties required by policy s against target g in period t
$SORT_{it}$	number of sorties of ith aircraft type available in period t
EW_{wgs}	expected number of type w weapons expended using policy s against target g
WPN_w	number of weapons of type w available
TGT_g	number of targets of type g available
EL_{ogst}	expected number of sensor o looks at target g required by policy s in period t
$LOOK_{ot}$	number of type o sensor looks available in period t
EA_{is}	expected attrition of aircraft i using policy s
$MAXATT_I$	maximum expected attrition allowed for aircraft of type i

- Variables

x_{gs}	number of type g targets attacked using policy s

- Objective Function

$$\max_x \sum_{g \in G} \sum_{s \in S_g} V_g PD_{gs} x_{gs} \quad (1)$$

- Constraints

$$\sum_{g \in G} \sum_{s \in S_g} ES_{igst} x_{gs} \leq SORT_{it}, \text{ for all } i, t(sd_{it}) \quad (2)$$

$$\sum_{g \in G} \sum_{s \in S_g} EW_{wgs} x_{gs} \leq WPN_w, \text{ for all } w(wd_w) \quad (3)$$

$$\sum_{s \in S_g} x_{gs} = TGT_g, \text{ for all } g(td_g) \quad (4)$$

$$\sum_{g \in G} \sum_{s \in S_g} EL_{ogst} x_{gs} \leq LOOK_{ot}, \text{ for all } o, t(ld_{ot}) \quad (5)$$

$$\sum_{g \in G} \sum_{s \in S_g} EA_{is} x_{gs} \leq MAXATT_i, \text{ for all } i(ad_i) \quad (6)$$

$$x_{gs} \geq 0, \text{ for all } g \in G, s \in S_g \quad (7)$$

The objective function (1) maximizes the expected value of destroyed targets. The sortie constraint (2) limits expected sortie consumption for each aircraft type and time period. The weapon constraint (3) limits expected weapon consumption across the time horizon. The policy allocation constraint (4) limits the assignment of policies to available targets; we include a "null" policy that assigns no resources to a target for feasibility. The sensor constraint (5) constrains expected sensor looks by type and time period, and the attrition constraint (6) limits expected losses by aircraft type.

The dual variables associated with each set of constraints are listed in parentheses. These variables, which give marginal resource costs, are important in this article and are defined below:

sd_{it}	marginal cost of a sortie of type i in period t
wd_w	marginal cost of a weapon of type w
td_g	marginal cost of a target of type g
ld_{ot}	marginal cost of a look from sensor type o in period t
ad_i	marginal cost of attriting an aircraft of type i

By assuming the aircraft, weapons, sensors, and targets of each type are indistinguishable (a common assumption for this class of models), we end up with an optimization model with a relatively small number of constraints. Note also that the details of the policies play no role in the LP; all that is required is the probability of target destruction and the expected resources used. The policy shown in Figure 2 could easily be represented in this LP.

Basic LP theory tells us that solving the LP with any subset of the possible policies yields a lower bound on an optimal solution. These policies can come from current doctrine; it is straightforward to generate the expected results of "shoot-shoot" or "shoot-look-shoot" tactics for various combinations of weapons and sensors. As a result, we can quickly produce a lower bound on the optimal solution.

FINDING IMPROVING POLICIES

We must now confront the main issue, which are the sizes of the sets S_g. The huge number of possible policies prohibits generating all policies in this set for inclusion into an optimization. Even our trivial example contained 8.3×10^{270} possibilities; an instance of the master LP with a realistic number of platforms, weapons, sensors, and target types would be unmanageable.

As noted above, we can easily construct a starting set of policies to run the LP and produce a lower bound. The challenge is to search the remaining policies to find those that can potentially improve the solution. But, we do not have to find very many of these policies. Basic LP theory also tells us that there exists an optimal solution to the LP that contains at most as many nonzero variables as there are constraints (e.g., Bazarra, Jarvis, and Sherali 1990, pp. 53–54). The LP above only contains constraints for each type of resource, so the row size of the problem is small (several hundred for the problem we solve later).

We clearly need a separate search procedure to find policies that can improve the LP solutions. The sample policy shown Figure 2 is a decision tree; at each stage and state, we have to make a decision on which resource to use. We know the performance parameters of the attack and sensor resources, so we can compute the expected consumptions and results of any policy by traversing the tree. What we lack are costs we can use to choose among resources at each decision node.

The master LP, however, explicitly defines the value of each target. If we could find values on the same scale for the aircraft, weapon, and sensor resources, we should be able to solve for policies that can improve the LP solution by solving the corresponding decision trees. With this in mind, we define the following additional notation for the "policy subproblem," which searches for improving policies:

$a \in ATK_g$	set of allowable attack tactics for target type g
p	pause action
n	stage, or number of time periods remaining; $n = 0, 1, \ldots, T$
NW_{aw}	number of type w weapons required by tactic a
NS_{ai}	number of type i aircraft sorties required by tactic a
PA_{aig}	probability of attrition of aircraft type i using tactic a against target type g
PK_{ag}	probability of destroying a target of type g using tactic a
δ_o	probability sensor type o reports the target is functioning, given that the target is destroyed
β_o	probability sensor type o reports the target is destroyed, given that the target is functioning
π	probability the target is destroyed

In the following, we assume the attacks and sensor looks are independent. Furthermore, we assume (without loss of generality) that the attacking aircraft has no capability of assessing a target, and that the sensor errors are independent of target type. In each stage, we can either attack the target, pause, or look at the target with a sensor, but the only mechanism that can change a target's true state is an attack. All targets are assumed to be functioning at the start of the time horizon.

At any stage, we can determine the probability a target is destroyed based on the action we take and the existing probability of destruction. Denote the *current* probability the target is destroyed as π. If we pause, then the updated probability the target is destroyed (denoted π') is unchanged. If we attack the target, then the updated probability is

$$\pi' = \pi + PK_{ag}(1 - \pi).$$

If we look at the target, we have to consider each of the possible assessment responses. Using Bayes' Theorem, π', given π and an assessment the target is still functioning, is

$$\pi' = \frac{\delta_o \pi}{(1 - \beta_o)(1 - \pi) + \delta_o \pi}.$$

Conversely, π', given π and an assessment the target is destroyed, is

$$\pi' = \frac{(1 - \delta_o)\pi}{\beta_o(1 - \pi) + (1 - \delta_o)\pi}.$$

The denominators of the two expressions give the probability of assessing the target as functioning and destroyed, respectively.

While this problem can be solved as a decision tree, it is more efficient to use stochastic dynamic programming (SDP) to find a solution.

SDP requires fewer operations than enumerating a decision tree, and the SDP can be written compactly. Using the probabilities above, we write the formulation of this SDP as

$$val_0^g(\pi) = (\text{value of target } g)\pi,$$

$$val_n^g(\pi) = \max \text{ of}$$

$$\left[\begin{array}{l}
(\text{pause})\ val_{n-1}^g(\pi), \\
(\text{attack}) \max\limits_{a \in ATK_g} \{-(\text{cost of attack}_{agn}) \\
\quad + val_{n-1}^g(\pi + PK_{ag}[1-\pi]), \\
(\text{look}) \max\limits_{o} \left\{ \begin{array}{l} -(\text{cost of look}_{ogn}) \\ \quad + [(1-\beta_o)(1-\pi) + \delta_o\pi] val_{n-1}^g \\ \left(\dfrac{\delta_o\pi}{(1-\beta_o)(1-\pi)+\delta_o\pi}\right) \\ \quad + [(1-\beta_o)(1-\pi) + \delta_o\pi] val_{n-1}^g \\ \left(\dfrac{(1-\delta_o)\pi}{\beta_o(1-\pi)+(1-\delta_o)\pi}\right) \end{array} \right\}
\end{array}\right]$$

$$n = 1, 2, \ldots T.$$

Solving this SDP will yield a policy that gives the maximum expected "payoff" for prosecuting a target of type g. The SDP solution will define, for each time period and value of π, the optimal action to take, similar to the policy shown in Figure 2. The SDP balances the value of the target with costs of the resources and their expected amount of use in the policy; if the target's value is low and the resource costs are high, the optimal policy is to ignore the target. Conversely, if the target's value is high and the resource values are low, the optimal policy can expend considerable resources and still achieve positive expected value.

We still need costs for attacks and looks. Fortunately, the master LP provides these resource values via the marginal costs given by the dual variables. We can use these costs as proxies for the values of the aircraft sorties and attrition, weapon expenditures, and sensor looks. For example, the total expected sortie cost for the policy in Figure 2 would be the dual cost of an F-16 sortie with 5 stages left ($sd_{F\text{-}16,\,9-5+1}$), plus the probability that the F-117 sortie is used in the policy times the dual cost of the sortie ($sd_{F\text{-}117,\,9}$).

Referring back to the formulation of the LP and the additional notation above, we can compute the attack and look costs using the marginal costs:

$$\begin{aligned}
&\text{cost of attack}_{agn} \equiv \text{sortie cost} \\
&\quad + \text{weapons cost} + \text{attrition cost} \\
&\quad = NS_{ai} sd_{i,T-n+1} + NW_{aw} wd_w \\
&\quad + NS_{ai} PA_{aig} ad_i \\
&\qquad n = 1, 2, \ldots T,\ a \in ATK_g; \\
&\text{cost of look}_{ogn} \equiv \text{sensor look cost} \\
&\quad = ld_{o,T-n+1},\ n \\
&\quad = 1, 2, \ldots T.
\end{aligned}$$

The attack cost is the cost of the sorties expended, weapons expended, and expected attrition, as "priced" by the master LP. Similarly, the cost of a sensor look is the LP-determined dual cost from the sensor constraints for each stage. Once we have solved for the policy, we can enumerate the resulting decision tree and compute the expected consumptions, so determining the expected use of each resource is straightforward.

THE COMPLETE DECOMPOSITION ALGORITHM AND BOUNDS

The master LP and the SDP above can be combined to form a complete decomposition algorithm as shown in Figure 3. Starting with an initial set of attack policies, the master LP assigns costs to the resources via the dual variables. The subproblems then find improving policies for each target type and introduce them to the master LP. The new LP solution then adjusts the marginal costs.

It is necessary to develop a stopping criterion for this scheme, as the huge number of possible policies can result in very long tail convergence. The criterion suggested in Figure 3 is to stop when the difference between an upper and lower bound is small.

Solving the master LP with any subset of the policies yields a lower bound on the solution. To get an upper bound, we can use Lagrangian relaxation (e.g., Fisher 1985). In Lagrangian relaxation, some of the constraints of the problem are rewritten as penalty terms in the objective function (hence "relaxing" the

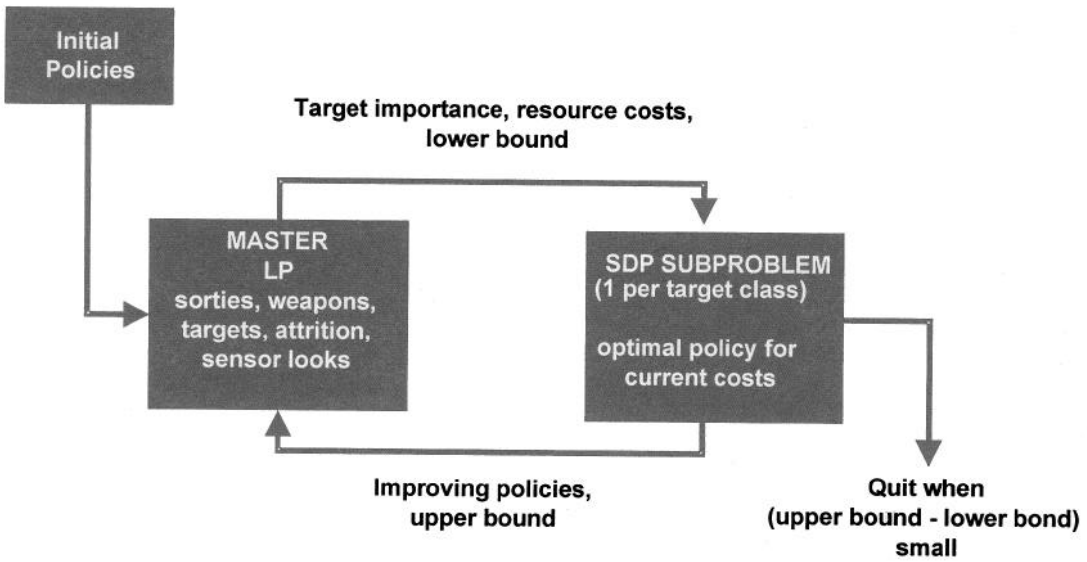

Figure 3. Basic decomposition scheme for the sensor-shooter model. Starting with a small set of initial policies, the LP produces implicit costs on sensor and attack resources via the dual variables. The SDP subproblem then produces the best policy for each type of target for those costs. The algorithm produces upper and lower bounds and stops when they become sufficiently close.

problem). The relaxed version of the master LP is

$$\max_x \left\{ \begin{array}{l} \sum_{g\in G}\sum_{s\in S_g} V_g PD_{gs} x_{gs} + \sum_{i\in I}\sum_{t} sd_{it} \\ \left(SORT_{it} - \sum_{g\in G}\sum_{s\in S_g} ES_{igst} x_{gs} \right) \\ + \sum_{w\in W} wd_w \left(WPN_w - \sum_{g\in G}\sum_{s\in S_g} EW_{wgs} x_{gs} \right) \\ + \sum_{o\in O}\sum_{t} ld_{ot} \left(LOOK_{ot} - \sum_{g\in G}\sum_{s\in S_g} EL_{ogst} x_{gs} \right) \\ + \sum_{i\in I} ad_i \left(MAXATT_i - \sum_{g\in G}\sum_{s\in S_g} EA_{is} x_{gs} \right) \end{array} \right\}$$

$$st \sum_{s\in S_g} x_{gs} = TGT_g \text{ for all } g \in G$$

$$x_{gs} \geq 0 \text{ for all } g \in G, s \in S_g.$$

To see that this LP provides an upper bound, consider a different version of the original master LP, but with the objective function shown above. Since the new objective function differs only in that it contains additional nonnegative terms, its optimal value must be at least as large as the original LP. Relaxing all but the target constraints yields the upper bound LP, but relaxing constraints in an optimization cannot decrease its optimal value. Therefore, the LP above provides an upper bound for any intermediate set of dual variable values.

Furthermore, this LP decomposes into |G| separate optimizations, one for each target type. The solution of each target type optimization is

$$u_g \equiv val_1^g[1],$$

where the argument 1 indicates that the target is assumed functional at the start of the optimization. Since u_g contains expected target payoff as well as the costs of the expected resources consumed, the upper bound can be rewritten as

upper bound

$$\equiv \left\{ \begin{array}{l} \sum_{i\in I}\sum_{t} sd_{it} SORT_{it} + \sum_{w\in W} wd_w WPN_w \\ + \sum_{o\in O}\sum_{t} ld_{ot} LOOK_{ot} + \sum_{i\in I} ad_i MAXATT_i \\ + \sum_{g\in G} TGT_g u_g \end{array} \right\},$$

This bound is the current marginal costs multiplied by the right-hand-sides of the constraints, plus the number of targets of each type times the value of the current optimal policy for

each target type. The global upper bound is the minimum of all upper bounds generated; each is a bound on the optimal solution, and there is no guarantee that the sequence of upper bounds will decrease. Nonetheless, each relaxation is an upper bound, so their minimum is as well.

This decomposition is guaranteed to converge. Each possible policy is a column in the LP, and the SDP subproblems explicitly search all possible policies for each target type. If the subproblems cannot produce at least one policy that prices favorably in the master LP, then there can be no further improvement, and we have found the optimal solution. The decomposition is really no different than any other linear program; we merely choose to search the available columns (policies) in a separate step to avoid generating an unmanageable (and unnecessary) number of them.

THE SDP SUBPROBLEM AS A POMDP

We still have more challenges to overcome with the subproblems. The state of the target, π, is continuous on the interval [0,1]. Therefore, it is not possible to iterate across all possible states, and standard stochastic dynamic programming iteration will not work. Systems such as this are known in the literature as Partially Observable Markov Decision Processes (POMDPs); Sondik (1971) formalized the theory of these systems.

The POMDP is a generalization of the more common Markov Decision Process (MDP) model. An MDP consists of an object with a finite set of states, a set of available actions with probabilistic outcomes, a specified time horizon, and a set of transition probabilities. The object's state is based solely on its previous state and the action taken. More importantly, the decisionmaker in the MDP model has perfect information on the state of the object. In our problem, this would make sensors unnecessary; we would know immediately the outcome of any attack.

The POMDP model extends the MDP by introducing observation errors, which result in "partial observability." A POMDP contains an additional set of observation probabilities, which are the probabilities of observing each state when the object is in a particular state. In our example, these probabilities correspond to the sensor assessment probabilities.

The dynamics of a POMDP are shown in Figure 4. The object we are controlling begins in a known state (or has a known distribution over its states). We take an action, which results in a probabilistic state change. The object then transitions, and we make an observation that is subject to error. We incur some cost based on the resources used in the action and observation, and receive a terminal reward based on the final state of the object.

In our application, solving the POMDP means finding the policy that maximizes the expected total reward (i.e., the expected terminal reward—the expected action and observation costs). The introduction of partial observability, however, makes the POMDP much more difficult to solve than the corresponding MDP. Since we now must make decisions based on the probability of a target being in a certain state, we have a continuous state space rather than the simple (functioning, destroyed) state space in the MDP. This means that the normal MDP solution approach (SDP) cannot be applied directly to the POMDP.

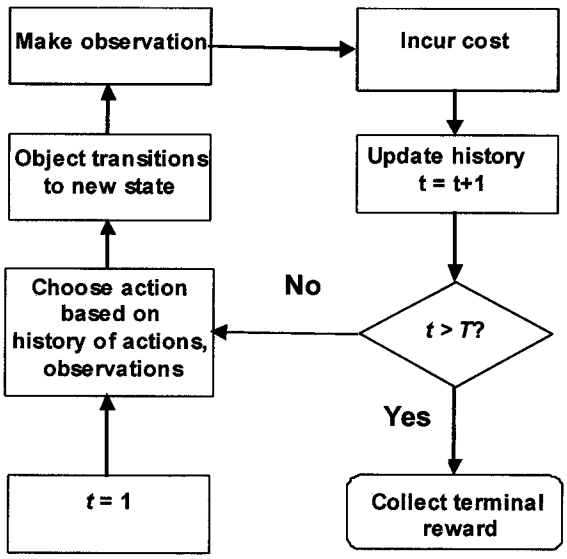

Figure 4. The dynamics of a T-stage POMDP. The object we are controlling begins in a known state (or has a known distribution over its states). We take an action, the object then transitions, and we make an observation that is subject to error. We incur some cost based on the resources used in the action and observation, and receive a terminal reward based on the final state of the object.

A description of exact solution techniques for POMDPs is beyond the scope of this article. In general, it is very difficult to get an exact solution for POMDPs with large numbers of states or long time horizons, and this issue is the most likely cause for the lack of POMDP applications. A complete technical description of the exact solution scheme for the sensor-shooter problem is given in Yost (1998); we also recommend POMDP survey papers by Lovejoy (1991a) and Cassandra (1994).

It is important to note, however, that the subproblem can easily be solved by SDP by quantizing the target states (e.g., 0, 0.1, 0.2, ... 1.0). This, coupled with an interpolation scheme for determining the value function, allows the use of standard dynamic programming iteration. This is a very attractive implementation option, particularly if the precision of input performance data (such as sensor error probabilities) does not justify the computational demands of an exact method. Also, the finer the quantization, the closer the policies will be to an exact optimum.

A drawback with quantization schemes is that the upper bound generated by the relaxed LP is no longer a global upper bound, because the quantized subproblems only search a subset of all possible policies. Lovejoy (1991b) and Hauskrecht (1998) offer methods that can compute legitimate upper bounds, but a much simpler mechanism is to stop the decomposition when the change in the optimal value of successive lower bound LPs is below some tolerance.

The reader may wonder if it is possible to solve the entire problem as a single POMDP. Unfortunately, this is even more intractable than trying to generate all possible policies for an LP. Trying to solve a POMDP for, say, m targets each with n states requires a state space of size n^m, with additional states required to keep track of weapons and sensors. Thus, the POMDP model alone cannot solve this problem.

AN INSTANCE OF THE SENSOR-SHOOTER PROBLEM

With the LP providing costs of resources and the POMDP producing improving policies based on those costs, the decomposition can solve the sensor-shooter problem in an integrated fashion, and can do so quickly. Our notational example is constructed from real USAF planning data for a theater campaign. The allocation problem contains 9 strike aircraft types, 42 weapon types, 65 target types, and 10 BDA sensor types. The planning horizon is 72 hours, and contains 3 attack waves in each 24-hour period, for a total of 9 stages. For every combination of aircraft, weapon, and target type, we have several feasible attack tactics; there are a total of 5203 feasible aircraft-weapon-target-tactic combinations.

This instance has an essentially infinite number of policies (approximately 10^{762}). Yet, the structure of the master LP ensures we will only pick a small number of these policies. There are only 287 constraints ($9 \times 9 = 81$ sortie availability constraints, 42 weapons availability constraints, 65 target availability constraints, $10 \times 9 = 90$ sensor availability constraints, and 9 aircraft attrition constraints). Consequently, an optimal solution exists that uses at most 287 policies; the difficulty is finding this small set among the huge number of possible policies.

We initialize the decomposition by computing a set of initial policies heuristically. In this example, we determine the 3 best "shoot once" policies for each target type for each time period, and then add the 3 best "shoot-look-shoot" policies for each target type and each applicable time interval. Finally, we include the best attack tactic for each aircraft and weapon type to ensure we generate dual costs for all resources in the optimization. This results in an initial set of 2214 policies, which are columns in the master LP.

Once we run the LP with the initial policies, we generate a set of marginal costs for sorties, sensor looks, weapons, target attacks, and aircraft attrition. With these costs and the value of each target, we solve the POMDP for each target type, yielding 65 new policies (one for each target type). We add these policies to the master LP, rerun it, and then use the new marginal costs to resolve the POMDPs for new policies. We terminate the decomposition when the difference between the upper and lower bounds is less than 0.1%.

In our implementation, we do not discard any policies, including the initial ones; we keep all generated columns in the master LP. Many decomposition schemes in the literature investigate schemes to control column growth, but we have not found this to be an issue in our test problems.

COMPUTATIONAL RESULTS

Despite the huge number of possible policies, this approach is very efficient. Figure 5 below shows the results of running the problem above; the decomposition converges in 98 iterations and only requires about 120 seconds of total time on a 300 MhZ Pentium-II PC. Approximately 60% of the time was spent solving the POMDP subproblems; the rest of the time was used to solve the master LPs. The POMDPs were solved using Visual Basic 5.0 (Microsoft 1997); the master LPs were solved using the CPLEX callable library (ILOG 1997).

In this example, we used an exact method known as the "linear support algorithm" (Cheng 1988) to solve the POMDP subproblems. We note, however, that quantizing the target states would produce results very close to those shown in Figure 5. During our research, we experimented with several schemes that gave near-exact results with much faster subproblem solution times. While using a quantization scheme complicates the problem of producing a valid upper bound, the algorithm could just as easily terminate when improvement between successive iterations becomes small.

Of more interest are the results. The objective function value is nearly triple that achieved with the initial policies, which reflect current doctrine. While this answer is derived from notional data, it gives strong evidence of our earlier claim that sensor allocations and attack allocations should not be computed separately.

We also implement the decomposition as part of a simulation. As we mentioned above, the typical campaign simulation follows preset rules to determine how C4ISR assets are employed. Instead, we use the decomposition to determine BDA and attack allocations in each stage, simulate the outcomes, and then recompute new policies based on the simulated outcomes.

Recall from the discussion of the subproblem that we started with each target assumed in state 1 (fully functional). This is not a requirement, as the POMDP subproblem can start in any state. In our simulation, we solve the decomposition for the 9-stage problem, but only implement the first-stage decisions. We then

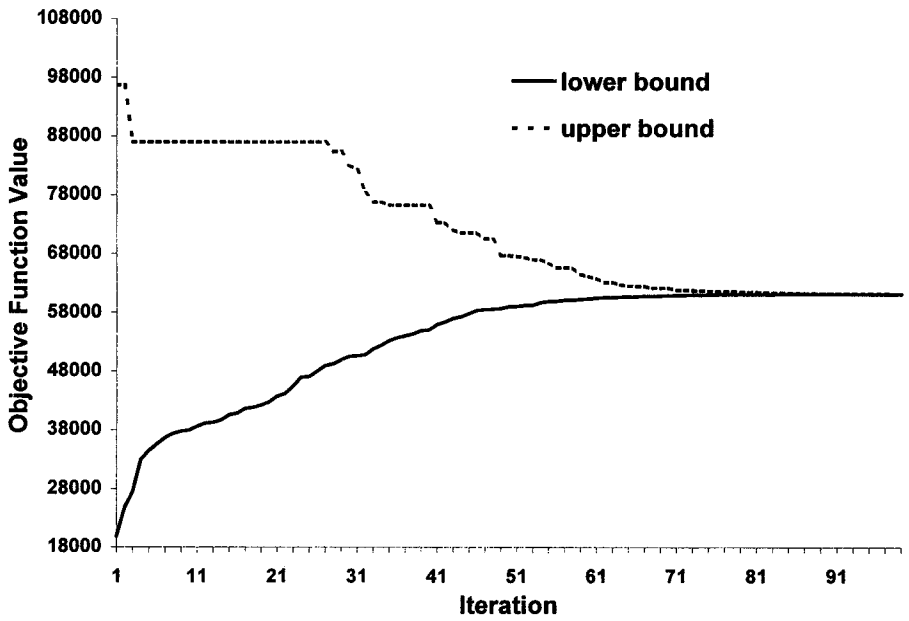

Figure 5. Results of the Decomposition with POMDP Solutions Exact to a Numerical Tolerance. The decomposition is initialized with 2,214 policies generated by a heuristic, and takes 98 iterations to solve to within a 0.1% decomposition gap (difference between the upper and lower bound). The POMDP subproblems generate a total of 4,507 policies, and the final objective function value is nearly triple the objective function value achieved using only the heuristically-generated policies.

simulate the attack and look outcomes, restart the decomposition for the remaining stages, and repeat the process until all stages have been simulated.

Yost (1998) shows that solving the decomposition gives an upper bound on the average objective function value. However, this yields no information on the *distribution* of objective function outcomes or allocations. The simulation described above allows us to estimate these distributions. Figure 6 below shows the distribution of total target value destroyed for 150 repetitions using the sample data.

Figure 6 also shows that the sample mean from the simulation is fairly close to the upper bound on the mean objective function value given by solving the decomposition. Most time-staged, deterministic optimizations do not do so well, because they in essence assume perfect information from stage to stage. However, this decomposition is different because the POMDP subproblems produce policies that optimize based on the attack and sensor probabilities.

To further demonstrate the value of the optimization-simulation approach, consider Figure 7. This is similar to Figure 6, but compares three different cases: having perfect BDA sensors (no error rates); having no BDA sensors; and having realistic BDA sensors.

The results follow our expectations; as the sensors improve, so do our results. The sample distributions, however, also allow us to measure variation and assess risk. The no sensor case, which corresponds to pure attack policies, has a sample mean similar to the other two cases. Nonetheless, the no sensor case has much more variation and does much worse more often than the other two cases.

The optimization-simulation approach mentioned above is particularly attractive because we are dealing with probabilistic target states. In existing weapons optimization models, the number of remaining targets is passed from time period to time period and is computed using expectations. In the technique above, the optimization takes actions based on

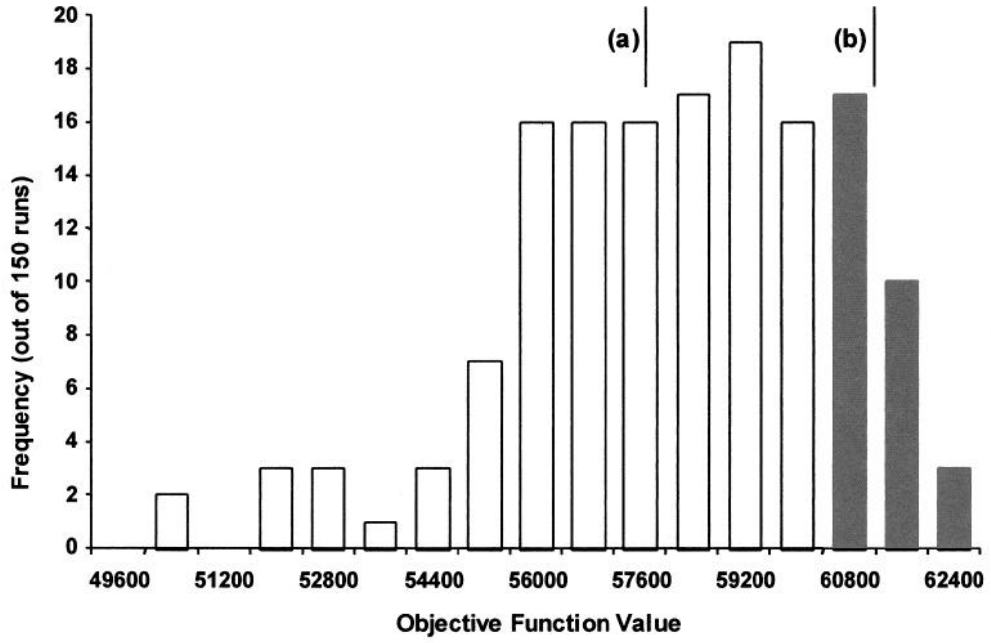

Figure 6. Sample distribution of objective function values from the sensor-shooter simulation. This graph results from running the decomposition, simulating the outcomes for a stage of attacks and sensor looks, and reoptimizing for the next stage. The sample mean is given by (a); the original solution from running the decomposition, which gives an upper bound on the mean, is given by (b). Using simulation allows estimating the distribution of objective function values; also, we see the simulation sample mean is within 6% of the upper bound on the mean. The black bars indicate the sample cases where the objective function exceeded the upper bound on the mean.

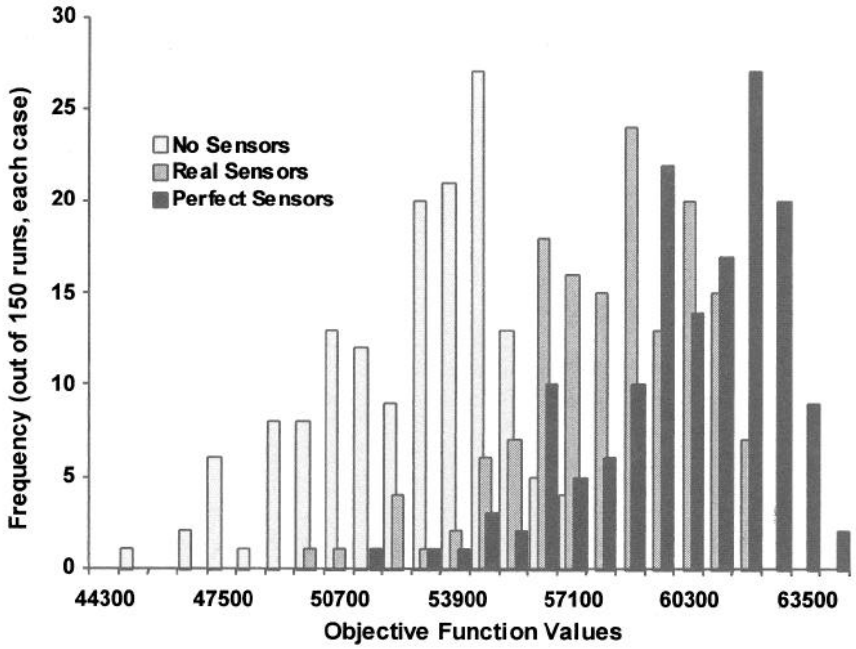

Figure 7. Comparison of simulation outcomes for perfect sensor, realistic sensor, and no sensor cases. This gives the sample distributions of objective function outcomes under the three cases. The case with no sensors, which corresponds to pure attack policies, has the most variation and the lowest mean. The case with realistic sensors performs better, while the perfect sensor case has the highest mean outcome and the least variation.

the estimated *states* of the targets, which are expressed as probabilities.

EXTENSIONS AND RELATED APPLICATIONS

An important point to make about the decomposition is the ease with which it handles very complex nonlinear relationships in the subproblems. The SDP shown in this article was extended in Yost (1998) to include platforms that could simultaneously attack and assess a target (e.g., a strike aircraft with targeting video), sensors with differing response times, and probabilities of sensors providing a response.

Another timely example is modeling of collateral damage. Suppose each attack has "no guide" probability, denoted NG_{ag}, and if the weapon fails to guide, it can cause substantial (and undesired) collateral damage. If the weapon causes collateral damage, then all attacks against the particular target must cease. The following modification to the subproblem implements this rule:

$$\text{(attack)} \quad \max_{a \in ATK_g} \left\{ \begin{array}{l} -(\text{cost of attack}_{agn}) \\ +(1 - NG_{ag}) val^{g}_{n-1} \\ (\pi + PK_{ag}[1 - \pi]) \\ +NG_{ag} V_g \pi \end{array} \right\}$$

$$n = 1, 2, \ldots T.$$

Including this has no effect on the master allocation problem. It remains linear, and only considers the expected payoff and resources consumed.

The methodology presented in this article also directly solves several other models presented in the literature. Evans (1996) discusses BDA and sortie requirements, and this methodology addresses issues he raises about the differences between bad BDA and no BDA. Aviv and Kress (1997) analyze various shooting and looking policies; this methodology is a generalization of the problem they study.

CONCLUSION

We contend that the allocation of modern long-range weapons should not be considered independently of the allocation of BDA sensor resources. Unfortunately, including BDA sensors requires maintaining probabilities of target states and enormously complicates the allocation problem.

Yet, the latter case is the reality of modern warfare. We must frequently make decisions based on incomplete or inconclusive information (such as shown in Figure 1), and many times it is not obvious whether we should attack, gather more information, or skip a target. The decision is a function of the value of the target and the value of the available resources, and, for a single target, we can use military judgement to come to a decision. It is when we have to make hundreds or thousands of such decisions that the problem becomes overwhelming.

The problem is too large to be solved as a monolithic linear program or as a single decision process. Nonetheless, by combining the LP and the POMDP, we can decompose the problem and solve the sensor-shooter problem in an integrated fashion. This theory not only helps us analyze the tradeoffs between information and transformation resources, but aids in finding the correct balance between the two. Finally, modeling the subproblem as a POMDP allows us to consider rules and relationships that would be difficult (or impossible) to implement in a conventional optimization.

REFERENCES

Atkinson, Rick, *Crusade: The Untold Story of the Gulf War*, Houghton-Mifflin, Boston, 1993.

Aviv, Yossi, and Moshe Kress, "Evaluating the Effectiveness of Shoot-Look-Shoot Tactics in the Presence of Incomplete Damage Information," *Military Operations Research*, Vol. 3, No. 1 (1997), pp. 79–90.

Bazarra Mokhtar, S., John J. Jarvis, and Hanif D. Sherali, *Linear Programming and Network Flows*, John Wiley and Sons, New York, 1990.

Brown, Gerald G., Dennis M. Coulter, and Alan R. Washburn, "Sortie Optimization and Munitions Planning," *Military Operations Research*, Vol. 1, No. 1 (1994), pp. 13–18.

Cassandra, Anthony R., *Optimal Policies for Partially Observable Markov Decision Processes*, Technical Report CS-94-14, Brown University, Providence, RI, 1994.

Cheng, Hsien-Te, *Algorithms for Partially Observable Markov Decision Processes*, Ph.D. dissertation, University of British Columbia, 1988.

Cohen, Eliot, ed., *Gulf War Air Power Survey, Vol. II, Operations and Effectiveness*, Department of the Air Force, Washington, D.C., 1993.

Cotsworth, William L., *The Conventional Targeting Effectiveness Model (CTEM) User's Manual*, AEM Services, Inc., S-93-020-DEN, 1993.

Evans, Dennis K., "Bomb Damage Assessment and Sortie Requirements," *Military Operations Research*, Vol. 2, No. 1 (1996), pp. 31–36.

Fisher, Marshall L., "An Applications-Oriented Guide to Lagrangian Relaxation," *Interfaces*, Vol. 15, No. 2 (1985), pp. 10–21.

Hauskrecht, Milos, *Planning and Control in Stochastic Domains with Imperfect Information*, Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, Mass., 1998.

ILOG, Inc., *Using the CPLEX Callable Library*, Incline Village, NV, 1997.

Institute for Defense Analysis (IDA), *Deep Attach Weapons Mix Study C4ISR Data*, unpublished working papers, December 1996.

Lewis, Richard B. H., "JFACC Problems Associated with Battlefield Preparation in DESERT STORM," *Airpower Journal*, Vol. 8, No. 1 (1994), pp. 4–21.

Lovejoy, William S., "A Survey of Algorithmic Methods for Partially Observed Markov Decision Processes," *Annals of Operations Research*, Vol. 28, No. 1 (1991a), pp. 47–65.

Lovejoy, William S., "Computing Feasible Bounds for Partially Observed Markov Decision Processes," *Operations Research*, Vol. 39, No. 1 (1991b), pp. 162–175.

Microsoft Corporation, *Visual Basic Programmer's Guide*, Document No. DD93011-1296, 1997.

Rice, Roy E., *Mathematical Formulation of the Sensor-Platform Allocation Model (SPAM)*, Working Paper, Teledyne-Brown Engineering, Huntsville, AL, 1997.

Scott, William B., "Computer/IW Efforts Could Shortchange Aircraft Programs," *Aviation Week and Space Technology*, January 19, 1998, p. 59.

Sondik, Edward J., *The Optimal Control of Partially Observable Markov Processes*, Ph.D. dissertation, Stanford University, Palo Alto, CA, 1971.

Shannon, Claude, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, 1948.

Sherrill, E. Todd, and Donald R. Barr, "Exploring a Relationship Between Tactical Intelligence and Battle Results," *Military Operations Research*, Vol. 2, No. 3, 1996, pp. 17–33.

TRAC-OAC, *TACWAR Integrated Environment Air Analyst Guide Model Version 4.0*, Fort Leavenworth, KS, July 1994.

Willstatter, Kurt, and James Barnes, "Intelligence, Surveillance, and Reconnaissance Investment Study," presentation, 66th Military Operations Research Society Symposium, Monterey, CA, June 1998.

Yost, Kirk A., "Consolidating the USAF's Conventional Munitions Models," *Military Operations Research*, Vol. 2, No. 4 (1996), pp. 53–72.

Yost, Kirk A., *Solution of Large-Scale Allocation Problems with Partially Observable Outcomes*, Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, September 1998.