A Game-Theoretic Model for Defense of an Oceanic Bastion Against Submarines

Dr. Gerald Brown
Jeff Kline
Adam Thomas
Dr. Alan Washburn and
Dr. Kevin Wood

Operations Research Department, Naval Postgraduate School

gbrown@nps.edu, jekline@nps.edu, athomas@nps.edu, awashburn@nps.edu, kwood@nps.edu

APPLICATION AREAS:
Strategic Operations, Littoral Warfare/ Regional Sea Control

OR METHODS:
Linear Programming, Network Methods, Stochastic Processes

ABSTRACT

We develop a game-theoretic model called BASTION to guide the employment of antisubmarine warfare (ASW) platforms such as ships and aircraft that are defending a stationary oceanic bastion from attack by hostile submarines. The model is an example of a two-person zero-sum game with some additional variables (ship locations) that are under the control of the maximizing defender, but known to the minimizing attacker. The attacker, knowing the ship locations, but not the locations of other platforms such as aircraft, must select a path to the bastion. The probability of detecting the attacker as it follows this path is the objective shared by the opponents.

INTRODUCTION

This paper develops a new planning aid for defensive ASW called BASTION. We envisage a Blue battle-group commander with a set of ASW platforms—these will typically comprise a group of surface ships, submarines, fixed-wing aircraft, and helicopters—that are available to search for enemy submarines. The commander must employ his forces to protect an HVU (High Value Unit, typically an aircraft carrier) from enemy (Red) submarines. The HVU and possibly some other ships are assumed to lie in a bastion of protected cells, and the object of the Red attackers is to penetrate to the bastion without being detected by Blue.

The Submarine Threat

Modern submarines, both nuclear (SSN) and diesel-electric (SSK), pose a significant threat to the U.S. Navy. Technological innovation since World War II has improved the capabilities of submarines to the point where Keegan (1986) questions the viability of surface ships in the face of submarine attacks. Air-independent propulsion now allows SSKs to operate submerged for weeks at a time, compared to only hours in World War II. SSNs are even more effective than SSKs, and improved weapons systems have greatly enhanced the lethality of both types. As a result, even a single submarine possesses the stealth and firepower necessary to put warships at risk.

More than 40 countries currently operate submarines, including several countries with large fleets (Benedict, 2006). Holland (1991) explains the effect submarine attacks might have: ''In a conflict with less than a superpower, public or political patience will run thin concerning losses or delays by submarines. The magnitude of the political catastrophe arising from the torpedoing of an aircraft carrier in a limited conflict can hardly be overestimated.''

Antisubmarine Warfare Platforms and Systems

The U.S. Navy uses multiple ASW platforms, including surface ships, aircraft of various kinds, and SSNs.

A surface ship's primary ASW sensor is its sonar, either hull-mounted or towed. Surface ships also employ radars that are capable of detecting surfaced submarines, or even submarine periscopes or masts. However, the locations of surface ships are usually detectable by submarines well in advance of any detection by the ships. In BASTION, we assume that the locations of all ships are known by Red when planning an attacking submarine's path to the bastion.

All aircraft enjoy the advantage of being much faster than submarines. Aircraft carriers operate MH-60 helicopters that include ASW among their many missions. The SH-60 helicopter, operated from cruisers, destroyers, and frigates, can drop expendable sonobuoys, and can also dip a sonar sensor into the ocean. Land-based maritime patrol aircraft (MPA) also employ sonobuoys, but in significantly larger numbers, and are equipped with a surface search radar. Every manned aircraft is equipped with human eyeballs, which remain one of the most effective ASW sensors. In BASTION we assume that all aircraft operations are inherently stealthy to Red submarines, except for the complicating feature that Red knows the locations of the bastion (the source of MH-60 sorties) and the ship hosts of the SH-60s.

A battle group typically includes one or two Blue SSNs. These ''direct support'' submarines often provide the most effective sonar search capability due to their quietness and their ability to vary search depth to adjust to the ocean's local acoustic conditions. To prevent conflicts with other Blue platforms, Blue submarines typically have exclusive use of certain regions known as Submarine Operating Areas (SOAs), and do not leave those regions without pressing reasons. The locations of these SOAs are assumed known to the attacker, but not the disposition of the SSNs within the SOAs. Red can infer the location of the SOA, more or less, from the locations of previous sub-versus-sub engagements and the lack of activity in the area by other Blue forces.

Current Planning Tools for ASW Missions

The U.S. Navy currently employs a number of tools for planning ASW missions.

The *Personal Computer-Based Interactive Multisensor Analysis Trainer* (PCIMAT), is the premier ocean acoustic analysis and planning tool available on all ASW platforms (SPAWAR, 2008). Although employable from a standalone laptop computer, an implementation of PCIMAT (the Sonar Tactical Decision Aid, or STDA) integrates its capabilities with a platform's entire fire-control system. Among other outputs, STDA offers a mission-planning module that provides a graphical representation of a platform's effective coverage area.

The ASW *Screen Planner Tactical Decision Aid* (SWDG, 2004) aids the planning of ASW screens for a battle group in transit. The planner specifies a threat submarine from a database and proposes assignments of available ASW platforms to sectors surrounding the battle group, not including MPA. The decision aid then calculates the probability of detecting the enemy submarine in each sector if it transits that sector in a straight line. The planner manually assigns platforms to sectors until he creates a solution with an acceptable probability of detection.

The *Active System Performance Estimate Computer Tool* (ASPECT) aids MPA in maximizing the effectiveness of active sonobuoy search (FAST, 2006). The planner manually specifies several sonobuoy patterns, and describes how Red can be expected to move. ASPECT then simulates approximately 500 submarine tracks in the search area, and reports the resulting sample detection probability for each pattern.

The *Operational Route Planner* (ORP) models the area search problem, seeking to identify routes for search platforms that are most likely to lead to the detection of an SSK in a specified area (Kierstead and DelBalzo, 2003; Wagner Associates, 2008). ORP simulates the SSK's actions based on a probabilistic description of enemy behavior. The planner specifies patrol-region assignments for available search platforms, and ORP heuristically optimizes search plans for multiple ASW searchers using a genetic algorithm.

None of these currently employed systems synchronously coordinates the actions of multiple ASW platforms in defense of a bastion. BASTION is intended to do that.

TWO BACK-OF-THE-ENVELOPE MODELS

Throughout the development of BASTION, we have wrestled with whether ASW search should be modeled as exhaustive or random. These are the only two viable alternatives, because our optimization ambitions for BASTION dictate a simple analytic model of some kind. To make the issues clear, in this section we outline two simple models, one based on exhaustive search and one based on random search. The exact formulation of BASTION, together with the assumptions behind it, are the subject of the next section.

For the moment, assume there are only three searching platform types: ships, subs, and MPA (helicopters are a special case because they are based on ships, so we omit them for the moment). Red knows all ship locations. Blue subs operate within an SOA that is known to Red, but otherwise Red does not know their locations. To prevent interference and even fratricide, Blue insists that only Blue SSNs operate within the SOA, and that Blue SSNs operate nowhere else. The bastion is assumed to have the shape of a circular disc with radius *r*, and the surrounding waters are assumed to be homogeneous.

Exhaustive Search Model

This is a one-dimensional model in which Blue attempts to make the perimeter of the bastion a barrier that Red subs cannot penetrate without being detected. Define the following four quantities:

- D = length that can be guarded by ships,
- S = length that can be guarded by SSNs,
- A = length that can be guarded by aircraft, and
- B = length of bastion perimeter ($2\pi r$).

The general idea is that each platform type is given responsibility for part of the barrier. If $D + S + A$ is larger than B, then the entire perimeter can be guarded and the detection probability is 1. Otherwise, because all points guarded by ships are known to Red, the SSNs and MPA must do their best to cover the remaining perimeter length $B - D$. Suppose that Blue chooses the SOA to have length L, with $L > S$. Because Red knows where the SOA is located, but not the locations of the submarines within it, the detection probability will be S/L if Red penetrates the SOA, or $A/(B - D - L)$ if Red penetrates the MPA part of the barrier (Red will avoid the ship part because it is completely covered). Blue should choose L (his only strategic decision in this model) to make the smaller of these two quantities as large as possible. As a result, the detection probability is $(S + A)/(B - D)$, with Red being indifferent between the submarine and MPA parts of the barrier. Note that Blue pays no penalty for having to reveal the SOA to Red; as long as he chooses the size of the SOA judiciously, the payoff is the same as if the Blue sub length S was simply added to the MPA length A in the first place. Note also that Blue does pay a penalty for having to reveal the ship locations, because the detection probability is smaller than $(D + S + A)/B$.

Our main objections to this model are:

- Most ASW sensors are not of the cookie cutter type. There is no distance R such that detection is certain within R, and nondetection certain at longer ranges.
- ASW platforms, particularly aircraft, cannot be assumed to be present all the time, especially in the face of enemy actions that would accompany a war in which Red attackers are attempting to attack Blue HVUs. There are many reasons for this, one of which is that so-called "ASW platforms" have other missions besides searching for Red submarines. One of them is to engage and kill Red submarines, a separate function.

Considerations such as these have classically led to the assumption that search is effectively "random," rather than "exhaustive" (Koopman, 1980). Because this assumption is essential to BASTION, we next outline a random search model for the current scenario.

Random Search Model

ASW sensors operate in two dimensions rather than one. For continuously moving sensors such as eyeballs or passive sonars, this observation has led to the definition of sweepwidth W as the effective width of a cleared strip. If the platform moves at speed V, the rate of covering area is VW. If the platform is present only f of the time, the average rate of covering area is reduced to fVW. If the target is located somewhere within area A, and if the sensor searches randomly within A, then the rate of detection is $\lambda = fVW/A$. To be precise, detections are assumed to be a Poisson process with rate λ. If multiple platforms are present, and all are searching independently at random, then the detection rates can be summed. These considerations lead to characterizing each platform type with its total rate of clearing area, rather than a guardable distance as in the exhaustive model. Define:

- D = total sweep rate of ships,
- S = total sweep rate of Blue submarines, and
- F = total sweep rate of aircraft.

In the abstract, random search amounts to distributing confetti over an area in the hope that some flake will cover the point that represents the target. Assume that the area is a ring surrounding the bastion that extends from r to some larger distance R that Blue controls (see Figure 1), and that the ring is divided into three segments, one for each of the three separate types of Blue platform. The segment areas should be selected to be proportional to sweep rates —we omit the proof— and the net effect of this observation is that we might as well simply add up the three sweep rates to get the total sweep rate $C = D + S + F$.

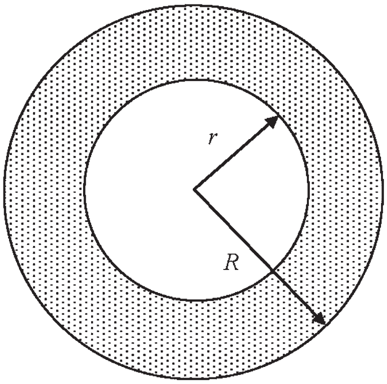

Figure 1. Blue surrounds the bastion with a ring where he searches randomly.

Blue's plan is essentially to distribute confetti at rate C over a ring that has area $\pi(R^2 - r^2)$. If Red's penetration speed is U, he will spend a time $(R - r)/U$ in this ring before arriving at the bastion. The resulting probability of detection is just the probability of at least one event in a Poisson process:

$$P = 1 - \exp\left(-\frac{C(R-r)}{\pi U(R^2 - r^2)}\right)$$
$$= 1 - \exp\left(-\frac{C}{\pi U(R+r)}\right).$$

The outer radius R is under Blue's control, but must be at least r in order to keep the ring separate from the bastion. The best value is $R = r$, and the resulting detection probability is $P = 1 - \exp(-\frac{C/U}{B})$, where B is the same barrier length defined earlier. In the limit, Blue distributes confetti inside a vanishingly small ring. We might insist that R be larger than r by an amount that is roughly the detection radius of the Blue sensors involved. Nonetheless, the essential fact is that Blue wants to defend a thin ring around the bastion.

In this model, Blue's only strategic concern is to make sure that each platform type is assigned a part of the ring that is proportional to its sweep rate. Note that ships pay no penalty for being visible to Red, because their sweep rate is just one of the terms in the sum defining C. This is a significant problem with the random search model, because ship visibility has important consequences in reality (see discussion under the heading Ships in the next section). Although BASTION is essentially a generalization of this model, it makes an exception for ships.

DEVELOPMENT OF BASTION

The U.S. Navy does its bastion planning on rectangular ''Four-Whiskey'' (4W) grids with cells whose sides are typically 5 or 10 nm. Platforms are assigned to cover cells or groups of cells, but not partial cells. BASTION adopts this rectangular reference system. Certain cells do not need to be defended because they are impassable to Red attackers (hereafter ''land'') or are part of the bastion. All other cells constitute the traversable set C, and the SOA is a subset of C. Figure 2 shows a typical categorization in BASTION.

A Red attacker is assumed to start outside the grid, and must choose a path to the bastion consisting of adjacent traversable cells, possibly including diagonal moves between cells that share a corner. Red's object is to get to the bastion without being detected. Blue's object is to detect Red before he gets to the bastion, so we have a two-person zero-sum game. The situation is assumed to be stochastically stationary, with no time limit within which Red must act.

The subscripts i and j will index cells in the grid. We define the area of cell i to be A_i, typically but not necessarily independent of i.

For modeling purposes, each traversable cell i in C is connected to each adjacent traversable cell j by a directed arc (i, j) in a network

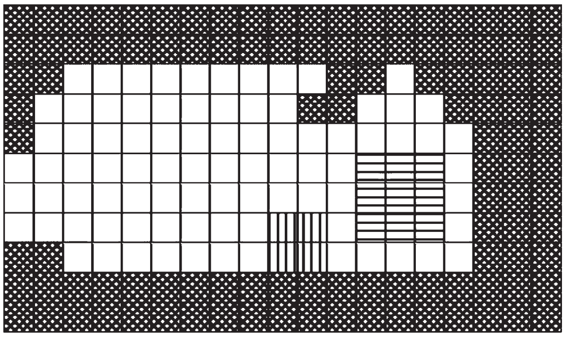

Figure 2. The grid is a rectangle of cells with latitude and longitude coordinates. Land is crosshatched and nearly surrounds a white area that Red can traverse. The bastion is horizontally marked, whereas the SOA is vertically marked. Red enters at the left border.

model, i being the tail and j being the head of the arc. The arcs are the feasible moves for the attacking Red submarine. Each arc has associated with it a time τ_{ij} that represents the amount of time required for Red to transit between i and j. These transit times are inversely proportional to the attacker's assumed, constant speed, an input to BASTION.

We let Red's transit begin at an artificial start cell i^+ that lies outside of the grid and connects to all cells through which Red can enter (traversable cells on the grid's border). Red's transit ends at an artificial terminal cell i^- that represents all cells in the bastion. The detection rate in the start and terminal cells is 0 by definition.

We denote the full network model as (C, A), where C denotes the set of traversable cells and A denotes the set of arcs. We denote Red's path through the network as $\mathbf{y}$, a vector of arcs in which the head of each arc is the same as the tail of its successor. The tail of the first arc in $\mathbf{y}$ is i^+, and the head of the last is i^-. We refer to the collection of all of Blue's ASW assignments as $\mathbf{x}$**,** and to the resulting detection rate in cell i as $s_i\,(\mathbf{x})$.

When a Red attacker is present in cell i, we assume that $s_i\,(\mathbf{x})$ is the rate of a Poisson process of detections, even though some of Blue's assets (aircraft, in particular) may operate on a schedule that is more regular than Poisson. In this we are relying on the tendency of point processes to become Poisson in the presence of complications. For example, the sum of many independent, stationary processes tends to become Poisson (Khinchin, 1960), and thinning a stationary process tends to have the same effect. A process of aircraft flights is "thinned" when aircraft are randomly assigned to patrol in specific cells. Blue is motivated to choose those cells randomly because he is playing a game with Red—as long as Red cannot predict the sequence of chosen cells, it is not particularly important if Red can predict the sequence of takeoffs and landings. In short, Blue's search for Red attackers is assumed to be "random".

Let

$$z(\mathbf{x},\mathbf{y}) = \sum_{(i,j)\in\mathbf{y}} \tau_{ij}(s_i(\mathbf{x}) + s_j(\mathbf{x}))/2. \qquad (1)$$

If we assume that half of Red's time in transiting from i to j is spent in each cell, then $z(\mathbf{x}, \mathbf{y})$ is the average number of times that Red is detected on his path from i^+ to i^-. We will refer to this quantity as the "pressure" that Blue's defensive efforts place on Red as he attempts to penetrate through to the bastion. We take this pressure to be the mean of a Poisson random variable, so the probability of (at least one) detection is 1 – exp(–$z(\mathbf{x}, \mathbf{y})$). Blue chooses $\mathbf{x}$ to maximize this payoff, while Red chooses $\mathbf{y}$ to minimize it.

The total search rate $s_i(\mathbf{x})$ in cell i is itself the sum of several components that are determined by the various platforms under Blue's command. There are four platform types in BASTION: submarines (Blue SSNs), land-based MPA, ship-based helicopters, and the ships themselves. Each platform type is discussed separately below. The entire Bastion model will then be specified in detail.

Submarines

Blue submarines operating within an SOA are the easiest platform to model, at least if their search is by passive sonar. Let RS_i be the detection rate of a submarine in cell i, normally obtained by multiplying the speed of the Blue sub by its sweepwidth and dividing by the cell's area, and let xs_i be the average number of submarines patrolling in cell i. Then the total submarine detection rate in cell i is $RS_i xs_i$. If there are NS Blue submarines in direct support, then the allocation variables are subject to the constraint $\sum_{i\in SOA} xs_i = NS$.

BASTION has two modes of operation. In the free mode, SOA is C and all platforms can operate anywhere. In the constrained mode, a specific SOA within C is set by the planner, and only SSNs can be located there.

Land-Based Aircraft (MPA)

An MPA squadron usually describes its capability in terms of a sortie generation rate GA, the average number of sorties per hour that can be launched by the squadron (a sortie is whatever happens between a takeoff and the following landing). If there is a difference between "surge" and "sustain" sortie rates, then the appropriate one here, because of the long time horizon envisioned in employing

BASTION, is ''sustain.'' Let BA_{jk} be the amount of area searched in cell j by a sortie assigned to cell k, divided by the area of cell j. The interpretation of this dimensionless number is ''the average number of detections if Red is located in cell j when a sortie is assigned to cell k.'' Our idea here is that BA_{kk} will be determined primarily by sonar search considerations, but that aircraft have a significant eyeball and radar search capability for exposed periscopes and surfaced attackers while transiting over cells other than k. The value for BA_{jk} will typically be a strong function of cell index k, with cells far away from the squadron's base having relatively small values. Determining BA_{jk} from more fundamental parameters is a significant task that we defer to Appendix A, taking BA_{jk} as given for the moment. Let xa_k be the number of MPA sorties per hour assigned to cell k. The MPA detection rate in cell j is then $\sum_k BA_{jk}xa_k$, and the applicable constraint is $\sum_k xa_k = GA$, where both sums are over all cells, possibly excepting SOA if aircraft are forbidden to search in that region.

We use the term ''squadron'' to mean all MPA operating from a given base, and will describe BASTION in the sequel as if only a single squadron were involved. If multiple squadrons are actually available, then each would be modeled as a separate asset.

Helicopters

MH-60 helicopters operate from aircraft carriers, typically located within the bastion, whereas SH-60 helicopters operate from other ships located outside the bastion. The effect of helicopter search is similar to that of MPA, except for the mobility of their bases.

To some extent Red can predict helicopter operations. Consider the operations of MH-60s from within the bastion. Might Red reason that, because the aircraft carrier is currently located in the south edge of the bastion, cells near the north edge should currently be safe from the attentions of MH-60s? Answering the question in the affirmative would require introducing an unwelcome time dimension to BASTION, because the track of the aircraft carrier(s) would have to be an input. The movements of aircraft carriers when launching and recovering aircraft are notoriously unpredictable, even to Blue, and Red would have to consider more than the current locations in planning his penetration. We therefore assume that carrier movements within the bastion are unknown to Red, except for remaining within the bastion, and that Blue takes advantage of this ignorance to maximize the on-station time of MH-60 flights. The computational effect of this in BASTION is that an MH-60 will always have its on-station endurance calculated as if it were both launched and recovered from the most favorable point in the bastion. Except for this flexibility, MH-60s are handled like MPA. In the case of SH-60s, we assume that the location of their base (a ship located outside the bastion) is known to Red. For all helicopters, we next calculate the area covered by one sortie and proceed as with MPA. The sortie generation rate GM for MH-60s should be only those sorties allocated to ASW search.

In the case of SH-60s, additional constraints are necessary to enforce the idea that a sortie cannot begin in cell i unless the helicopter's host occupies that cell (see formulation below). SH-60s typically search using a dipping sonar, an active device that in principle provides information to Red about the helicopter's location, and therefore a basis on which to predict and avoid subsequent dip locations. However, because helicopters operate at much higher speeds than submarines, the effect of this information is ignored in BASTION.

Ships

Ships are the most difficult ASW platform to model. All other platforms are sufficiently fast or well hidden that Red cannot predict their exact locations when planning his route to the bastion, but this is not true for ships. Ships reveal their locations in several ways: they are large surface vessels, and therefore subject to observation by eyeball and radar; they operate radars themselves, which are subject to intercept; their engines are noisy, which permits submarines to determine bearing at considerable distance. Finally, ships sometimes operate powerful active sonars, the signals from which can be intercepted at long distances by submarines. For these reasons, ships are the only platforms whose locations are assumed to be known to Red before planning his penetration route. This is accomplished in BASTION by requiring the variables

xf_{mi} denoting the presence of ship m in cell i to be either 0 or 1 (binary). This necessity forces BASTION to be a Mixed Integer Program (MIP), rather than a linear program.

In spite of their locations being known to Red, ships can still be effective ASW search platforms. BASTION currently models a ship's detection rate as a Gaussian plume of the form $\lambda\exp(-(r/R)^b)$, where r is the distance from the ship, λ is the maximum possible detection rate, R is a relaxation distance, and b is a shaping constant. The connection between the (λ, R, b) parameters and more fundamental parameters such as ship speed and sonar range is tenuous, especially for a platform whose location is known by the target that it seeks. Washburn (2010a) is relevant, but not definitive. By making λ and b large, one can implement the idea that the ship controls a disk with radius R that Red will not challenge. This could easily be generalized to model noncircular regions of control, such as those provided by PCIMAT or STDA.

Let S_{mij} be the detection rate in cell j of ship m located in cell i, obtained by substituting relevant parameters into the Gaussian plume model. The total rate of detection in cell j due to ships is then$\sum_{m,i} S_{mij}xf_{mi}$. Note that ships are the only platforms modeled individually; all other platforms are modeled as aggregations of identical units. The resulting proliferation of ship indices is necessitated by Red's information advantage, but is also useful because ships differ significantly from one another. The fact that xf_{mi} is required to be binary usually means that the detection rate will be wastefully high near the ship's location, but Blue cannot avoid this because the ship's location is known to Red.

After accumulating the sweep rates from all platforms in cell i, we have the total detection rate $s_i(\mathbf{x})$ as a linear function of $\mathbf{x}$ for all i, and therefore (using (1)) the detection probability 1 – exp(–z($\mathbf{x}$, $\mathbf{y}$)). We are dealing with a two-person zero-sum game in which the payoff function is a concave function of $\mathbf{x}$, while $\mathbf{x}$ (except for the integer variables representing ship locations) is constrained to lie in a compact set. Therefore (Washburn, 2003) Blue has an optimal pure strategy, and the value of the game is $\max_{\mathbf{x}}$ $\min_{\mathbf{y}}$ (1 – exp(–z($\mathbf{x}$, $\mathbf{y}$))). Although the detection probability is our ultimate concern, the pressure z($\mathbf{x}$, $\mathbf{y}$) itself will suffice for an objective function because the transformation from pressure to detection probability is strictly increasing. The object is therefore to find $\max_{\mathbf{x}}$ $\min_{\mathbf{y}}$ z($\mathbf{x}$, $\mathbf{y}$), the largest pressure that Blue can guarantee against the worst-case path. This value and Blue's optimal pure strategy $\mathbf{x}$ can be obtained by solving a MIP with variables ($\mathbf{x}$,v), where v is the maximal pressure. If we let X represent the set of feasible Blue defensive allocations (we will specify X in detail later) and Y represent the finite set of feasible Red paths, this program can be compactly expressed as MIP0:

$$\begin{aligned}&\max\ v\\&\text{subject to } z(\mathbf{x},\mathbf{y})-v\geq 0;\text{for all }\mathbf{y}\in Y\\&\mathbf{x}\in X.\end{aligned}$$

The difficulty with MIP0 is that the set Y does not scale well—the number of feasible paths increases rapidly with the number of cells. It would be better if there were a constraint for every arc in A, rather than one for every feasible path in Y. The first step in finding such a revision is to define variables u_{ij} that indicate whether arc (i, j) is in Red's path (u_{ij} = 1) or not (u_{ij} = 0). In terms of those variables, $z(\mathbf{x},\mathbf{y}) = \sum_{(i,j)\in A}\left[\tau_{ij}(s_i(\mathbf{x})+s_j(\mathbf{x}))/2\right]u_{ij}$. This expression replaces the reference to Y in MIP0 with a reference to A. Following standard techniques (Fulkerson and Harding, 1977), MIP0 can then be shown to be equivalent to MIP1 with variables ($\mathbf{x}$, $\mathbf{v}$) . MIP1 is:

$$\begin{aligned}&\max\ v_{i^-}\\&\text{subject to } v_j\leq v_i+\tau_{ij}(s_i(\mathbf{x})+s_j(\mathbf{x}))/2;\\&\text{for all } (i,j)\in A,\text{ and for all }\mathbf{x}\in X.\end{aligned}$$

In MIP1 there is one variable v_i for every cell i in C and one constraint for every arc in A, in addition to the variables $\mathbf{x}$ and the set X that constrains them. It is permissible to take $v_{i^+}=0$, in which case v_i can be interpreted as "the minimal average number of detections up to and including cell i on any path that starts in cell i^+, given $\mathbf{x}$." Variable $v_{i^-}$is the desired maximal pressure, and the detection probability is $1-\exp(-v_{i^-})$. Note that Red's path is not explicitly represented in MIP1.

In moving from MIP0 to MIP1, we have exploited the fact that z($\mathbf{x}$, $\mathbf{y}$) is a sum, and this analytic form is the case because we have assumed that various search processes are independent of each other. Without these independence assumptions, we would have to retreat to a model like MIP0 where the paths are enumerated.

The precise specification of X is in the summary formulation below. In that formulation, ''cell'' means traversable cell in C, unless artificial cells are explicitly included in the comments. All variables are nonnegative and real, except for xf_{mi}.

BASTION formulation

Indices and sets [$\sim$cardinality]

i^+ and i^-	Artificial start and end cells for Red's path
$i, j, k \in C$	Traversable cells [$\sim$500]
m	Ships [$\sim$10]
$(i, j) \in A$	Directed arcs, assumed sufficient to permit at least one path from i^+ to i^-
SOA	Traversable cells that constitutes the SOA, a subset of C
$NSOA$	Traversable cells that can be occupied by non-submarines, a subset of C

Data [units]

S_{mij}	Detection rate in cell j of ship m located in cell i [/hr]; $i \in NSOA, j \in C$
GH_m	SH-60 sortie rate by ship m [/hr]
GM	MH-60 sortie rate from the bastion [/hr]
GA	MPA sortie rate [/hr]
BH_{mijk}	Average detections in cell j by a SH-60 sortie to cell k from ship m located in cell i; $i, k \in NSOA, j \in C$
BM_{jk}	Average detections in cell j by a MH-60 sortie to cell k; $i, k \in NSOA, j \in C$
BA_{jk}	Average detections in cell j by an MPA sortie to cell k; $k \in NSOA, j \in C$
NS	Number of SSNs in direct support
RS_j	Detection rate of an SSN in cell j [/hr]; $j \in SOA$
τ_{ij}	Time required for an attacker to move from i to j [hr], artificial cells included

Variables [units]

xf_{mi}	1 if ship m is located at cell i; 0 otherwise, $i \in NSOA$
xh_{mik}	Rate of SH-60 search sorties to cell k from ship m in cell i; $i, k \in NSOA$[/hr]
xm_k	Rate of MH-60 search sorties to cell k; $k \in NSOA$[/hr]
xa_k	Rate of MPA sorties to cell $k \in NSOA$[/hr]
xs_j	Average number of SSNs searching in cell j; $j \in SOA$
s_j	Total detection rate in cell j [/hr]
v_i	Average detections up to cell i on a path that starts in i^+, artificial cells included

Constraints {dual variables}

$\sum_{i \in NSOA} xf_{mi} = 1$	Each ship m selects a cell to occupy
$\sum_{k \in NSOA} xh_{mik} = GH_m xf_{mi}$	Helicopter sortie generation limits for all m and i $\{dh_{mi}\}$
$\sum_{i \in NSOA} xa_i = GA$	MPA sortie generation limit $\{da\}$
$\sum_{i \in SOA} xs_i = NS$	Overall SSN population limit $\{ds\}$

$$s_j = \sum_{i \in NSOA, m} S_{mij} xf_{mi} + \sum_{i,k \in NSOA, m} BH_{mijk} xh_{mik} + \sum_{k \in NSOA} BM_{jk} xm_k + \sum_{k \in NSOA} BA_{jk} xa_k + RS_j xs_j \text{ for all } j$$

$v_j \le v_i + \tau_{ij}(s_i + s_j)/2;$ for all$(i, j) \in A$	Including artificial cells $\{-y_{ij}\}$
$v_{i^+} = 0$	Path starts out with no detections

Objective

Maximize v_{i^-}, the average number of detections up to the bastion.

All dual variables referenced above should be thought of as the dual variables of a linear program with the integer variables fixed at their optimal values. Thus dh_{mi} is of no interest for cells i that ship m does not occupy, because it represents the incremental value of SH-60 sorties from a cell where there is no place to take off or land, a concept of little interest to helicopter pilots. If ship m occupies cell i, then dh_{mi} is the incremental value of sorties from that ship. The dual variables of the value constraints are of interest because y_{ij} can be interpreted as the probability that Red includes arc (i, j) in his path. In principle, an optimal path selection strategy for Red could be derived from them (Ahuja, Magnanti, and Orlin, 1993; Washburn and Wood, 1995). Although we have scant interest in that mixed strategy, Red's probabilities are still of diagnostic value.

EXAMPLES AND COMPUTATIONAL RESULTS

Standard scenario

All examples considered in this section are variations of the standard scenario described below. The platform details will be completely specified here only for Blue submarines (see Appendices A and B for the rest).

In all cases we employ a 26 × 26 grid of cells, although only part of the grid will be shown in the figures below. The cell width is $L = 10$ nm. There are $NS = 2$ submarines in direct support, each of which has a speed of $V = 5$ kt and a sweepwidth of $W = 5$ nm. Red's speed is assumed to be $U = 5$ kt. We assume that Blue's velocity will at most times be perpendicular to Red's, so we take the relative speed between the two platforms to be $\sqrt{U^2 + V^2}$. Direct support submarines are assumed to be available for search only 90% of the time, so the detection rate of a Blue submarine patrolling in an arbitrary cell i is

$$RS_i = (0.9)\sqrt{U^2 + V^2}W/L^2 = 0.3182/\text{hr}.$$

There is one MPA squadron generating sorties at the rate of 0.1/hour, 24 hours per day. A given MPA sortie will be most effective in the northwest part of the grid, because the base is located in that direction. MPA and helicopter sorties are assumed effective only in the cells to which they are assigned; that is, the possibility of Blue's detecting Red while Blue is in transit is ignored.

There are two ships, named s5 and s6, each of which has an effective detection range of about 15 nm, not considering its helicopters. The model of ship effectiveness used is in Appendix B, a "Gaussian plume" that decreases gradually over the effective range. A ship is the only platform capable of detection in cells other than the cell it occupies.

Verification Examples

In our first example we assume that only submarines are available to defend the bastion. Figure 3 shows the bastion as a single cell surrounded by eight cells with p or q written in each, except that the p and q symbols are skipped in two cells containing arrows.

The back-of-the-envelope random search model introduced earlier suggests that the ideal way to protect the bastion is to form a ring around it as narrow as possible, the eight illustrated p/q cells. Let q be the average number of SSNs patrolling in each of the four corner cells, and define p similarly for the other four. The corner cells require less pressure than the others because Red has to spend more time in them on his way to the bastion. To penetrate through a p cell, as illustrated with arrows in

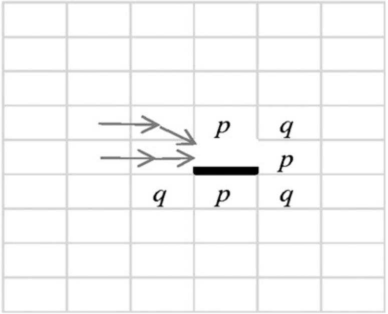

Figure 3. The bastion is a single cell surrounded by eight cells that Blue defends. Two possible paths for Red are illustrated with arrows.

Figure 3, Red must spend L/U hours. To penetrate through a q cell, Red must spend 0.5 L/U on the way in, and $0.5\sqrt{2}L/U$ on the way out, a total of $0.5(1+\sqrt{2})L/U$ hours (taking the diagonal route into the q cell would unnecessarily increase the transit time, but Red has no choice but to take the diagonal route out). To equalize the pressure on all Red routes into the bastion, the ratio of p to q should therefore be $p/q = 0.5(1+\sqrt{2}) \cong 1.2$. Because we also have $4p + 4q = NS = 2$, we can solve these two equations for p and q. The maximized minimal average number of detections will then be $z = p(RS_i)L/U$. With parameters as specified in the default scenario, this quantity is 0.1740.

Figure 4 shows a formulation and solution of this problem in BASTION. The problem is shown as one of protecting a bastion located in a large bay, but the only purpose of all the land is to limit the size of the MIP, which would be much larger if all of the ''land'' were converted to traversable cells. As long as Blue does not choose to exert pressure in any cell bordering land, as is the case in the solution shown by the + marks, the configuration is effectively open ocean. The BASTION objective function agrees exactly with the analysis given above, but the solution does not—submarines patrol in four unanticipated additional cells. BASTION has discovered an alternative optimal solution where Red could move into a p cell diagonally and still be detected exactly z times, just as he could using the two routes portrayed in Figure 3. Because the achieved z is the same as in the analytic solution given above, we take this as evidence that BASTION performs as intended.

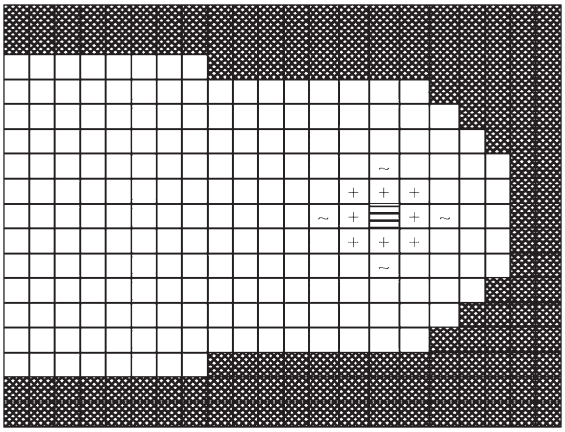

Figure 4. BASTION finds an optimal solution for the same situation as in Figure 3, but with unexpected positive allocations in the cells labeled ~.

Consider next a scenario (not illustrated) in which the only available platforms are MPA, and all cells are land except for a 7 × 2 set located near the western border. The seven cells actually on the border are traversable, whereas the seven cells just to the east are the bastion. The optimal solution has the MPA presence in the seven traversable cells increasing from north to south, which at first seems odd because MPA sorties are more effective in the north (the MPA base is in the northwest). However, this tendency to search mostly where one is least efficient is a known characteristic of game theoretic solutions to search problems, so we take this to be further evidence of verity. By emphasizing cells in the south, the MPA make Red indifferent among seven paths to the bastion.

SOA Selection Example

BASTION can be used as an aid in locating an SOA. Figure 5 shows an example where the bastion has been located north of a continent and south of two islands, hoping to take advantage of considerable land in the proximity. The SOA has been located between the continent and the eastern island in the hope that other forces will seal off the western approach and the strait between the islands. The optimal solution is also partially shown in Figure 5, using s5 and s6 to locate the two ships and + to indicate activity by MPA and helicopters.

The solution may appear odd. Note that Red can pass between the islands without fear of detection by aircraft, as long as he stays to the west, and that there appears to be a large gap between s6 and the north edge of the continent. The gap is illusory, however, because s6 is capable of long-range detections, and Red gains nothing by passing between the islands.

Blue's strategy might be called ''protect the west with ships and make Red pay for getting to the east of the bastion.'' The MPA activity along the north edge of the continent protects against Red's making an end run around s6 to get east. The submarine activity within the SOA also blocks access from the east. The maximized objective function is 0.3213.

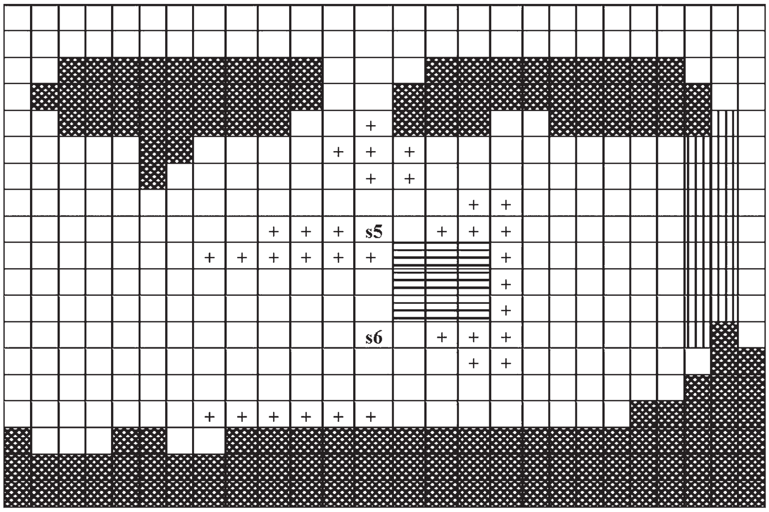

Figure 5. The SOA is located in the east, expecting to block access from that direction. The optimal solution has two ships located as pictured, with aircraft activity shown by + marks.

There is a strategic question in situations such as that depicted in Figure 5. Should one seal off the region by protecting the various gaps between land masses, or just retreat to surrounding the bastion with search activity, as one would in the open ocean? The pictured SOA almost enforces the former viewpoint, because submarines are dedicated to protecting one of the gaps. To explore which viewpoint is correct, one can simply delete the SOA and run BASTION in its free mode. Doing so in this case reveals a solution in which all search platforms retreat to surrounding the bastion, with an increase in the objective function to 0.3545. Although this increase would probably not be large enough to make Blue feel comfortable, it is still a significant improvement. The free solution has submarines and other platforms being jointly active in some cells, but an objective nearly as large can be found by locating the SOA approximately where the submarines are located in the free solution. Figure 6 shows the result. The objective function decreases slightly to 0.3420. Note that the SOA is not a connected region in Figure 6. Making it connected would enforce an additional decrease in the objective function.

We have found various graphics to be of use in debugging and understanding some of the

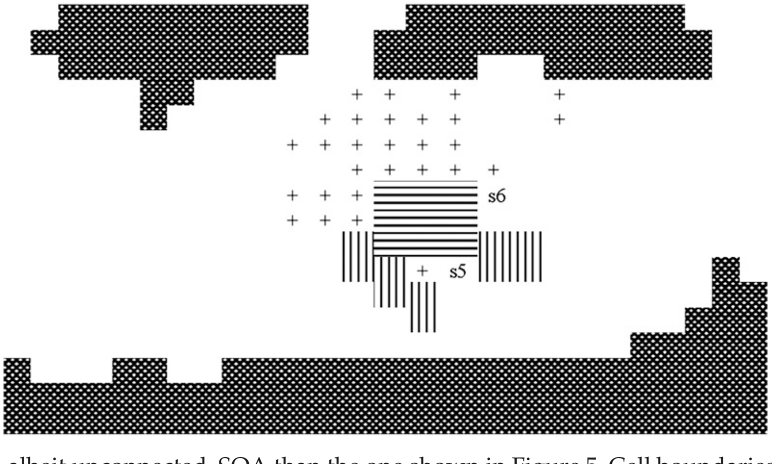

Figure 6. A better, albeit unconnected, SOA than the one shown in Figure 5. Cell boundaries omitted for clarity.

plans produced by BASTION. One of them is shown in Figure 7, a plot for the same scenario shown in Figure 6 that shows v_i in the various cells. As Red approaches the bastion, regardless of the path chosen, he must pay a higher and higher price to achieve proximity.

Computational Comments

The most computationally stressful situation would seem to be when there is no land, the bastion consists of a single, central cell, and there is no SOA to limit the activities of any platform. Such a case generates an instance with 915,302 decision variables, 1,350 of these binary, and 6,548 constraints. On a Lenovo T510 laptop, problem generation in GAMS (GAMS, 2008) requires about six minutes, and optimization with CPLEX 12.2 (ILOG 2007) MIP to a 0.1% integrality gap requires about 1.5 GB of memory and three minutes. However, it turns out that there are problems with fewer decision variables that are much more difficult, especially if we insist on a pure optimal (i.e., 0% gap) solution as we have done for the examples reported above. We have found examples requiring hours, rather than minutes, including the just-discussed SOA examples.

CAVEATS AND EXTENSIONS

We have made many assumptions and approximations in the course of developing BASTION, some of which are at odds with reality. In this section we describe some of these difficulties, suggest remedies, and also suggest how BASTION might be modified or extended.

Ship Difficulties

All platforms engaged with defending the bastion will have missions other than ASW, but the conflict between objectives is likely to be greatest for ships, which have important roles in command and control, power projection, surface defense, and air defense. Ship locations that are optimal for ASW may very well limit the ship's effectiveness in other roles. The only remedy that retains BASTION's role as a strictly ASW tool is to allow the planner to locate the ships as he wishes, using BASTION to plan the employment of other assets to fill in the ASW capability around whatever the ships are able to offer. This is easily done, and even has the benefit of reducing BASTION to a linear program, rather than a MIP. BASTION's ability to optimize ship locations still serves to quantify the sacrifice that ASW must make to the other roles.

Another difficulty is that we have not found a simple model that can quantify a ship's detection rate in terms of more fundamental quantities such as speed and detection range, whether through the Gaussian plume model or any other. We are still looking.

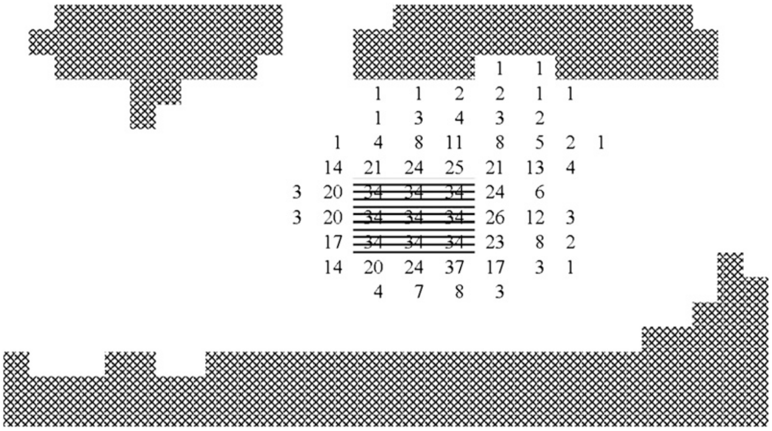

Figure 7. This figure shows $100v_i$, rounded to the nearest integer. The bastion has 34 written in it because the objective function is 0.3420. The other cells show how detections accumulate as Red approaches the bastion. Cell boundaries omitted for clarity.

MPA Difficulties

We earlier argued that $s_i(\mathbf{x})$ should be taken to be the mean of a Poisson process, but there are some features of ASW that make the assumption problematic. Chief among these is the "pulse problem" that arises when platforms concentrate ASW effort in time and space. This is especially true of MPA. When the cell dimension L is small, a single MPA sortie can quickly cover an area that is much larger than that of the sortie's assigned cell. If Red is unfortunate enough to be in the cell at the same time as the sortie, he will surely be detected, but otherwise Blue will wish that he had some way of spreading all that confetti over multiple cells to avoid overcoverage. Except for ships, BASTION analytically permits Blue to do exactly that by using infinitely divisible allocation variables, but how can BASTION's results about sortie allocation be used to provide practical guidance to MPA? We suggest two possibilities.

One way to "fix" the pulse problem is to let the MPA squadron itself solve it. The squadron is not given a grid showing the rate of flying sorties to various cells, which would be the direct output from BASTION. Instead, the BASTION output to the squadron is a grid showing the fraction of area covered that is in each cell, symbolically $fa_j \equiv \sum_k BA_{jk}xa_k / \sum_{j,k} BA_{jk}xa_k$, together with advice to "Fly as many sorties as you can, and make it so that cell j gets a fraction fa_j of the total area searched." Note that the numerator of this fraction is the only occurrence of the variables xa_k in the objective function. This less-specific advice leaves it to the squadron to formulate missions and fly sorties. The underlying assumption is that the squadron will find some way of (nearly) retaining the efficiency of the single-cell sorties used in BASTION, while simultaneously avoiding overcoverage. As long as this assumption is correct, BASTION can be a useful operational tool without dealing directly with MPA tactics.

Another way to fix the pulse problem is by assigning platforms to missions, rather than cells, a mission being a sequence of cells together with a detailed program of activity in each visited cell. In this manner a large pulse of covered area can be spread out over enough cells to prevent overcoverage in any given cell. This method has the advantage that the activities suggested by the optimal solution will always be reasonable because only reasonable activities (missions) are considered in the first place. It is certainly implementable—one merely has to let the index i on xa_i. refer to a larger set of missions than those that are named for a single cell. We have experimented with this successfully (Thomas 2008, Pfeiff 2009); however, it must be employed carefully. The number of missions is potentially enormous, and also BASTION has a tendency to prefer missions that overcover a small number of cells, whether those missions are realistic or not. Indeed, if the mission set includes all single-cell missions, then BASTION will exclude all others in its optimal solution (see Appendix C for a proof of this). Thus care must be exercised to provide a rich set of missions, all of which avoid overcoverage. Because overcoverage is a gradual phenomenon, rather than a sudden one, and given BASTION's tendencies, construction of a good multicell mission set is problematic. For the moment, BASTION manipulates only single-cell missions.

SOA (SSN) Difficulties

BASTION currently models SSNs as a class, just like aircraft, but there are good reasons for modeling them individually. Like ships, they tend to be present in small numbers and have much lower speeds than aircraft. We have experimented with individual SSNs, and rather than require SOAs as input, to recommend SOAs by specific assignment of these SSNs to cells. We can optimally partition search cells into a number of SOAs. Each SOA can be planned to be patrolled by some number of SSNs (say, one or two each), these patrols can be planned to invest some minimum and maximum search pressure in each cell, and the diameter of each SOA can be limited. Empirically, these embellishments do not noticeably add computational effort to the optimization. Similar means could be employed to shape patrol areas for MPA, but we have not implemented this.

Multiple SSKs

Throughout we have referred to "the" penetrating Red SSK. If there are actually several Red SSKs, the same analysis applies as long as the penetration attempts are well separated in time.

However, Red has every motive to locally overwhelm Blue's defenses by making multiple, simultaneous penetration attempts. The fraction of successful penetrations in that case could substantially exceed the detection probability predicted by BASTION. We view the task of modifying BASTION to account for cooperative Red penetration tactics as formidable, and have no plans to do so.

If Red possesses multiple types of submarines, then Blue should plan against the worst of them.

The 4W Grid May Be an Unsupportable Historical Artifact

The square search regions of U.S. Navy 4W grids may have been historically easy to specify by boundary coordinates, but they complicate estimates of search effectiveness that tend to be circular, rather than rectilinear. The tessellated hexagonal region used by many land warfare models appeals here.

Given the introduction of new, long-range submarine-launched antiship weapons, we anticipate the operational necessity to increase the number of cells that have traditionally been employed to enable patrol of much greater sea-space.

Scaling Up for Larger Numbers of (Unmanned) Search Platforms

Future bastion planning problems may include scores of high-endurance unmanned helicopters, rather than just a few manned ones, and perhaps more unmanned autonomous submarine searchers. As long as such platforms are modeled collectively, as all aircraft are currently modeled in BASTION, this expansion presents no computational difficulty.

If such unmanned submarine searchers are viewed as expendable, it is also possible to include the decision whether each should search passively (and covertly), or actively (revealing itself at some risk, but with much enhanced search effectiveness). Thomas (2008) pursues this embellishment for SSN searchers. BASTION's treatment of SSNs does not include this option, implicitly assuming that an SSN will not search in a manner that would reveal its location.

SUMMARY

BASTION's function is to help plan the defense of a stationary oceanic segment from attack by submarines. Blue first locates his ships, and then the rest of the Blue forces engage in a two-person zero-sum game with Red. The principal summary statistic is BASTION's objective function, the probability that an optimally arranged Blue defense will detect an optimally operated Red submarine before it can penetrate to the bastion.

Blue's problem has aspects of an assignment problem in which the various platforms attempt to accomplish an overall mission while each platform type does what it is efficient at. BASTION can partition the action space, providing guidance to each platform type about how to best operate cooperatively with the other platforms. This function is particularly important for Blue submarines, where BASTION can be of use in locating and sizing an SOA.

The capabilities of modern computers and software permit the solution of realistically scaled problems in reasonably quick response times, as we have shown by example.

REFERENCES

Ahuja, R., Magnanti, T., and Orlin, J., 1993, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, NJ.

Benedict, J., 2006, ''ASW Lecture to Naval Postgraduate School,'' presented at Menneken Lecture, Naval Postgraduate School, Monterey, CA, 16 November.

FAST (Fleet Assistance and Support Team), 2006, ASPECT User's Guide, Naval Air Systems Command.

Fulkerson, D., and Harding, G., 1977, ''Maximizing the Minimum Source-Sink Path Subject to a Budget Constraint,'' *Mathematical Programming*, **13**, pp. 116-118.

GAMS, 2008, On-line Documentation, http://www.gams.com/docs/document.htm, accessed 25 May 2011.

ILOG, 2007, CPLEX 11 Solver Manual, http://www.gams.com/dd/docs/solvers/cplex.pdf, accessed 25 May 2011.

Khinchin, A., 1960, *Mathematical Methods in the Theory of Queueing*, Hafner, New York, chapter 5.

Kierstead, D. and DelBalzo, D., 2003, "A Genetic Algorithm Approach for Planning Search Paths in Complicated Environments," *Military Operations Research*, **8**(2), pp. 45-59.

Koopman, B., 1980, *Search and Screening*, Pergamon Press, New York.

Pfeiff, D.M., 2009, "Optimizing Employment of Search Platforms to Counter Self-Propelled Semi-Submersibles," M.S. Thesis in Operations Research, Naval Postgraduate School.

SPAWAR, 2008, PCIMAT Version 7.0 User's Manual, SPAWAR Systems Center San Diego.

SWDG (Surface Warfare Development Group), 2004, ASW Screen Planner TDA User Manual—TACMEMO SWDG 3-21.3-04, SWDG Norfolk, VA.

Thomas, A. J., 2008. "Tri-level Optimization for Anti-Submarine Warfare Mission Planning," M.S. Thesis in Operations Research, Naval Postgraduate School.

Wagner Associates, Inc., 2008, ORP Algorithm Description, http://www.wagner.com/technologies/missionplanning/ORP/, accessed 25 May 2011.

Washburn, A., and Wood, K., 1995, "Two-Person Zero-Sum Games for Network Interdiction," *Operations Research*, **43**(2), pp. 243-241.

Washburn, A., 2003, *Two-Person Zero-Sum Games*, 3rd ed., Institute for Operations Research and the Management Sciences, Linthicum, MD, p. 85.

Washburn, A., 2010a, "Barrier Games," *Military Operations Research*, **15**(3), pp. 31-41.

Washburn, A., 2010b, "A Multistatic Sonobuoy Theory," Naval Postgraduate School Technical Report NPS-OR-10-005.

APPENDIX A: DETERMINING THE AREA COVERED BY A GIVEN MPA SORTIE

We assume that MPA search using fields of multistatic sonobuoys where some buoys are sources and some are receivers. Sonobuoys remain functional for some time after they are activated, so there are two modes of detection: a sonobuoy might detect a submarine as soon as it is dropped (mode 1) or the moving submarine might run into it later (mode 2). Sources and receivers need not be colocated, but, if they are, the detection radius determines d, one of the inputs. This radius will typically depend on the cell in which the buoys are located. Other required inputs are listed below with [nominal values]:

- $d =$ detection radius [2 nm];
- $D=$ distance from MPA base to cell [1000 nm];
- $V =$ MPA transit speed [330 kt];
- $U =$ submarine transit speed [5 kt];
- $T =$ MPA endurance [12 hr];
- $t =$ time between sonobuoy fields while on station [1 hr];
- $s =$ number of sources in a field [8]; and
- $r =$ number of receivers in a field [32].

Washburn (2010b) shows that the equivalent area covered by a multistatic field containing s sources and r receivers is $a = 0.8\pi d^2\sqrt{rs}$ [160.9 nm^2]. Because there is a new sonobuoy field laid every t hours as long as the MPA is on station, the mode 1 area covered by one sortie is $A_1 = a(T - 2D/V)/t$ [955.3 nm^2]. To account for mode 2 detections, we assume that the useful life of a sonobuoy is t, the same as the time between sonobuoy fields, and that the effective diameter of the region covered is $2\sqrt{a/\pi}$ [7.155 nm]. The area covered by type 2 detections is then the product of the length of time on station, the submarine speed, and the effective diameter: $A_2 = (T - 2D/V)2U\sqrt{a/\pi}$ [425.0 nm^2].

Note that the mode 2 area is proportional to the submarine speed, whereas the mode 1 area is independent of submarine speed. A Red submarine that is aware of an aircraft overhead, but does not know exactly what the aircraft is doing, would be well advised to slow down. Because the submarine's state of awareness is unlikely to be known with any accuracy by the Blue planner, this makes the calculation of the mode 2 area problematic. A conservative planner might ignore the possibility of mode 2 detections.

APPENDIX B: DETAILS OF THE STANDARD SCENARIO

The standard scenario used for generating our examples uses the MPA parameters of Appendix A, except that the distance D is not always 1,000 nm. Instead, D is the distance to the relevant cells from an MPA base located

800 nm west and 900 nm north of the NW corner of the grid. The MPA sortie generation rate is 2.4/day.

The cell dimension is $L = 10$ nm, and Red's speed is 5 kt.

Each ship has four parameters, three for the Gaussian plume plus an SH-60 sortie generation rate *GH*. Table 1 shows the data.

Table 1. Ship Parameters

Ship	λ	*R*	*b*	GH
s1	.2/hour	10 nm	2	1.5/day
s2	.1/hour	20 nm	2	1.8/day

MH-60 helicopters are assumed to carry 10 monostatic sonobuoys with a detection radius of 1 nm. They transit at 150 kt, and have an endurance of 3 hours. As long as the center of the destination cell is within range, an MH-60 can cover an area of $10\pi(1\text{ nm})^2$. Because the cell dimension is 10 nm, the average number of times a Red submarine in the chosen cell will be detected is 0.314. The bastion generates MH-60 sorties at the rate of 1.6/day.

SH-60 helicopters search by dipping a sonar with a range of 0.5 nm. Dips occur every 0.5 hour while the aircraft is on station. Endurance is 4 hours, and transit speed is 150 kt, so the area covered by a sortie to range r is $\pi(0.5)^2(4 - 2r/150)/(0.5)$.

Blue SSNs are as described in the ''Development of BASTION'' section.

APPENDIX C: PROOF THAT SIMPLE SORTIES ARE DOMINANT

The text claims that simple sorties are dominant when a given rate of sortie generation must be split among missions that include simple sorties. It is equivalent to prove that a given sortie is dominated by a probabilistic mixture of simple sorties. Assume, then, that an aircraft is located at point 0, with endurance T. A collection of m additional points i is given, with d_{ij} being the time required to fly from i to j. For a given flight, the aircraft must decide which points $(0, \ldots, i, j, \ldots, 0)$ to visit, and the amount of time x_i to spend monitoring each point in the flight, subject to the constraint that $\sum_{i,j} d_{i,j} + \sum_i x_i \leq T$. Mixed strategies are permissible.

Theorem. Assume that the triangle inequality holds: $d_{ij} \leq d_{ik} + d_{kj}, \forall i, j, k$. Then there exists a mixture of flights visiting a single cell (simple flights) that will dominate any flight visiting multiple cells, in the sense of spending at least as much time in every cell, on the average.

Proof. For a simple flight visiting cell i, let the time available for search in cell i be $X_i = T - d_{0i} - d_{i0}$, which without loss of generality we can assume positive in every cell. Now consider a flight that visits points $(0, 1, \ldots, n, 0)$, in that order, spending a time x_i at point i. Let

$$x \equiv \sum_{i=1}^{n} x_i.$$

If the flight is feasible, we must have $T \geq d + x$, where $d \equiv d_{0,1} + d_{1,2} + \ldots + d_{i-1,i} + d_{i,j+1} + \ldots + d_{n-1,n} + d_{n,0}$ is the total time spent in transit. We will show that this flight is dominated by a particular mixture of simple flights, in the sense that the mixture spends at least as much time at each of the n points, on the average. This is trivial if $x = 0$, so assume $x > 0$.

For $1 \leq i \leq n$, we have, after multiple applications of the triangle inequality, $d_{0,i} \leq d_{0,1} + d_{1,2} + \ldots + d_{i-1,i}$ and $d_{i,0} \leq d_{i,i+1} + \ldots + d_{n-1,n} + d_{n,0}$.

Therefore $d_{0i} + d_{i0} \leq d$, and, because $T = X_i + d_{0i} + d_{i0}$, we have $X_i \geq x$. Let $K = \sum_{i=1}^{n} x_i/X_i$. Then $K \leq \sum_{i=1}^{n} x_i/x = 1$. Now let the probability that a simple flight of type i is used be y_i, with $y_i = x_i/(KX_i)$. It is a simple matter to confirm that $(y_1, \ldots, y_n)$ is a probability distribution. But the expected time spent at point i by this mixed strategy is $y_i X_i$, and $K \leq 1$, so the mixture spends at least x_i at point i, on the average. Because i is arbitrary, this completes the proof. ■