(1)

Naval Research Laboratory

Washington, DC 20375-5320

NRL/FR/5750--94-9721

On the Integrated Scheduling of Hardkill and Softkill Assets Using Dynamic Programming

Douglas W. Oard
Anthony Ephremides

University of Maryland

Sheldon I. Wolk

Advanced Techniques Branch
Tactical Electronic Warfare Division

July 18, 1994

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE CNLY *(Leave Blank)*	2. REPORT DATE July 18, 1994	3. REPORT TYPE AND DATES COVERED Jan 1992 - Dec 1993

4. TITLE AND SUBTITLE On the Integrated Scheduling of Hardkill and Softkill Assets Using Dynamic Programming	5. FUNDING NUMBERS
6. AUTHOR(S) Douglas W. Oard, Sheldon I. Wolk, and Anthony Ephremides	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Washington, DC 20375-5320	8. PERFORMING ORGANIZATION REPORT NUMBER NRL/FR/5750--94-9721
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research Arlington, VA 22217-5660	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The problem of integrated employment of cruise missile defenses by a single ship is considered in this report. Two defensive systems, surface to air missiles and chaff, are examined, and a mathematical model of their performance is developed. An optimal scheduling problem is posed using this model, and a dynamic programming solution is developed. The computational complexity of this solution is beyond the capability of current computer facilities, therefore several simplifications are proposed. The study concludes with a discussion of the potential for application of heuristic techniques to this class of optimization problem.

[illegible]

14. SUBJECT TERMS Resource allocation Scheduling Dynamic programming Optimization	15. NUMBER OF PAGES 42
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

ON THE INTEGRATED SCHEDULING OF HARDKILL AND SOFTKILL ASSETS USING DYNAMIC PROGRAMMING

1 INTRODUCTION

In this report we investigate the Anti-Ship Cruise Missile (ASCM) point defense problem. Our focus is the integrated employment of defensive systems that have a potential for harmful interference. Because most shipboard defensive systems exploit the electromagnetic spectrum, the potential for interference exists between many of these systems. Often the adverse effects of this interference can be eliminated during system design or through retrofit programs. When this is not practical, simple and effective policies have been developed for cases in which system effectiveness is restricted to disjoint range bands [1]. The case of overlapping effective regions is considerably more complex and less well understood, and it is the focus of this study.

Modern ASCMs rely on a combination of high speed, low altitude, and internal guidance to reduce the period between ASCM detection and ASCM impact to under one minute. The concept of defense in depth, illustrated in Fig.1, has been applied to the design of defensive systems to increase survival probability. Multiple systems that use different techniques and are effective in different zones are used to minimize the probability that the ASCM will reach the ship. Outer zone systems operate beyond the range of on board sensors and are provided by other platforms. Point defense systems are restricted to the middle and inner zones. Middle zone systems are typically effective at ranges from the horizon down to a few kilometers, and inner zone systems typically operate within a few kilometers.

Systems that work by disrupting ASCM guidance, known as softkill systems, are most effective in the middle zone. This is the same zone in which medium range "hardkill" systems such as Surface to Air Missiles (SAM) are employed. Softkill systems are designed to disrupt ASCM guidance through electromagnetic effects, but sometimes it is not possible to eliminate the interaction between a softkill system and other defensive systems. To better understand the effect of this interaction on the effectiveness of defensive system employment policies, we will investigate the interaction between one softkill system, chaff, and one middle zone hardkill system, a SAM system. We chose a SAM system because other hardkill systems have demonstrated more limited effectiveness in the middle zone. Our choice of chaff was motivated by its widespread availability and by the difficulty of finding a SAM firing schedule that does not unduly reduce its effectiveness.

We begin by developing system effectiveness models for the chaff and SAM systems against a single ASCM. These single ASCM models provide the basis for our development of a multiple-ASCM engagement model that we use to compute the probability of surviving an attack for a given defensive system employment policy. We then investigate algorithms to find optimal policies. Two approaches to the development of an optimal policy, exhaustive search and dynamic programming, are presented.

Manuscript approved March 9, 1994.

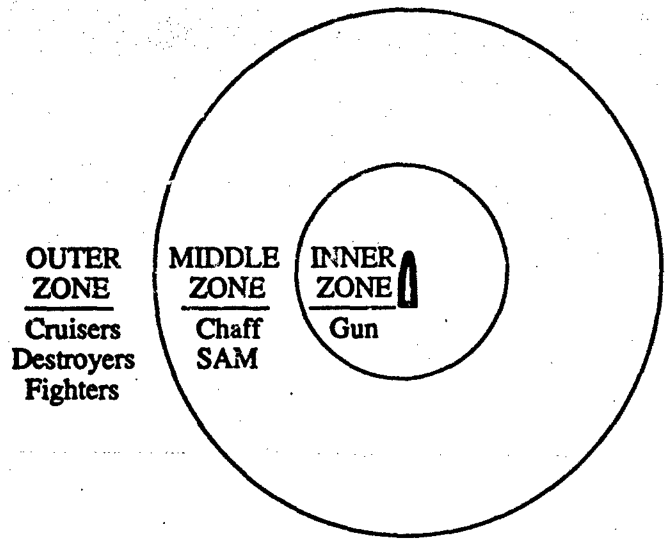

Fig. 1 — Defense in depth

2 SYSTEM EFFECTIVENESS MODELS

To understand the interaction between the SAM and chaff systems, it is necessary to understand the operation of each in some detail. Since our goal is to understand the consequences of harmful interference, we will introduce only those details necessary to quantify the effects of that interference. Furthermore, where the interference can be eliminated by tactics that do not reduce system effectiveness we will adopt those tactics to focus our attention on a single significant interaction. While practical application of this research will require more detailed models, our restricted focus should be sufficient to develop an understanding of the effect of such interactions on the effectiveness of defensive system employment policies.

2.1 ASCM Guidance

Since the purpose of the chaff is to deceive the ASCM's guidance system, we begin our discussion with a brief review of ASCM guidance. Modern ASCMs use a variety of guidance techniques, and some use multiple sensors. At ranges in the middle zone, radar is the most common sensor used for ASCM guidance. Accordingly, we model a radar guided ASCM.

A radar seeker operates by radiating microwave energy in the direction of the target and process the reflected signal to determine the target's location. The seeker that we model is a low-resolution, monopulse seeker that uses leading edge tracking. This type of seeker processes signals returned from a relatively small region in range, which is known as the "range gate." The attacker programs the initial size and location of the range gate to ensure that the ship is contained within it. After detecting the ship, the seeker periodically measures the energy received in each part of the range gate and then updates the position of the range gate so that the target remains within it. Leading edge tracking is a simple countermeasure against pulse delay techniques that might be used by a defender. However, a leading edge tracking seeker is susceptible to deception by chaff. To implement leading edge tracking, the ASCM seeker attempts to center the range gate on the nearest edge of the target by biasing the tracker to place the leading edge of the signal near the center of the range gate.

2.2 Chaff System

Chaff rounds are deployed ballistically from the ship by using a launcher with a fixed position and orientation. After a preprogrammed delay, the chaff round blooms into a large cloud of conducting strips that float slowly to Earth. The strips are designed to efficiently reradiate incident electromagnetic energy in the range of frequencies used by the ASCM seeker. Because the range gate initially includes the ship, we must select a launcher orientation and a bloom delay that places the chaff cloud near the ship when it blooms, if we hope to deceive the ASCM seeker. We can then increase the range separation (along the ASCM ship axis) by moving the ship so that the ASCM seeker must eventually choose betw .ı the two. If the seeker chooses the chaff cloud, and we also increase the cross-range separation between the chaff cloud and the ship sufficiently, the ASCM will miss the ship. We call this a successful "seduction." Unfortunately, the success of a seduction attempt is difficult to determine while the ASCM is in the middle-zone, because there is no direct way for the defender to measure the position of the ASCM's range gate, and the trajectory change caused by a successful seduction is very small at those ranges.

The defender can exploit the leading edge bias of the ASCM seeker by initially placing the chaff cloud between the ship and the ASCM and choosing a ship velocity vector that simultaneously increases both the range and cross-range separation between the two objects. Figure 2 shows this geometry with a reference frame centered on the moving ship. The time required to establish the required cross-range separation establishes the minimum range at which a seduction can be effective. We call this time τ_I, the time spent by the ASCM in the inner zone where chaff is ineffective. Similarly, we refer to the time from ASCM detection until the ASCM enters the inner zone as τ_M, the time spent by the ASCM in the middle zone.

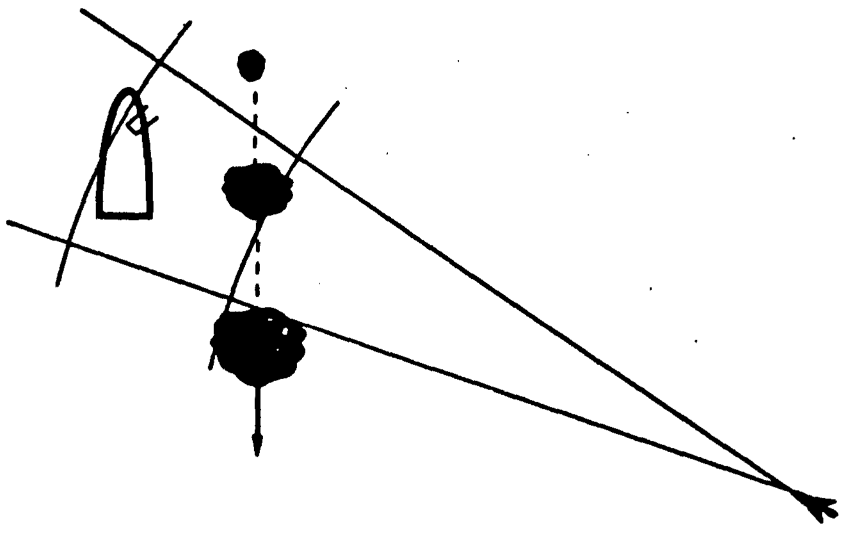
Fig. 2 — Chaff seduction geometry

When more than one object is in the range gate, the behavior of the ASCM seeker is based on the combined signal return from all of the objects. The signal returned by the ship has been observed to undergo large amplitude fluctuations as ship motion and multipath effects combine to produce constructive and destructive interference in the signal returned by a small number of dominant scatterers in different locations on the ship. The observed amplitude fluctuations in the signal returned by the chaff cloud are much smaller because the chaff cloud is made-up of a large number of small scatterers that experience more consistent motion.

Instead of presenting a choice of range gate positions, we could force the ASCM to choose between two objects that remain within the range gate but separate in bearing. Establishing the required bearing separation requires either a very large cross-range separation or a very close approach by the ASCM. The required cross-range separation is difficult to achieve while the ASCM is in the middle zone, because the available time and ship speed are limited. For this reason "bearing seduction" is an inner zone phenomenon. Hence, we will restrict our attention to range gate seduction, which we henceforth refer to simply as seduction.

To develop an analytic model of seduction effectiveness, we performed extensive computer simulation of typical seduction scenarios. The C-based Routines for Understanding the Interaction between ships, electronic warfare and missiles (CRUISE Missiles) simulation developed by the Naval Research Laboratory was used. A ship model for a destroyer was used in conjunction with a chaff model for super rapid blooming offboard chaff and an ASCM seeker model for a subsonic sea-skimming radar-guided missile using a monopulse seeker and a low-resolution leading edge range tracker. The relatively low sea state (0.5 meter root mean square wave height) we chose increased sea surface reflections and thereby created significant multipath fading.

Figure 3 shows the position of the inner and outer edges of the ASCM's range gate as the ship-chaff separation increases during a typical simulation run. The range extents of the ship and the chaff cloud are plotted for reference. In repeated simulations, the ASCM seeker seemed to show a marked preference for tracking either the ship or the chaff, even when both were in the range gate. The range gate was observed to shift from the ship to the chaff cloud when the signal return from the ship underwent a deep fade, and to shift back when the signal return from the ship again dominated that of the chaff cloud. Once the separation became so large that only one object was in the range gate, no further transitions were observed.

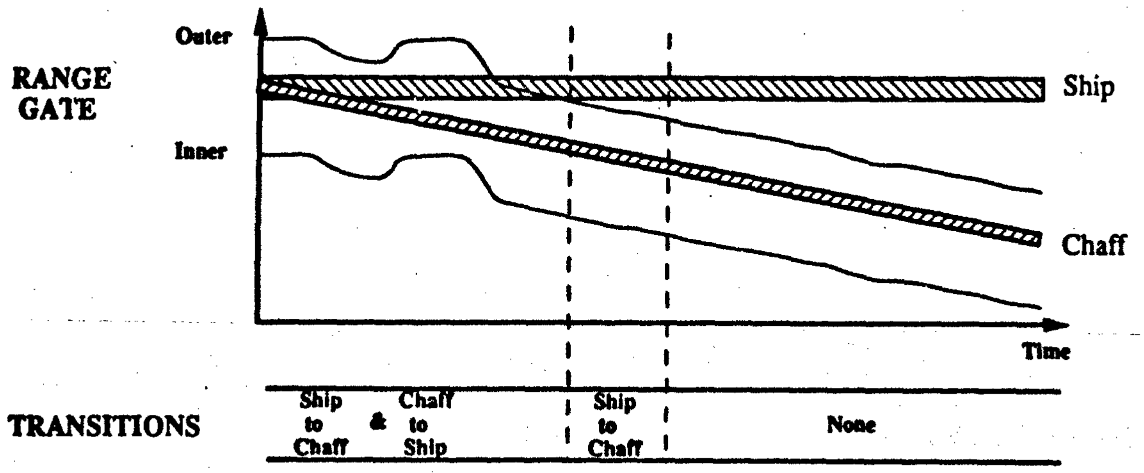

Fig. 3 — Range gate position

To understand this behavior we must discuss radar seeker design in more detail. The optimum ratio of peak signal to mean noise is obtained when the ASCM's radar receiver uses a filter that is matched to the transmitted pulse. For the rectangular transmitted pulse typically used by a low-resolution seeker, a single point scatterer produces an output signal from the matched filter that initially increases linearly with time for the duration of the transmitted pulse and then linearly decreases with time for the same period. Objects with range extent will result in an output that is the magnitude of the coherent sum of several such triangles, each with a potentially different amplitude, phase, and time delay. Figure 4 is a conceptual diagram that illustrates the time relationship between the output of a matched filter for the chaff and the ship, as they separate in range.

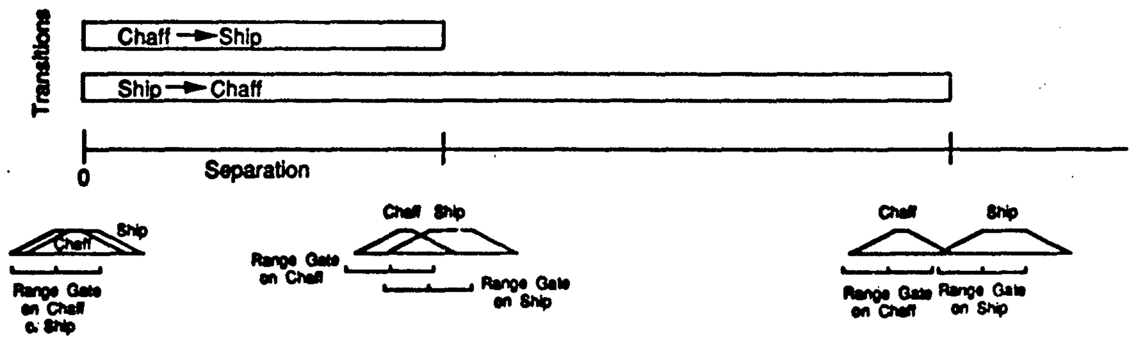

Fig. 4 — Range gate behavior

To capture all of the matched filter output associated with a single scatterer, the range extent of the range gate is chosen to correspond to twice the duration of the transmitted pulse. To implement the leading edge tracking feature, we integrate the output of the matched filter over the range gate, assigning twice as much weight to the output in the first half of the gate, as we do to the output of the filter in the second half of the gate.

The position that the range gate would assume, if only the chaff or only the ship were present and if the signal return from each object were uniformly distributed in range, is shown for reference in Fig. 4. The range gate positions actually shift slightly as the relative predominance of the scatterers that comprise the two objects change, but such shifts are small compared with the size of the range gate. When the signal returned by the ship fades severely, the range gate moves to the chaff-based position, if enough of the matched filter output resulting from the chaff is still in the range gate at the time of the fade.

As the range separation between the chaff and the ship increases, the ability of the range gate to move between them becomes more constrained. When they are superimposed in range (at the far left in Fig. 4), the range gate can move easily between positions based on a dominant return from the chaff or the ship. Eventually a separation is reached beyond which the leading edge bias of the tracker precludes chaff-to-ship transitions. Ship-to-chaff transitions remain possible, however. Finally, a sufficient separation is reached to prevent any transitions. If the range gate is tracking the chaff when this separation is reached, the seduction will be successful. If the ship is being tracked, the seduction will be unsuccessful. It is this one-way transition period that makes a leading edge tracker particularly susceptible to seduction by chaff for some geometries.

If more than one chaff cloud is present simultaneously, the same analysis can be applied to the motion of the range gate from one chaff cloud to another. Superimposed chaff clouds act like a single chaff cloud that is more dense, while closely spaced chaff clouds act like a single chaff cloud with a larger range extent. Chaff clouds with a sufficient range separation operate independently, and only the one closest to the ship is capable of causing a seduction.

2.3 SAM System

Like an ASCM, the SAM that we model uses reflected microwave radiation for guidance. Instead of placing the radar illuminator in the SAM, however, a large shipboard microwave illuminator, known as a "SAM director," is used to illuminate the relatively small ASCM. The SAM is designed to fly a standard trajectory from launch until guidance information is received. This trajectory prevents the use of the SAM system in the inner zone. Although the minimum range at which the SAM system can be used is not necessarily the same as the boundary between the inner and middle zones we defined for the chaff, the difference is slight, and we consider them to be the same in order to avoid introducing additional notation.

The surveillance radar equipment installed on modern warships is able to detect an ASCM that is above the horizon when tactical considerations allow its use. Against a sea-skimming ASCM, surveillance radar equipment has an initial detection range of approximately 20 km. Midcourse guidance can be provided by a separate command channel or by homing on the reflected signal energy from the SAM illuminator. Regardless of which technique is used for midcourse guidance, the ASCM must be illuminated for several seconds immediately before the SAM reaches it to facilitate terminal guidance.

We have chosen to model a SAM system that uses reflected signal energy for midcourse guidance, because the requirement for continuous illumination increases the potential for harmful interference from the SAM system. SAM director illumination is also limited by the horizon to about 20 km against a sea-skimming ASCM. Because the ASCM must be illuminated for SAM guidance, it is not sensible to launch a SAM until the ASCM is detected. Once an ASCM is detected, however, a SAM can be launched immediately. Because the SAM director and the SAM do not share a common time base, a SAM using reflected signal energy for midcourse guidance has no way to determine the distance to the ASCM. Therefore, each SAM guides towards all ASCMs in the middle zone on the bearing illuminated by the SAM director. We assume here, for the sake of simplicity, that each SAM tracks towards the closest ASCM.

Because SAMs employ a proximity fuse, the SAM is destroyed when it reaches the closest ASCM, regardless of the fate of the ASCM. The defender on the ship can rapidly determine whether an ASCM has been destroyed by observing the reflected SAM director signal.

2.4 Chaff Effectiveness Model

The SAM system can significantly influence the probability of a successful seduction. This results from domination of the signal returned from the chaff by the signal returned from the ship when the SAM director is oriented towards the ASCM. The effect depends on the relative strength of the signal returns from the ship itself and from the chaff, the design of the SAM director antenna, and the design of the ASCM receiver.

Because we wish to study the effect of this interaction rather than its cause, we model the interaction by introducing an overwhelmingly dominant point scatterer that is present only during SAM director illumination. We have chosen a point scatterer with such a large radar cross section that even during the deepest fade it will dominate the signal return from the chaff.

Figure 5 shows the effect of introducing the SAM director into our analysis of range gate behavior. In that figure we have separately depicted the matched filter output associated with the SAM director by using dashed lines. As before, when the chaff and the ship are superimposed in range, the range gate can move easily between positions based on a dominant return from the chaff, the ship, or the SAM director. Eventually separation A is reached, beyond which the leading edge bias of the tracker precludes chaff-to-ship transitions when the SAM director is not illuminating the ASCM. Ship-to-chaff transitions (and SAM director-to-chaff transitions) remain possible, however, and SAM director illumination could still result in a chaff-to-SAM director transition. Subsequently, separation B is reached, at which the matched filter output from the SAM director signal is outside the range gate whenever the range gate's position is based on the chaff. Beyond separation B chaff-to-SAM director transitions become impossible, although the leading-edge bias of the tracker

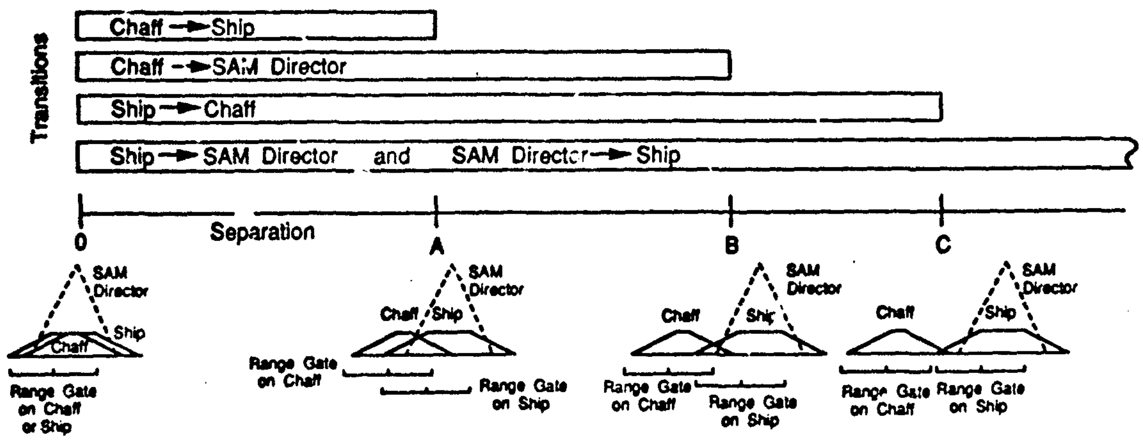

Fig. 5 — Range gate behvior with a SAM director

still permits ship-to-chaff (and SAM director-to-chaff) transitions. Finally, separation C is reached. Separation C is sufficient to prevent any transitions except the slight change in position associated with a ship-to-SAM director transition or a SAM director-to-ship transition.

To achieve a successful seduction, the range gate must be tracking the chaff when this final separation is reached. The probability of this occurrence depends on the signal returns from the chaff, the ship, and the SAM director during the seduction attempt. We describe in detail the case in which the range gate is initially tracking the ship in order to develop an expression for the probability of a successful seduction, given that no previous seduction attempt was successful.

As Fig. 3 shows, at each instant in time the range gate is captured by the object within it that is producing the strongest weighted output from the matched filter. When the SAM director is not illuminating the ASCM, the ship's signal return usually dominates that of the chaff, even when the chaff signal is weighted more heavily. Thus, when both objects are in the range gate, the range gate is normally tracking the ship. Occasional fades by the ship reverse this predominance, however, and lead to a temporary capture of the range gate by the chaff.

At relatively large separations, a range gate capture by the chaff places most of the ship outside the range gate. When this occurs, recapture of the range gate by the ship is precluded as long as the SAM director remains quiescent. If the range gate still includes the position of the SAM director, however, the extremely strong signal from the SAM director captures the range gate whenever the SAM director illuminates the ASCM. Once the SAM director moves out of the range gate, SAM director illumination no longer affects the range gate's position. So, if the range gate is tracking the chaff when the separation increases to the point where the SAM director is outside the range gate, a successful seduction is assured.

Reference to Fig. 5 allows us to construct a seduction model based on this behavior. We have assumed that the range gate is initially placed over the ship by the platform that launches the ASCM. To ensure that the chaff cloud is initially in the range gate, we select launch parameters that ensure it to bloom at the same distance from the ASCM as the ship. As in the previous case, the range gate will then alternate between positions based on the ship and the chaff, usually in a position based on the ship, until separation A in Fig. 5 is reached. SAM director illumination during this period simply serves to reduce the time spent tracking the chaff because during illumination the position of the range gate will be based on the SAM director signal.

Between separations A and B, SAM director illumination still results in capture of the range gate by the SAM director. When the illumination terminates, the range gate shifts to the ship, and capture of the range gate by the chaff becomes possible. Between separations B and C, capture of the range gate by the chaff remains possible, but the SAM director is no longer able to recapture it. In this region, if the range gate is tracking the ship, it moves slightly to track the SAM director. If, however, it is tracking the chaff, it remains on the chaff. So SAM director illumination during this period simply serves to lock the range gate in its present position.

Summarizing, if the chaff captures the range gate after the last SAM director illumination between separations A and B, a successful seduction is assured. The probability of such a capture by the chaff depends upon the pattern of illumination between the time of that last illumination and the time separation C is reached. A complete description of an illumination pattern requires specifying whether the SAM director is illuminating the ASCM at each time step in this period. Fortunately, the minimum duration of a SAM flight restricts the set of feasible illumination patterns to those that contain a small number of contiguous periods of illumination. The maximum number of separate no illumination periods depends on the length of the period between separations B and C, the minimum range of the SAM system, and the speed of the SAM.

Further simplification results from a time invariance that is a consequence of random fading. First consider the class of illumination patterns with a single no illumination period of fixed duration. Since we can not readily control when the ASCM will observe a fade by the ship, we make an a priori estimate for the probability that a fade by the ship will occur during the no illumination period that is sufficiently deep to cause a range gate transition. This probability estimate depends upon a specific set of conditions that includes sea state, ship heading, and ship speed. The fades are a direct result of ship motion, which is a narrow band random process. For simplicity, we assume here that for any time chosen at random, the next fade is equally likely to occur at any time within an interval roughly corresponding to a fundamental period of that narrowband process. Since we assume the phase of the fading patterns to be uniformly distributed, we hypothesize that the probability will be the same regardless of when the no illumination period begins. We call the duration of this single quiescent period D. At the beginning of the period the ASCM tracks the ship, having just shifted there from the SAM director. As D increases, the cumulative probability of a sufficiently deep fade by the ship increases as well.

Once separation C is reached, the outcome of the seduction attempt is completely determined. We define the time at which separation C is reached as our reference time t_R and define the $D(t)$ to be the unilluminated time that has been accumulated by time t. The probability of seduction is thus a function of the duration of the no illumination period at the reference time

$$P_S : D(t_R) \mapsto \Pr\{\text{Successful seduction at time } t_R;\, D(t_R)\}. \tag{1}$$

We could construct a similar function for more complex illumination patterns as well. Doing so would improve the fidelity of our model by identifying the effect of temporal correlations within the possible fading patterns. We believe, however, that we can bound the performance of the chaff with Eq. (1). Because the probability is computed by averaging over every phase of every possible fading pattern, the P_S function provides an upper bound on the seduction probability, when it is applied to the total no illumination time and a lower bound on the seduction probability when applied to the duration of the longest no illumination period. Since either bound reflects the interaction between chaff and SAM employment and the argument for the upper bound is easier to compute, we use the total no illumination time as the domain of $P_S(\cdot)$ instead of developing a more complex chaff effectiveness model.

The computation of D is then quite straightforward. If there is SAM director illumination between separations A and B, we begin counting D from zero at the end of the last such illumination. When there is no such illumination, we can begin counting D from zero when separation A is reached without significantly changing our analysis, because the ASCM is very likely to be tracking the ship at that time. If there is SAM director illumination between separations B and C, we stop incrementing D when it begins and resume counting when it ends.

2.5 Measurement of Chaff Effectiveness

If $D(t_R) = 0$ the continuous illumination prevents a successful seduction ($P_S = 0$). To quantify the remainder of the dependence of P_S on D we simulated a series of experiments by using the CRUISE Missiles testbed. Figure 6 shows the geometry we used. This geometry results in sufficient range and cross-range separation between the chaff and the ship to allow a seduction, and it exploits the leading edge bias of the tracker by assigning the ship a velocity component away from the ASCM. No wind or current was applied, and unaccelerated ship motion was assumed. Because the maximum range of the SAM system against low altitude ASCMs depends on the elevation of the SAM director, the SAM director is normally placed high on the superstructure near the middle of the ship. For this reason we placed the SAM director 35 m directly above the ship's center of gravity.

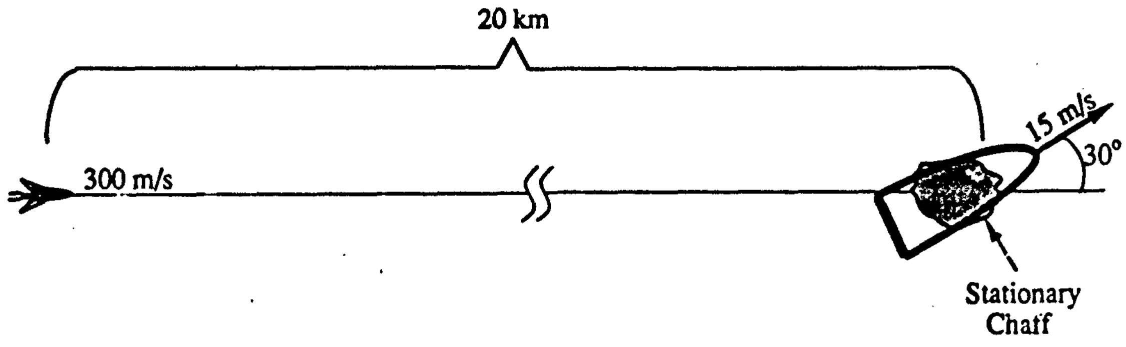

Fig. 6 — Monte Carlo simulation initial conditions

The ordinate of the graph in Fig. 7 shows the empirical probability of seduction that resulted from continuous illumination from the beginning of the run until the separation plotted on the abscissa was reached and then no further illumination until the ASCM reached the inner zone. The measure of separation that we have chosen is the range separation between the ship's center of gravity and the geometric center of the chaff along the ASCM-to-chaff line of sight. We calculated this empirical probability by conducting a set of simulation runs and observing whether the ASCM was tracking the ship or the chaff at the end of each run. In each trial we independently selected a pseudorandom seed for the ship and sea motion components of the model. This results in trial outcomes that are independent and identically distributed. Under this condition, the empirical probability approaches the parameter of the underlying binomial distribution as the number of runs increases. A sufficient number of runs were conducted to achieve a 0.9 confidence that the true seduction probability lies within a ± 0.1 confidence interval of the plotted value. Data points are plotted at approximately 25 m intervals.

Examination of Fig. 7 reveals that continuous illumination past 375 m of range separation always prevents a successful seduction. Therefore, 375 m in this geometry corresponds to separation C in Fig. 5. The time without illumination (D) before separation C is reached is plotted on the top of the graph. The data show a nearly linear increase in seduction probability with an increase in D. The limiting probability of seduction (in this case 1.0) depends on the relation between the signal from the chaff and the signal from of the ship at the deepest point of a fade by the ship for this geometry. We call this limiting value P_{max}. The time without illumination that is required to achieve this limiting seduction probability depends on the frequency with which fades occur, which is determined by the ship motion and the sea motion in the simulation. We call the smallest value of $D(t_R)$ that maximizes the probability of a successful seduction D_{max}.

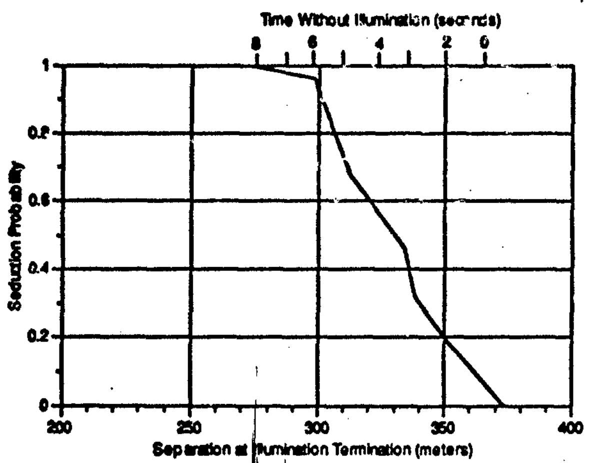

Fig. 7 -- Typical seduction probability function

Separation B in Fig. 5 can be found by determining the position of the range gate when it is tracking the chaff and then by computing the minimum range separation necessary to place all the energy from the SAM director cutside of the range gate. In our low-resolution ASCM seeker we use a 2 μs range gate because we have a 1 μs pulse width and a matched filter. When the leading edge tracker is tracking the chaff, the trailing edge of the range gate is located approximately 0.63 μs beyond the geometric center of the chaff cloud in signal space. The geometric center of the chaff cloud in signal space corresponds to the center of the chaff's flat spot in Fig. 5. When coupled with the 1 μs matched filter, a 1.63 μs round-trip transit time difference between the geometric center of the chaff and the location of the SAM director (the peak of the SAM director signal in Fig. 5) is sufficient to prevent the signal returned by the SAM director from influencing the tracker's behavior. For simplicity we have placed the SAM director directly over the ship's center of gravity. Thus, separation B is 245 m for this geometry. Note that we have implicitly assumed that the bias in the range gate is sufficient to prevent the signals reflected by closer parts of the ship from significantly affecting the leading edge tracker's behavior in this case.

Determination of separation A in Fig. 5 requires an additional experiment. We began by creating a single no illumination period of duration D_{max} that ends when separation C is reached. We then shifted the no illumination period earlier in time while maintaining a duration of D_{max} until the period barely lasted past separation B. If a reduction in seduction probability had been noted, the smallest separation that avoids the reduction would be separation A. However, no such reduction occurred. From this we conclude that separation A is less than 155 m (the smallest separation attempted) for this geometry. In that case the precise value of separation A is insignificant, because a lack of illumination between the actual value of separation A and separation B would have the same effect as a lack of illumination between a separation of 155 m and separation B. Therefore, we can consider separation A to be 155 m. Although this choice could result in a failure to count D between the actual value of separation A and a separation of 155 m, the resulting seduction probability would be unaffected because either the seduction probability would already be maximized or an intervening illumination would reset D.

We found similar seduction probability functions for other initial geometries as well. Variations in ship velocity produced the expected compression or expansion of the time axis. Slight shifts in separations A and C were also observed. This occurs because the tracker is affected by the range extent of the ship along the ASCM-to-ship line of sight, which varies with the ship's heading. Some ship headings also result in larger or smaller signal returns from the ship, which changes the effectiveness of the chaff. When present, this effect results in compression or expansion along the seduction probability axis. The value of P_{max} is easily found by conducting a single set of runs with no illumination. Changing the initial position of the chaff cloud slightly did not affect the outcome of our experiments. In particular, initial range separations between 0 and 270 m resulted in truncated versions of Fig. 7, and similar initial cross-range separations had no significant effect. Varying the position of the ASCM in the middle zone had no significant effect, because the period between multipath fades is nearly constant, while the ASCM is in the middle zone.

2.6 SAM System Effectiveness Model

The position of the chaff cloud can influence the performance of the SAM system. If the chaff cloud is placed directly between the ship and the ASCM at a low altitude (near the end of its life), it could prevent the SAM direct illumination from reaching the ASCM. This would result in loss of SAM guidance and require destruction of the SAM before it reaches the ASCM. This situation can be avoided by choosing a chaff bloom position that places it astern of the ship-to-ASCM line of sight before its altitude decays enough to cause a problem. Since that tactic is compatible with the requirements for optimal chaff effectiveness, we shall adopt it and consider SAM system effectiveness in isolation.

Because our focus is on the SAM/chaff interaction, we have chosen the simplest possible model for SAM system effectiveness. We model the effect of a SAM intercept with a constant P_K that represents the probability that a SAM intercepting an ASCM in the middle zone will destroy it. The parameter P_K can be chosen based on simulation results or operational experience. If an ASCM is intercepted but not destroyed, it can later be intercepted again by another SAM. By choosing P_K to be constant, we are treating each intercept as an independent event.

3 ENGAGEMENT MODEL

Although the individual effectiveness models are adequate to determine the distribution on the outcome of a single event, the integration of these into a unified whole remains to be done. We are interested in the effectiveness of several applications of the SAM and chaff systems in defending against an attack by multiple ASCMs. We restrict our analysis to the case in which all the ASCMs arrive from the same direction, and each travels with approximately the same velocity, because by doing so we simplify the engagement model while preserving the interaction we wish to study. We chose the arrival angle shown in Fig. 6, because it accommodates the requirements of our tactics. Since Fig. 7 indicates that the seduction probability is closely approximated by a negative-going ramp function, we have used that approximation to construct an idealized P_S function defined as

$$P_S = P_{max} \cdot \frac{D_{max} - \min\{D, D_{max}\}}{D_{max}}. \tag{2}$$

Inspection of Fig. 7 suggests that a reasonable value for D_{max} is 8. The interaction we wish to study is only present when P_{max} assumes a moderate value. If high assurance of a successful seduction were possible, SAM employment is not necessary. On the other hand, a low value for P_{max} keeps seduction from being worth considering. Thus, the interaction we wish to study is only significant when P_{max} assumes a moderate value. For this reason we have chosen to assign P_{max} a value of 0.5. Similar considerations dictate the choice of a moderate value for P_K. Therefore, we have somewhat arbitrarily assigned $P_K = 0.3$.

In developing a model for this multiple ASCM problem, which we will call the "engagement model," we must examine the effect that a single application of each defensive system has on different ASCMs. Because it is extremely unlikely that two ASCMs would be so close to each other that both could be destroyed by the same SAM, we will ignore that case. Hence, each SAM can effect at most one ASCM. A single chaff cloud could, however, affect every ASCM in the middle zone. Each ASCM arriving from a given direction will observe the same range separation between the chaff and the ship at the same time. Since we have restricted our attention to the case in which all ASCMs arrive from the same direction, every ASCM in the middle zone will experience each seduction attempt simultaneously.

The fading on which seduction depends is caused by multipath propagation and by changes in the ship's orientation. Multipath fading occurs when the length of two raypaths differ by a half-wavelength (modulo the wavelength). The dominant multipath interference occurs between the direct raypath and the raypath reflected once off the sea surface. ASCMs with the same path length difference (modulo the wavelength) will observe synchronized fading, while for other ASCMs the observed multipath fades will occur at different times. Fading due to changes in the ship's orientation occurs when many of the normally dominant scatterers are viewed from an orientation in which their reflectivity is low. All ASCMs arriving from the same direction should observe synchronized orientation-based fading. Although it should be possible to construct an accurate model of the relationship between the fading observed by multiple ASCMs, we have chosen to treat simultaneous seduction attempts as mutually independent events. This choice allows us to specify state variable transition probabilities individually rather than in all possible combinations, thereby reducing the complexity of the model.

The high speed of each ASCM and the limited number of ASCMs that an adversary could reasonably use naturally leads to a finite time horizon formulation. To facilitate computational solution, we have chosen a discrete time specification for our engagement model. To simplify the formal specification of the model we will assume that all moving objects travel at a constant velocity. This assumption allows us to express distances as time periods, eliminating unnecessary unit conversions.

3.1 States and Controls

Table 1 shows the state variables for the engagement model. We use the index i to distinguish between similar state variables that refer to different ASCMs and allow i to range from 1 to m, where m represents the maximum number of ASCMs that may arrive. Each state variable is discussed in detail below. We refer to the entire collection of state variables at time t as the "state" at time t, X_t. Note that while we use a subscript i on a state variable to indicate the associated ASCM, we use the subscript t on the state (and later on the control function) to represent time. We have included enough information in the state to construct a controlled one-step markov model, which is required for one of the optimization techniques that we will consider.

Table 1: State variables

A_i	Time until ASCM i reaches the ship (s)
C	Time until chaff enters seduction region (s)
D	Time since SAM director status change (s)
N	Number of remaining SAMs
S	Time until SAM intercepts closest ASCM (s)
T_i	Whether ASCM i is tracking the ship (True or False)

We sample state transitions once each Δ seconds, with the first transition beginning at time 0 and the last at time $\tau - \Delta$. Because the next state may depend on the outcome of one or more random events (ASCM detection, SAM intercept, or seduction attempt), the transition from the present state to the next state will, in general, be stochastic. Figure 8 specifies the allowed transitions for each state variable. Each arc is labeled with the condition under which that arc may be taken. When the transition is not deterministic, the probability the transition occurs is separated from the condition by a comma.

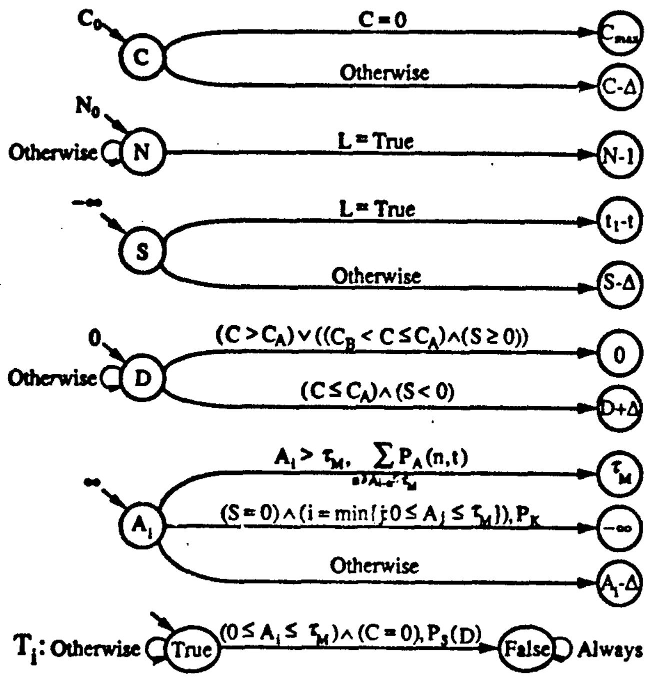

Fig. 8 — State variable transition diagrams

We will restrict our attention to ASCM arrival distributions that depend only on time and the number of ASCMs that have already been detected. We will use the information about the number of previously detected ASCMs to limit the attacker to m ASCMs. This is a relatively simple formulation that reflects both the attacker's lack of detailed knowledge of the defender's state and the defender's uncertainty about the attacker's strategy while producing a controlled one-step markov model. Therefore, we take as given:

$$P_A(n,t,a)$$
$$= \Pr\{n \text{ ASCMs will arrive at time } t \text{ given that } a \text{ ASCMs have already arrived}\}. \quad (3)$$

Because a relatively large range separation is required between chaff clouds if separate seduction attempts are to occur, relatively few seduction opportunities can be created for each ASCM. We have chosen to repeatedly fire chaff rounds at the minimum effective interval to maximize the chance of a successful seduction. This decision reduces the complexity of the control policy we seek without eliminating our ability to observe the effects of the SAM/chaff interaction.

Accordingly, the ship has available one control $L(t)$. Setting $L(t)$ to True represents the launch of a SAM at time t. Setting $L(t)$ to False represents foregoing the opportunity for a launch at that time. $L(t)$ is constrained to be False if a SAM is already in flight or if the SAM inventory is exhausted. We make it possible to enforce an inventory constraint by defining N to be the number of SAMs remaining. Initially we set N to N_0, the number of SAMs that are available at the beginning of the engagement.

The periodic nature of chaff deployment is represented by a counter C that represents the time until the next chaff cloud will next be properly positioned for a seduction attempt. C initially has a value C_0, which represents the phase of the periodic chaff replacement. Subsequently, C counts down modulo $C_{max} + \Delta$.

We use A_i to represent time remaining before ASCM i reaches the inner zone. A_i is initially set to ∞, which represents a remaining time greater than τ_M. As long as ASCM i is outside the middle zone there is a chance that it will arrive at the outer boundary of that zone and be detected. There will be n arrivals at time t with probability $P_A(n,t)$. By convention we assume that ASCMs are numbered in the order of their arrival. Therefore, ASCM i arrives at time t if there are more arrivals at time t than there are lower numbered ASCMs that have not yet arrived. When ASCM i is detected, A_i is set to τ_M. A_i subsequently decrements until ASCM i reaches the inner zone or is destroyed by a SAM intercept. For ASCMs, which reach the inner zone, we allow the value of A_i to continue to decrement below zero and interpret negative values of A_i to represent ASCMs that are no longer in the middle zone. If ASCM i is the closest ASCM in the middle zone (i.e., i is the smallest j for which $0 \leq A_j \leq \tau_M$) and it is destroyed by a SAM intercept, we set A_i to $-\infty$, representing an ASCM that will never reach the inner zone (and thus never hit the ship).

Intercepts occur when the most recently launched SAM reaches the closest ASCM in the middle zone. We assume the SAM and ASCM both move with constant velocity and define V_S to be the ratio of the velocity of the SAM to the velocity of the ASCM. The distance between the SAM and the ASCM will decrement at $V_S + 1$ times the rate at which A_i is decrementing. At time t, $\min\{A_i(t) : 0 \leq A_i(t) \leq \tau_M\} + \tau_I$ more seconds would be required for the closest ASCM in the middle zone to reach the ship. If we divide these two quantities and adjust the result to be an integer multiple of Δ, then at the time the SAM is launched we can calculate the time at which the intercept will occur t_I as

$$t_I(t) = t + \left\lfloor \frac{\min_i\{A_i(t)\} + \tau_I}{V_S + 1} \cdot \frac{1}{\Delta} \right\rfloor \cdot \Delta. \tag{4}$$

The state variable S is used to carry this information forward from the SAM launch time to the time of the intercept. Initially S is set to $-\infty$, where we interpret negative values of S to represent a state with no airborne SAM. When a SAM is launched, S is reset to the calculated intercept time t_I. Subsequently, S decrements to zero, at which time an intercept is recognized. Since at most one SAM can be in flight at a time, a scalar value suffices to represent this information.

The illumination duration D remains zero until separation A is reached, which occurs when $C = C_A$. Subsequently, D is incremented by Δ whenever there is no airborne SAM and reset to zero when there is an airborne SAM. Once C reaches separation B, which occurs when $C = C_B$, D continues to increment when there is no airborne SAM but holds its value when a SAM is airborne.

The Boolean state variable T_i whether ASCM i is tracking the ship. Initially T_i is True $\forall i$ because all newly detected ASCMs are assumed to be tracking the ship. Seduction attempts occur when $C = 0$, and they can be effective against any ASCM in the middle zone (i.e., any ASCM i for which $0 \leq A_i \leq \tau_M$). At each seduction attempt T_i will become False if the seduction attempt is successful.

We have collected all of the parameters of the engagement model in Table 2 and indicated typical values for a simple instance of the model. While these parameters are often somewhat arbitrary, the entire set of parameters must be chosen in a consistent manner. For instance, τ_M must be some multiple of Δ so that it will be possible for A_i to reach zero.

Table 2: Engagement Model Parameters

Parameter	Value	Units	Description
Δ	1	seconds	Time quantum
τ	147	seconds	Final time
τ_M	48	seconds	ASCM flight time in the middle zone
τ_I	12	seconds	ASCM flight time in the inner zone
C_0	30	seconds	Time of first seduction attempt
C_A	15	seconds	Value of C for separation A
C_B	8	seconds	Value of C for separation B
C_{max}	19	seconds	Maximum time until the next seduction attempt
D_{max}	8	seconds	Maximum no-illumination duration
m	3	ASCMs	Maximum number of ASCMs which may arrive
N_0	5	SAMs	Initial SAM inventory
P_K	0.3		Probability that a SAM will kill an ASCM
P_{max}	0.5		Maximum seduction probability
V_S	3	ASCM Speed	Relative speed of a SAM

3.2 Observations

The value of most of the state variables can be observed or deduced without error at each time. Figure 8 identifies the initial value for each state variable using a small arrow. Since C has a known initial value and it evolves deterministically, $C(t)$ is known a priori for all t, regardless of the policy selected. The evolution of N is completely determined by the control $L(t)$. which becomes known as it is chosen at each time. So $N(t)$ becomes known by time t.

Although A_i evolves stochastically, its value can be observed at each time. Because the velocity of an ASCM is nearly constant, A_i is directly proportional to the distance between the ASCM and the ship. Thus, $A_i(t)$ can be observed at time t without significant error by using surveillance radar equipment. Since $\{A_i(t)\}_1^m$ is known in this way at time t and $L(t)$ is also known at time t, $S(t)$ can be calculated by using Eq. (4) at time t. And finally, once $S(t)$ and $C(t)$ are known, $D(t)$ can be calculated.

As long as no seduction has been attempted against ASCM i, T_i evolves deterministically and its value can be computed. Once a stochastic state transition occurs, however, this is no longer possible. Furthermore, it is impractical to observe T_i while ASCM i is in the middle zone. $T_i(t)$ again becomes known with certainty only after ASCM i enters the inner zone and reaches either the ship or the chaff cloud.

We will refer to the known state variables at time t as the observation at time t, O_t and define a function $O : X_t \mapsto O_t$. O_t consists of every state variable in X_t except $T_i(t)$ for those ASCMs that were in the middle zone during a prior seduction attempt and have not yet reached either the ship or the chaff.

3.3 Reward Function

We assume that a single hit by an ASCM is sufficient to disable the ship for the remainder of the engagement. For small ships such as frigates and destroyers this is a reasonable assumption. Therefore, we wish to find a policy that will maximize the probability that no ASCMs will hit the ship between time 0 and time τ.

By a policy we mean a set of functions indexed by time, the range of each being the control to be applied at that time. In general, we would like to base our control decisions on all of the useful information available at each time. If we were to choose X_t as the domain for the control function at time t we would obtain a one-step markov model. Furthermore, since X_t captures all of the relevant information about prior states and controls that is necessary to determine the next state, adding prior state and control information to the domain of the control function would not change the optimal value of the reward function. Unfortunately, it is sometimes impossible to observe part of X_t. When part of X_t cannot be observed, a control function that requires all of X_t as its domain would not be useful to a defender.

Another obvious choice for the domain of the control function at time t is O_t, the observation at time t. Since $O_t \subset X_t$, this choice would yield a one-step markov model using a control function that could be used directly by the defender. In this case, however, adding prior observations to the domain of the control function could potentially improve the optimal value of the reward function. In addition, our knowledge of system dynamics also makes it useful to include the prior controls we have applied in the domain of the control function. To quantify this dependence we define the information vector at time t to be

$$\begin{aligned} I_0 &= O_0 \\ I_t &= \{O_0, \ldots, O_t, L(0), \ldots, L(t-\Delta)\}, \ (t \in \{\Delta, \ldots, \tau - \Delta\}). \end{aligned}$$

The policy we seek is a set of functions $\pi = \{\mu_t : I_t \mapsto L(t)\}_0^{\tau-\Delta}$. We wish to find the policy π that maximizes the probability that the ship is not hit when that policy is employed. Therefore, we choose as our reward function

$$J_\pi \equiv \Pr\{\text{Ship not hit}; \pi\}$$

In this notation we have explicitly identified both the event ({ Ship not hit }) and the parameter (π) that determines the probability of that event. We shall use this notation extensively to call attention to the functional dependence of a distribution on certain parameters.

4 OPTIMAL SCHEDULING TECHNIQUES

We seek to find an optimal policy π^* for which

$$J_{\pi^*} = \max_\pi\{J_\pi\}.$$

We shall present two techniques for finding an optimal policy, exhaustive search and dynamic programming.

4.1 Exhaustive Search

Exhaustive search is a brute-force strategy that can be efficient when the number of alternatives that must be considered is small. In the exhaustive search strategy we first compute the value of J_π for every possible policy π and then choose the policy that maximizes that value. We begin by developing an algorithm for computing J_π. Starting with the definition of the reward function we apply the law of total probability:

$$\begin{aligned} J_\pi &= \Pr\{\text{Ship not hit}; \pi\} \\ &= 1 - \Pr\{\text{Ship hit}; \pi\} \\ &= 1 - \sum_{\{X_0, \ldots, X_T\}} \Pr\{\text{Ship hit} | X_0, \ldots, X_T; \pi\} \cdot \Pr\{X_0, \ldots, X_T; \pi\} \qquad (5) \end{aligned}$$

The engagement model described in Fig. 8 specifies $\Pr\{X_{t+\Delta}|X_t, L(t)\}$ and specifies a delta function for $\Pr\{X_0\}$. Since $L(t) = \mu_t(I_t)$, we can use the one-step markov structure of the model to compute the second probability in Eq. (5):

$$\Pr\{X_0, \ldots, X_T; \pi\} = \Pr\{X_0\} \cdot \prod_{t=0}^{T-\Delta} \Pr\{X_{t+\Delta}|X_t, \mu_t(I_t)\}. \tag{6}$$

Computation of the first probability in Eq. (5) is somewhat more complex. In our analysis above we determined that the ship will be hit if and only if at least one ASCM reaches the inner zone without being seduced. That occurs if and only if there exist an ASCM i and time t for which both $A_i(t) = 0$ (i.e., at time t ASCM i reaches the inner zone) and $T_i(t) = \text{True}$ (i.e., at time t ASCM i is tracking the ship). This formulation captures the effect of a successful seduction directly by using T_i but relies on A_i to keep ASCMs that are destroyed by SAMs from being considered. Using the indicator function $1_{\{Event\}}$ that has value 1 when the event occurs and 0 when it does not

$$\Pr\{\text{Ship hit}|X_0, \ldots, X_T; \pi\} = 1_{\{\exists(i,t)(A_i(t)=0)\wedge(T_i(t)=\text{True})\}} \tag{7}$$

Combining Eqs. (5), (6), and (7) and then eliminating zero terms from the summation we get

$$\begin{aligned} J_\pi &= 1 - \sum_{\{X_0,\ldots,X_T\}} 1_{\{\exists(i,t)(A_i(t)=0)\wedge(T_i(t)=\text{True})\}} \cdot \Pr\{X_0\} \cdot \prod_{t=0}^{T-\Delta} \Pr\{X_{t+\Delta}|X_t, \mu_t(I_t)\} \\ &= 1 - \sum_{\{X_0,\ldots,X_T\}\ni\exists(i,t)(A_i(t)=0)\wedge(T_i(t)=\text{True})} \Pr\{X_0\} \cdot \prod_{t=0}^{T-\Delta} \Pr\{X_{t+\Delta}|X_t, \mu_t(I_t)\}. \end{aligned} \tag{8}$$

Since we can repeatedly apply the definitions of μ_t, I_t, and O to find $\mu_t(I_t)$, given $\{X_0, \ldots, X_t\}$, this equation provides a somewhat cumbersome way to compute J_π for any policy π. By trying every policy, we will eventually find the one that maximizes J_π. Examination of Eq. (8) reveals that this computation will be most efficient when the time horizon is short and the number of possible state sequences is small. Since the computation is repeated for every policy π, practical application of this technique is only possible when there are few times when more than one alternative is available.

4.2 Incremental Computation of the Reward Function

The exhaustive search technique requires that every possible path through the state space be considered. Multiple control alternatives or stochastic state transitions will result in exponential growth in the number of paths. If we define computational complexity to be the greater of the asymptotic time or space requirements of an algorithm, the computational complexity of the exhaustive search technique grows exponentially as the number of time steps is increased.

The controlled one-step markov structure of the engagement model suggests that it may be useful to view this optimization problem as a multistage decision process and consider incremental computation of J_{π^*}. Incremental computation can result in greater efficiency by computing intermediate values for each unique subpath only once. Two approaches to incremental computation are possible.

Incremental computation forward in time is done by walking a decision tree to find the optimal policy and considering several branches at each step in which a stochastic state transition occurs. Only reachable states are considered when walking a decision tree. As the number of reachable states becomes large, however, a tree walk will consider many of these states more than once. This requires either repeating the computation each time a state is considered or storing the previous result. If recomputation is chosen, the computational complexity of a tree walk grows exponentially as the number of time steps is increased. When intermediate results are stored, a large intermediate storage area that grows as the number of time steps is increased will be required; however, computational complexity will grow linearly with the number of time steps.

Incremental computation backward in time is done by constructing a state lattice backward from each possible final state. The algorithm for incremental computation backward in time is known as dynamic programming. Because every state is considered only once at each time, the computational complexity of a dynamic programming solution grows linearly as the number of time steps is increased. Although dynamic programming requires intermediate storage for one value for each state, the required storage area does not change as the number of time steps is increased. For that reason, we will next apply the dynamic programming algorithm to this problem.

4.3 Dynamic Programming

In dynamic programming we seek to reduce the problem of selecting the optimal policy to a sequential selection of optimal control functions for each time. The dynamic programming algorithm works by associating with every state at time t a value that represents the expected reward that would be earned if we started in that state at time t and employed an optimal policy between time t and time τ. This set of optimum expected rewards can then be used to compute the optimum expected reward associated with any state at time $t - \Delta$, if the incremental effect of each possible control that could be applied from that state at time $t - \Delta$ is known.

Conventional dynamic programming is valid for a reward function formed as the expectation of a sum of partial rewards. Appendix A describes a similar algorithm for a reward function that is the expectation of a product of partial rewards. Appendix A identifies five properties that a model must possess before that dynamic programming algorithm can be applied:

(a) A finite set of states

(b) A control function with the state as its domain

(c) A controlled one-step markov model in which the distribution on the next state is determined by the present state and the present control.

(d) A set of nonnegative partial rewards functions, each of which has the state at one time as its domain.

(e) A reward function that is the expectation over the state sequence of a product over time of those partial rewards.

4.3.1 Computation of the Reward Function Using the Information Vector

Because we do not have perfect knowledge of the state at each time, we must use the information vector I_t as the domain of control function at time t. To satisfy property (b) we must treat I_t as if it were a state in our development of a dynamic programming solution. That choice satisfies property (a) because I_t is a finite union of finite sets.

Property (c) then requires that $I_{t+\Delta}$ be a known stochastic function (called the next state function) of I_t and $\mu_t(I_t)$. We will show the existence of a next state function for I_t by briefly sketching its derivation. We begin by applying the definition of I_t to the next state function from property (c) in Appendix A and then simplify the resulting expression, thus observing that

$$L(t) = \mu_t(I_t)$$

$$\begin{aligned}
&\Pr\{I_{t+\Delta}|I_t;\mu_t(I_t)\}\\
&= \Pr\{O_0,\ldots,O_t,O_{t+\Delta},L(0),\ldots,L(t-\Delta),L(t)|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);\mu_t(I_t)\}\\
&= \Pr\{O_{t+\Delta}|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}. \qquad (9)
\end{aligned}$$

Now recall that each $X_{t+\Delta}$ is associated with exactly one $O_{t+\Delta}$. So $\mathcal{O}$ partitions the state space. We can therefore express the value in Eq. (9) as a sum of probabilities, then apply the law of total probability and simplify the result by using the one-step markov property of the engagement model:

$$\begin{aligned}
&\Pr\{I_{t+\Delta}|I_t;\mu_t(I_t)\}\\
&= \sum_{X_{t+\Delta}\ni O_{t+\Delta}=\mathcal{O}(X_{t+\Delta})} \Pr\{X_{t+\Delta}|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}\\
&= \sum_{X_{t+\Delta}\ni O_{t+\Delta}=\mathcal{O}(X_{t+\Delta})}\ \sum_{\{X_0,\ldots,X_t\}\ni\{O_0,\ldots,O_t\}=\mathcal{O}(\{X_0,\ldots,X_t\})}\\
&\quad \Pr\{X_{t+\Delta}|X_0,\ldots,X_t,O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}\cdot\\
&\quad \Pr\{X_0,\ldots,X_t|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}\\
&= \sum_{X_{t+\Delta}\ni O_{t+\Delta}=\mathcal{O}(X_{t+\Delta})}\ \sum_{\{X_0,\ldots,X_t\}\ni\{O_0,\ldots,O_t\}=\mathcal{O}(\{X_0,\ldots,X_t\})}\\
&\quad \Pr\{X_{t+\Delta}|X_t;L(t)\}\cdot\Pr\{X_0,\ldots,X_t|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}\\
&= \sum_{\{X_0,\ldots,X_{t+\Delta}\}\ni\{O_0,\ldots,O_{t+\Delta}\}=\mathcal{O}(\{X_0,\ldots,X_{t+\Delta}\})}\\
&\quad \Pr\{X_{t+\Delta}|X_t;L(t)\}\cdot\Pr\{X_0,\ldots,X_t|O_0,\ldots,O_t,L(0),\ldots,L(t-\Delta);L(t)\}. \qquad (10)
\end{aligned}$$

The first probability in Eq. (10) is specified in Fig. 8 and the second can be found by first applying Bayes' theorem, then simplifying the result through the observation that in Eq. (10) $\{O_0,\ldots,O_T\} = \mathcal{O}(\{X_0,\ldots,X_T\})$, and finally by using the markov property to write the joint probability as a product of conditional probabilities

$$\begin{aligned}
&\Pr\{X_0,\ldots,X_t|O_0,\ldots,O_t;\pi\}\\
&= \frac{\Pr\{X_0,\ldots,X_t;\pi\}\cdot\Pr\{O_0,\ldots,O_t|X_0,\ldots,X_t;\pi\}}{\sum_{\{X_0,\ldots,X_t\}\ni\{O_0,\ldots,O_t\}=\mathcal{O}(\{X_0,\ldots,X_t\})}\Pr\{X_0,\ldots,X_t;\pi\}\cdot\Pr\{O_0,\ldots,O_t|X_0,\ldots,X_t;\pi\}}\\
&= \frac{\Pr\{X_0,\ldots,X_t;\pi\}}{\sum_{\{X_0,\ldots,X_t\}\ni\{O_0,\ldots,O_t\}=\mathcal{O}(\{X_0,\ldots,X_t\})}\Pr\{X_0,\ldots,X_t;\pi\}}\\
&= \frac{\Pr\{X_0\}\cdot\prod_{t'=0}^{t-\Delta}\Pr\{X_{t'+\Delta}|X_{t'};L(t')\}}{\sum_{\{X_0,\ldots,X_t\}\ni\{O_0,\ldots,O_t\}=\mathcal{O}(\{X_0,\ldots,X_t\})}\Pr\{X_0\}\cdot\prod_{t'=0}^{t-\Delta}\Pr\{X_{t'+\Delta}|X_{t'};L(t').\}}
\end{aligned}$$

So the next state function is well defined and property (c) is satisfied. Properties (d) and (e) require that the reward function be formed as the expectation over the state sequence of a product over time of nonnegative functions of the state. To put our reward function in this form, we start with the definition of the reward function and again apply the law of total probability followed by the definition of expectation to obtain:

$$\begin{aligned}
J_\pi &= \Pr\{\text{Ship not hit};\pi\}\\
&= 1-\Pr\{\text{Ship hit};\pi\}\\
&= 1-\sum_{\{O_0,\ldots,O_T\}}\Pr\{O_0,\ldots,O_T;\pi\}\cdot\Pr\{\text{Ship hit}|O_0,\ldots,O_T;\pi\}\\
&= 1-\mathop{\mathrm{E}}_{\{O_0,\ldots,O_T\}}[\Pr\{\text{Ship hit}|O_0,\ldots,O_T;\pi\};\pi]. \qquad (11)
\end{aligned}$$

In our notation for expectation, we place the parameter on which the distribution depends inside the brackets after a semicolon. Now we can derive an analytic expression for the probability that the ship is hit by recalling the condition we introduced earlier

$$\begin{aligned}\Pr\{\text{Ship hit}|O_0,\ldots,O_T;\pi\} &= \Pr\{\exists(i,t)(A_i(t)=0)\wedge(T_i(t)=\text{True})|O_0,\ldots,O_T;\pi\}\\ &= 1-\Pr\{\forall((i,t)\ni A_i(t)=0)(T_i(t)=\text{False})|O_0,\ldots,O_T;\pi\}\\ &= 1-\prod_{(i,t)\ni A_i(t)=0}\Pr\{T_i(t)=\text{False}|O_0,\ldots,O_T;\pi\}\end{aligned}$$

The last step is based on the mutual independence of $T_i(t)\forall i$ at every time t, given the same control sequence. This independence is a consequence of the known value of $T_i(0)$ and the (assumed) mutual independence of simultaneous seduction attempts. Substituting our result into Eq. (11) we get

$$J_\pi = 1-\underset{\{O_0,\ldots,O_T\}}{\mathrm{E}}[1-\prod_{(i,t)\ni A_i(t)=0}\Pr\{T_i(t)=\text{False}|O_0,\ldots,O_T;\pi\};\pi]$$

$$J_\pi = \underset{\{O_0,\ldots,O_T\}}{\mathrm{E}}[\prod_{(i,t)\ni A_i(t)=0}\Pr\{T_i(t)=\text{False}|O_0,\ldots,O_T;\pi\};\pi]. \qquad (12)$$

Since $T_i(t)$ is not known, we must compute its distribution based on the available information.

$$\begin{aligned}&\Pr\{T_i(t)=\text{False}|O_0,\ldots,O_T;\pi\}\\ &= 1-\Pr\{T_i(t)=\text{True}|O_0,\ldots,O_T;\pi\}\\ &= 1-\Pr\{\text{No successful seduction attempt against ASCM } i \text{ by time } t|O_0,\ldots,O_T;\pi\}.\end{aligned}$$

The relevant seduction attempts for each ASCM are those that occur while it is in the middle zone. From the state transition diagram for T_i in Fig. 8, we see that seduction attempts only occur when $0\le A_i\le\tau_M$ and $C=0$. Using this observation, the markov structure of T_i, and the definition of P_S we proceed as follows

$$\begin{aligned}&\Pr\{T_i(t)=\text{False}|O_0,\ldots,O_T;\pi\}\\ &= 1-\Pr\{\forall(t'\le t)((C(t')=0)\wedge(0\le A_i(t')\le\tau_M))\\ &\qquad\Rightarrow(\text{Failed to seduce ASCM } i \text{ at } t')|O_0,\ldots,O_T;\pi\}\\ &= 1-\prod_{(t'\le t)\ni(C(t')=0)\wedge(0\le A_i(t')\le\tau_M)}\Pr\{(\text{Failed to seduce ASCM } i \text{ at } t')|O_0,\ldots,O_T;\pi\}\\ &= 1-\prod_{(t'\le t)\ni(C(t')=0)\wedge(0\le A_i(t')\le\tau_M)}(1-\Pr\{(\text{Seduced ASCM } i \text{ at } t')|O_0,\ldots,O_T;\pi\})\\ &= 1-\prod_{(t'\le t)\ni(C(t')=0)\wedge(0\le A_i(t')\le\tau_M)}(1-P_S(D(t'))).\end{aligned} \qquad (13)$$

Combining Eqs. (12) and (13) and then separating the outer product into products over t and i we get:

$$\begin{aligned}J_\pi &= \underset{\{O_0,\ldots,O_T\}}{\mathrm{E}}[\prod_{(i,t)\ni A_i(t)=0}(1-\prod_{(t'\le t)\ni(C(t')=0)\wedge(0\le A_i(t')\le\tau_M)}(1-P_S(D(t'))));\pi]\\ &= \underset{\{O_0,\ldots,O_T\}}{\mathrm{E}}[\prod_{t=0}^{T}\prod_{i\ni A_i(t)=0}(1-\prod_{(t'\le t)\ni(C(t')=0)\wedge(0\le A_i(t')\le\tau_M)}(1-P_S(D(t'))));\pi].\end{aligned} \qquad (14)$$

We now observe that the policy π establishes a one-to-one correspondence between the sequence of observations $\{O_0,\ldots,O_T\}$ and the sequence of information vectors $\{I_0,\ldots,I_T\}$. Therefore, we can rewrite Eq. (14) to get an expression in the form required to apply dynamic programming:

$$J_\pi = \underset{\{I_0,\ldots,I_T\}}{\mathrm{E}}[\prod_{t=0}^{T} \prod_{i \ni A_i(t)=0} (1 - \prod_{(t' \leq t) \ni (C(t')=0) \wedge (0 \leq A_i(t') \leq \tau_M)} (1 - P_S(D(t')))); \pi]$$

$$= \underset{\{I_0,\ldots,I_T\}}{\mathrm{E}}[\prod_{t=0}^{T} g_t(I_t)] \quad (15)$$

where we define,

$$g_t(I_t) = \prod_{i \ni A_i(t)=0} (1 - \prod_{(t' \leq t) \ni (C(t')=0) \wedge (0 \leq A_i(t') \leq \tau_M)} (1 - P_S(D(t')))) \; \forall t \in \{0, \ldots, \tau\}. \quad (16)$$

The form of Eq. (15) satisfies property (e). Equation (16) satisfies property (d) because $g_t(I_t)$ can be computed without reference to states for times later than t. In particular, $g_t(I_t)$ can be computed solely by reference to $C(t')$, $A_i(t')$, and $D(t')$ for $t' \leq t$. Furthermore, $g_t(I_t)$ is nonnegative because it is the probability that every ASCM leaving the middle zone at time t has been seduced, and probabilities lie in the interval $[0, 1]$. We have therefore demonstrated all five of the properties required to apply the dynamic programming algorithm in Appendix A. As a result we can compute J_{π^*} as,

$$V_T(I_T) = g_T(I_T) \quad (17)$$

$$V_t(I_t) = \max_{\mu_t(I_t)}\{ \underset{\{I_{t+\Delta}\}}{\mathrm{E}} [V_{t+\Delta}(I_{t+\Delta}) \cdot g_t(I_t); \mu_t(I_t)] \; (t \in \{0, \ldots, \tau - \Delta\}) \quad (18)$$

$$J_{\pi^*} = \underset{\{I_0\}}{\mathrm{E}}[V_0(I_0)]$$

4.3.2 Constraining State Space Growth

As t increases, the cardinality of I_t grows rapidly. This means that the maximization in the dynamic programming algorithm must be performed over functions with rapidly expanding domains. However, very little of the information in I_t is actually used each time to compute the reward function. Using this insight, we will construct a related optimization problem with a state of fixed cardinality that can be used to find the optimal policy for our original problem.

Examination of Eq. (16) reveals that the partial reward at time t depends on the value of D for every time an ASCM leaving the inner zone at time t might have been seduced. By including this information in our new state, we can construct a one-step markov model without recourse to an information vector. Equation (13) can be used to interpret the inside product in Eq. (16) as the probability that T_i is True at time t. We define a function $\mathcal{T}$ to compute that inside product as follows

$$\mathcal{T}(t, I_t) = \prod_{(t' \leq t) \ni (C(t')=0) \wedge (0 \leq A_i(t') \leq \tau_M)} (1 - P_S(D(t'))) \quad (19)$$

For convenience, we will also define a set of symbols $\{\hat{T}_1, \ldots, \hat{T}_m\}$ as

$$\hat{T}_i(t) = \mathcal{T}(t, I_t). \quad (20)$$

Equation (14) then becomes

$$J_\pi = \underset{\{O_0,\ldots,O_T\}}{\mathrm{E}}[\prod_{t=0}^{T} \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t)); \pi]. \quad (21)$$

It is possible to compute $\hat{T}_i(t)$ from $\hat{T}_i(t - \Delta), C(t - \Delta)$, $A_i(t - \Delta)$, and $D(t - \Delta)$ as shown in Fig. 9. To see why, recall that $\hat{T}_i(t)$ represents the probability no seduction attempt for ASCM i

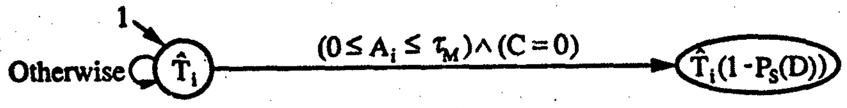

Fig. 9 — Transition diagram for $\hat{T}_i$

has been successful by time t. Before ASCM i is detected, that probability is 1. When a seduction attempt occurs with ASCM i in the middle zone, $\hat{T}_i(t)$ is multiplied by the probability that the seduction attempt is unsuccessful. Figure 1 is a transition diagram for $\hat{T}_i$.

We are now ready to formally define our new model. We begin with the counterparts to I_t, $\mu_t(\cdot)$, and π

$$\begin{aligned} \hat{O}_t &= O_t \cup \bigcup_i \hat{T}_i(t) \\ \hat{\mu}_t &: \hat{O}_t \mapsto L(t),\ (t \in \{0, \ldots, \tau - \Delta\}) \\ \hat{\pi} &= \{\hat{\mu}_0(\cdot), \ldots, \hat{\mu}_{T-\Delta}(\cdot)\}. \end{aligned}$$

We can replace the σ-algebra defined by O_t in Eq. (21) with the finer σ-algebra defined by $\hat{O}_t$ to get an expression for the reward function

$$J_\pi = \mathop{\mathrm{E}}_{\{\hat{O}_0,\ldots,\hat{O}_T\}} [\prod_{t=0}^{T} \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t)); \pi]. \tag{22}$$

Before we can apply the dynamic programming algorithm we must replace π with $\hat{\pi}$ to satisfy property (b). Accordingly, we define

$$\hat{J}_{\hat{\pi}} = \mathop{\mathrm{E}}_{\{\hat{O}_0,\ldots,\hat{O}_T\}} [\prod_{t=0}^{T} \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t)); \hat{\pi}] \tag{23}$$

and seek to find

$$\hat{J}_{\hat{\pi}^*} = \max_{\hat{\pi}}\{\hat{J}_{\hat{\pi}}\}. \tag{24}$$

We claim that maximizing $\hat{J}_{\hat{\pi}}$ is equivalent to maximizing J_π in the sense that for every optimal policy π^* there exists an optimal policy $\hat{\pi}^*$ such that $\hat{J}_{\hat{\pi}^*} = J_{\pi^*}$. We prove this claim in Appendix B by showing that $\mu_t(I_t)$ can be chosen to be the same for every I_t that corresponds to the same $\hat{O}_t$ without changing the optimal reward. This means that $\hat{\mu}_t(\hat{O}_t) = \mu_t(\mathcal{T}(t, I_t))$ is an optimal control function for the original optimization problem.

Now we are ready to show that the same five properties hold for the new optimization problem. We will treat $\hat{O}_t$ as the state. Property A(a) is satisfied because the periodic launch of chaff limits us to a small number of seduction attempts for each ASCM. Since D takes values from a finite set and the number of products of functions of D we are performing is finite, $\hat{T}_i$ takes values from a finite set. Therefore, $\hat{O}_t$ is a finite set.

Property A(b) is satisfied because the domain of the control function is the state. Furthermore, the control policy will be useful to a defender. Since $\hat{O}_0$ is known, and each $\hat{T}_i(t)$ is computed incrementally based only on information contained in $\hat{O}_{t-\Delta}$, $\hat{T}_i(t)$ becomes known by time t. Because O_t is also known by time t, we conclude that $\hat{O}_t$ becomes known by time t. So the domain of the control function is known in real time.

The transition diagram in Fig. 9 is deterministic, and Fig. 8 can be used to determine the next state function for O_t without knowledge of the outcome of stochastic transitions of T_i. So property A(c) is satisfied as well.

To apply the dynamic programming algorithm we define

$$\hat{g}_t(\hat{O}_t) = \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t)) \; \forall t \in \{0, \ldots, \tau\}. \tag{25}$$

This allows us to rewrite the reward function in the form required by property (e):

$$\hat{J}_{\hat{\pi}} = \underset{\{\hat{O}_0,\ldots,\hat{O}_T\}}{\mathrm{E}} [\prod_{t=0}^{T} g_t(O_t); \hat{\pi}]$$

This quantity can be computed based only on $\hat{O}_t$, and it is nonnegative because it is a probability. So properties (d) and (e) are satisfied. Since all five properties in Appendix A are satisfied we may write:

$$\hat{V}_T(\hat{O}_T) = \hat{g}_T(\hat{O}_T) \tag{26}$$

$$\hat{V}_t(\hat{O}_t) = \max_{\{\hat{\mu}_t(\hat{O}_t)\}} \{ \underset{\{\hat{O}_{t+\Delta}\}}{\mathrm{E}} [\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) \cdot \hat{g}_t(\hat{O}_t); \hat{\mu}_t(\hat{O}_t)]\}, \; (t \in \{0, \ldots, \tau - \Delta\}) \tag{27}$$

$$\hat{J}_{\hat{\pi}^*} = \underset{\{\hat{O}_0\}}{\mathrm{E}} [\hat{V}_0(\hat{O}_0)]. \tag{28}$$

5 DYNAMIC PROGRAMMING IMPLEMENTATION

Three well-known ways have been developed in which dynamic programming equations can be used to find optimal controls. The equations can be used to develop analytic proofs of optimality, or they can be used in one of two numerical techniques: policy iteration or value iteration. In the following paragraphs, each of these approaches are described. Because it is the most straightforward of the three, we then consider value iteration in detail. Applicability of the other two methods to this problem is a topic for future research.

Because we have shown that any policy that satisfies Eqs. (26), (27), and (28) is optimal, this set of equations can be used to test candidate policies for optimality. One way to apply this observation is to develop a policy through some independent technique and then to attempt to construct an analytic proof that the control applied for every state at every time is a maximizing control in Eq. (27). While such an approach is probably only feasible for relatively simple policies, it offers the possibility of avoiding a computational implementation altogether.

Howard has developed a computationally efficient technique called policy iteration that takes advantage of the fact that the dynamic programming equation can be used to define a contraction mapping in policy space [2]. His approach was developed for models with a long-time horizon and a reward function that is formed as the sum of a set of stage rewards. Beginning with an arbitrary control function and an arbitrary assignment of rewards for each state, the policy iteration algorithm iteratively applies Eq. (27) to develop a more nearly optimum control function and the associated rewards. Although this "policy iteration" must continue until a fixed point is reached, Howard reports that the algorithm often converges after just a few iterations.

Howard's policy iteration algorithm computes each succeeding control function by solving a set of linear equations. For reward functions formed as a product of a set of stage rewards, such as we have in our model, it is not clear that a practical analogue to this approach can be developed. Furthermore, we would expect that reformulation of our model to incorporate a longer time horizon would significantly alter the optimal policy choice. Nonetheless, it may prove worthwhile to investigate further the application of policy iteration to this class of reward functions given the problems that we describe below with implementation of the value iteration technique.

Equations (26), (27), and (28) can also be used directly to calculate the optimal control for each state at each time. Conceptually, this approach is quite simple. First, the results of Eq. (26) are computed and stored for each state. Equation (27) is then applied to each state, iterating this operation backwards in time until time 0 is reached. At each time the optimizing policy for each state is stored in an array. Finally, the given distribution on the initial state is used to find the expected reward by applying Eq. (28). This technique is known as value iteration. In the remainder of this report we describe the implementation details and the resulting computational complexity of value iteration dynamic programming.

5.1 State Coding

Because each state must be considered at each time, reducing the cardinality of the state space proportionally reduces both time and space requirements. For this reason, redundant information should be removed from the state before coding the algorithm. Redundant information is information that would not affect the value of the reward function if it were deleted.

Two types of redundant information exist in our model. The most obvious type is represented by unreachable states. Certain state variable combinations will never be reached, regardless of the control policy or the outcome of random events. For example, in the engagement model it is not possible for a SAM to be in flight ($S > 0$) when the SAM inventory is at its maximum value ($N = N_0$) since launching a SAM decrements N. Other state variable combinations can occur at some times, b\ not at others. For example, since only one SAM can be launched at each time step, N cannot reach 0 until N_0 time steps have elapsed. So the set of unreachable states varies with time.

A more subtle type of redundant information is represented by states that can be excluded without changing the value of the reward function, although the policy that achieves that reward function may change. We call these states redundant states, because, although they are initially reachable, the optimization problem can be reformulated in a way that makes them unreachable. Often, more than one policy will result in the same reward. We define policies that result in the same reward to be equivalent and use this relation to partition the policy space into equivalence classes. By introducing additional constraints on the control function, we may reduce the size of some of these equivalence classes while increasing the number of unreachable states. As long as one policy remains in the optimizing equivalence class, the dynamic programming algorithm presented above will find it.

A simple example may help to clarify this somewhat abstract concept. Consider an instance of the engagement model with one ASCM that is known to arrive at time 0, one SAM (i.e., $N_0 = 1$), and no chaff (i.e., $C_0 > \tau$). Since P_K is constant, the one SAM could be launched at any time that would allow it to reach the ASCM while the ASCM is in the middle zone. Every such policy would result in the same reward in the model we have presented. So every policy with exactly one SAM launch between time 0 and time $\tau_M - \frac{\tau_L}{V_S}$ is in the optimizing equivalence class. Introducing the constraint $(t < \tau_M - \frac{\tau_L}{V_S}) \Rightarrow L = \text{Hold}$ reduces the optimizing equivalence class to a single "Don't shoot until you see the whites of their eyes" policy and makes every state with $S > 0$ unreachable between time 0 and time $\tau_M - \frac{\tau_L}{V_S}$.

Because we have not developed a general procedure to recognize redundant states and craft control constraints that make them unreachable while preserving at least one policy in the optimizing equivalence class, an ad hoc approach has been adopted here. As our example shows, the choice of constraints and the resulting set of unreachable states depends on the values of the model parameters (consider how different the result would have been with $N_0 = 2$). For this reason, we have chosen not to pursue the issue of redundant states further.

Unreachable states are more easily recognized and eliminated because no new control constraints are required. In the engagement model, $T_i \in \hat{O}_t$ until the first seduction attempt occurs with ASCM i in the middle zone. But knowledge of T_i is never required to find the next state, and it is never used to compute $\hat{g}_t$. So we can calculate $\hat{J}_{\hat{\pi}}$ by using Eqs. (26), (27), and (28) without ever knowing T_i. This means that by simply eliminating T_i from the state (or equivalently forcing it to a fixed value) will change neither the policy nor the value of the reward function.

Similarly, C can be eliminated from the state because it depends in a known way on time, regardless of the policy employed. It is included in Fig. 8 simply as a notational convenience. Once C_0, C_{max} and t are specified; we can compute C as:

$$(t < C_0) \Rightarrow C = C_0 - t$$
$$(t \geq C_0) \Rightarrow C = (C_0 - t) \bmod C_{max}.$$

The remaining state variables (N, S, D, A_i, and $\hat{T}_i$) are all influenced by the control policy and are used in the computation of the reward function. The variables A_i and $\hat{T}_i$ are used directly to find $\hat{g}_t(\hat{O}_t)$, while D is used to find $\hat{T}_i$, S is used to find D and to find the distribution on A_i, and N is used to constrain the control function. It is still possible to find combinations of these state variables that cannot occur, however, as our first example in this section demonstrates.

Since the set of reachable states changes over time, a thorough search for unreachable states would require extensive analysis. The most general approach to eliminating unreachable states would be to enumerate the reachable states and map each reachable state to an index in this enumeration. As was the case for redundant states, however, unreachable states can be difficult to recognize in advance. We cannot use the dynamic programming algorithm to discover which states are unreachable because the algorithm proceeds backwards in time. Once we discovered that a state was unreachable, the work we sought to avoid would already have been done! A tree walk would discover which states are reachable, but it would also find the optimal policy (at great expense), obviating the need for dynamic programming. So again we are reduced to an ad hoc approach to eliminating unreachable states.

Programmers must also balance the potential savings obtained from encoding the state against the time and space required by the coding and decoding operations. Unreachable state variable combinations that have simple specifications in terms of existing state variables are easily eliminated. Those that require complex specifications or computationally expensive transformations on the state space may not be worth eliminating even if they could be specified. Using this observation to guide our search, we have found no combinations beyond the example given above ($(N = N_0) \Rightarrow S = 0$) that apply for all possible model parameters.

5.2 Data Structures

The dynamic programming algorithm iteratively uses the expected partial reward for every state to find the expected partial reward for each state one time step earlier. Because the partial rewards represent an expected probability of survival, they are real numbers between 0 and 1. So we will need at least two floating point arrays. No more than two arrays are needed because only the present partial rewards are required to find the partial reward for each state one time step earlier. We simply alternate between the two arrays to represent the present time step as we iterate backwards in time.

If our goal were simply to discover the optimal expected probability of survival, these two arrays would be sufficient. The dynamic programming algorithm will, however, also identify a policy that results in this optimal reward. Storing that policy requires an array indexed by both time and state. Because the launch control is binary valued, a Boolean array will suffice.

5.3 Algorithm Implementation

While Eqs. (26), (27), and (28) can be used to compute the optimal policy, two simplifications can reduce the implementation effort. If we define $\hat{V}_{T+\Delta}(\hat{O}_{T+\Delta}) = 1 \; \forall \hat{O}_{T+\Delta}$, Eq. (26) becomes a special case of Eq. (27). Unnecessary computations are easily avoided by choosing an arrival distribution with no arrivals at time τ and adopting a constraint that prevents SAM launches at time τ. Since $\hat{O}_0$ is known, Eq. (28) can be replaced with $\hat{J}_{\hat{\pi}^*} = \hat{V}_0(\hat{O}_0)$. So all we must compute is:

$$\hat{V}_t(\hat{O}_t) = \max_{\{\hat{\mu}_t(\hat{O}_t)\}} \{ \mathop{\mathrm{E}}_{\{\hat{O}_{t+\Delta}\}} [\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) \cdot \hat{g}_t(\hat{O}_t); \hat{\mu}_t(\hat{O}_t)]\}, \; (t \in \{0, \ldots, \tau\}). \tag{29}$$

To compute this quantity, we must find the distribution on the next state given the present state and the control. At each time, Eq. (3) can be used to compute the probability of detecting any particular number of ASCMs for each state. Figure 8 can also be used for each state to find the probability that a SAM will kill the closest ASCM. Because we have modeled these as independent events, we can then compute the probability of each possible random outcome. For each possible control the other state variables make deterministic transitions that are specified in Figs. 8 and 9.

Only control values that satisfy the constraints on the launch control need be considered. We do not allow another SAM to be launched if a SAM is already in flight ($S \geq 0$), the SAM inventory is exhausted ($N = 0$), or the final time has been reached ($t = \tau$). If one of these conditions holds, we only consider $L(t) =$ False. Otherwise, we choose the control that maximizes Eq. (29). Specifically, we compute:

$$\begin{aligned} \text{V1[state1]} = & \max_{\{\text{L[state1]}\}} \{ \sum_{\text{state2}} \Pr\{\text{transition from state1 to state2 ; L[state1]}\} \\ & \cdot \text{V2[state2]} \cdot \prod_{i \ni A_i = 0 \text{ in state1}} (1 - \hat{T}_i \text{ in state1})\} \end{aligned} \tag{30}$$

where V1 is the array of partial rewards at the prior time, V2 is the array of partial rewards at the present time, and L is the slice of the policy array for the prior time.

We are now ready to tie up the loose ends and present the algorithm. The time variable t is initially set to τ, and every value in V2 is set to 1. V1[state 1] is then calculated by using Eq. (30) for each state 1 at time t and the maximizing L[state 1] (either True or False) is stored in the policy array for state 1 at time t. Once V1[state 1] has been computed for every state, V2 is replaced by V1, t is decremented by Δ, and the process repeats using Eq. (30). This time $\hat{V}_{T-\Delta}(\cdot)$ and $\hat{\mu}_{T-\Delta}(\cdot)$ are computed. The iteration continues until the pass for t=0 has been completed. The value of $\hat{J}_{\hat{\pi}^*}$ can then be read from V1[state 1] for the value of state 1 that corresponds to the known initial value of $\hat{O}_0$. The policy array will contain a specification for a SAM launch policy that would result in this expected probability of survival.

5.4 Computational Complexity

The dynamic programming algorithm requires that for each control all of the states that can be reached from every state at every time be considered. Therefore, the number of operations that must be performed is directly proportional to the number of controls times the fanout from each state times the number of states times the number of time steps.

The fanout is the number of states that can be reached from a given state in one time step for a specified control. Since at most $m + 1$ ASCMs can arrive simultaneously and states that include an intercept (i.e., $S = 0$) could lead to one of two different states for each possible number of arrivals, the maximum fanout is $2(m + 1)$. Most states do not, however, include an intercept and most restrict the maximum number of arrivals to fewer than m ASCMs because some ASCMs have already been detected. For this reason, $\frac{m+1}{2}$ is probably a closer estimate of the average fanout. For the parameters in Table 2, an average fanout of approximately 2 would be expected.

The number of states turns out to be very large:

A_i There are $\frac{\tau_M+3}{\Delta}$ useful values for each A_i. Those values are $\{-\infty, \Delta, 1, \ldots, \tau_M, \infty\}$. Other positive numbers do not occur. All negative numbers can be mapped to $-\infty$ because doing so preserves the effect of A_i on each branch of every transition diagram in Figs. 8 and 9. Since there are m ASCMs, there are $(\frac{m(\tau_M+3)}{\Delta})^m$ possible values for $\{A_1, \ldots, A_m\}$. The parameters in Table 2 result in 51 possible values for each A_i, or 1.3×10^5 possible values for $\{A_1, \ldots, A_m\}$.

D There are $\frac{D_{max}}{\Delta} + 1$ useful values for D. Those values are $\{0, \Delta, \ldots, D_{max}\}$. Negative numbers do not occur. Other positive numbers can be mapped to D_{max} because the definition of D_{max} ensures that $(D > D_{max}) \Rightarrow P_S(D) = P_S(D_{max})$. The parameters in Table 2 result in 9 possible values for D.

N There are $N_0 + 1$ possible values for N. They are $\{0, 1, \ldots, N_0\}$. The parameters in Table 2 result in 6 possible values for N.

S There are $\lfloor \frac{\tau_M+\tau_I}{\Delta(V_S+1)} \rfloor + 2$ useful values for S. These values are $\{-\infty, 0, \Delta, \ldots, \lfloor \Delta \frac{\tau_M+\tau_I}{V_S+1} \rfloor\}$. Figure 8 shows that the largest possible value for S is $t_I(t) - t$. Substitution of the maximum possible value for A_i in Eq. (4) yields $\Delta \lfloor \frac{\tau_M+\tau_I}{\Delta(V_S+1)} \rfloor$ for this value. All negative numbers can be mapped to $-\infty$ because doing so preserves the effect of S on each branch of every transition diagram in Figs. 8 and 9. The parameters in Table 2 result in 17 possible values for S.

$\hat{T}_i$ Since there are $\frac{\tau_M}{\Delta} + 1$ time steps and $\frac{C_{max}}{\Delta} + 1$ time steps elapse between each seduction, there can be at most $\lceil \frac{\tau_M+\Delta}{C_{max}+\Delta} \rceil$ seduction attempts. The transition diagram in Fig. 9 shows that $\hat{T}_i$ will be multiplied by a function of D (which could assume one of $\frac{D_{max}}{\Delta} + 1$ values) at each seduction attempt. Since an unsuccessful seduction attempt ($D = 0$) results in the same value as no seduction attempt, $D = 0$ can be used to represent either situation. Because multiplication is commutative and associative, there are as many possible values of $\hat{T}_i$ as there are choices (with replacement) of values of D for each seduction attempt. Since there are m ASCMs, there are at most m times this number of possible values for $\{T_0, \ldots, T_m\}$. This value can be computed as:

$$\left[\begin{pmatrix} \frac{D_{max}}{\Delta} + 1 \\ \lceil \frac{\tau_M+\Delta}{C_{max}+\Delta} \rceil \end{pmatrix} + \begin{pmatrix} \frac{D_{max}}{\Delta} + 1 \\ \lceil \frac{\tau_M+\Delta}{C_{max}+\Delta} \rceil - 1 \end{pmatrix} + \cdots + \begin{pmatrix} \frac{D_{max}}{\Delta} + 1 \\ 1 \end{pmatrix}\right]^m$$

The parameters in Table 2 result in at most 3 possible seductions and 9 possible values for D at each seduction attempt, resulting in 129 possible values for $\hat{T}_i$, or 2.1×10^6 possible values for $\{\hat{T}_0, \ldots, \hat{T}_m\}$.

Multiplying the values for the parameters in Table 2 together, we find there are over 2.6×10^{14} possible states. Since $(N = N_0) \Rightarrow (S = 0)$, a slight reduction to approximately 2.2×10^{14} states can be achieved by coding N and S together.

Because the launch control is binary, the maximum number of possible controls is two. The constraint that precludes a SAM launch when a SAM is already in flight makes the average number of controls much closer to one, though, because $S > 0$ in most of the states. Finally, there are $\frac{T}{\Delta} + 1$ possible time steps. For the parameters in Table 2, there are 148 possible time steps.

Combining these results, the dynamic programming algorithm requires retrieval of approximately 6.5×10^{16} floating point numbers from memory and requires us to perform a proportional number of arithmetic and logical operations. Even with a 10 gigabyte per second memory band-width it would take over two months to simply perform the required memory accesses. Each of the two floating point partial reward arrays contains approximately 220 trillion values; together, they require about 1800 terabytes of random access memory when stored with single precision. The policy array consists of approximately 220 trillion bits for each of the 148 time steps, and they require over 4000 terabytes of offline storage during the computation and an equal amount of random access memory during policy employment.

The strength of value iteration dynamic programming is that it provides an exact global solution for combinatorial optimization problems without resorting to exhaustive search. For the model we have developed above, the computational complexity of using dynamic programming in this way is obviously well beyond the capability of existing computer systems. In the following discussion we will describe two alternative approaches for resolving this dilemma. First we consider the potential for restricting our model in ways that will result in improved implementation efficiency. Since it remains an open question whether the approaches we describe below retain sufficient fidelity to allow application of value iteration dynamic programming to practical problems, we conclude with a brief discussion of potential for developing heuristic optimization techniques for these problems.

5.5 Reducing Computational Complexity

The four factors contributing to the complexity of the dynamic programming algorithm are the number of states, the number of available controls in each state, the fanout from each state, and the number of time steps. There is little point to further constraining the number of available controls in each state because only one possible control exists in most of the states and no more than two controls are possible in any state in our model. However, we can gain significantly by reducing the number of time steps. The number of time steps can be reduced by increasing Δ or by introducing an assumption such as clustered arrivals that would allow τ to be decreased. Furthermore, increasing Δ would significantly reduce the number of states and clustered arrivals, and would also reduce fanout. Thus we begin by considering changes that reduce the number of time steps and then discuss other assumptions that could be introduced to reduce the state space still further.

5.5.1 Time Step Reduction

The value of Δ in Table 2 results in a separation between values of D that is appropriate to the accuracy with which data for $P_S(D)$ was collected. Increasing Δ results in coarser coding for D, leading to a less accurate estimate of seduction effectiveness. This decreased accuracy could lead to a overdependence or underdependence on the chaff, depending on the type of error introduced. Thus, a significant increase in Δ could make the results of the optimization less useful.

In addition to reducing the number of time steps, increasing Δ also reduces the number of possible values for every state variable except N. For the parameters in Table 2, the number of possible states decreases with the eighth power of the factor by which Δ is increased. Doubling Δ would reduce the number of possible states by a factor of 256. Coupled with the time step reduction, the computational complexity is reduced by the ninth power, a factor of 512 for doubling Δ. This extreme sensitivity to Δ places great weight on the choice of the maximum practical value for this parameter.

The potential improvement resulting from reducing τ is less dramatic. We consider an engagement to begin when the first ASCM is detected. The value of τ in Table 2 is sufficient to find the optimal policy regardless of the subsequent ASCM arrival pattern, subject only to the requirement that every ASCM shares the middle zone with at least one other ASCM for at least one time instant. It was found by considering the case in which each subsequent ASCM is detected just as the prior ASCM reaches the inner zone. We have not considered the case of ASCMs that never share the middle zone with another ASCM because that case is easily handled by decomposing the multiple ASCM engagement into a sequence of single ASCM engagements. The optimal policy for the multiple ASCM engagement could then be found by sequentially finding the optimal policy for each single ASCM engagement.

Restricting all arrivals to the first few seconds of the engagement would reduce τ by a factor between 2 and 3. This is a reasonable restriction because clustered arrivals are known to be an effective attack technique. Restricting ASCM arrival times in this way, times also reduce the average fanout from around 2 to a value close to 1. Taken together, this approach offers a factor of 5 improvement in speed and a factor of 2.5 improvement in the number of states.

5.5.2 State Space Reduction

Together, the two approaches to time step reduction could yield a performance improvement of about three orders of magnitude if, for example, analysis showed that Δ could safely be doubled. Even with that speedup, additional improvement would be required before a practical implementation could be developed. One approach is to simply eliminate the inventory constraint. If enough SAMs were available to the defender to assure their availability throughout the engagement, the choice of controls would not depend on the inventory. Thus the state variable N could be deleted and the SAM launch constraint simplified by eliminating the $N > 0$ requirement. For the parameters in Table 2, this would reduce the number of state variables by a factor of 6.

A bolder approach would be to arbitrarily reduce the precision with which a state variable is recorded. While this approach cannot be applied to counting processes such as N, S and A_i, it offers great promise when applied to $\hat{T}_i$. Initial experiments with the parameters shown in Table 2 (with a restriction to clustered arrivals to speed execution) show that approximately 10 evenly spaced values are sufficient to compute a value for J_{π^*} that is accurate to within 0.05 of the value found when exact values were used. Such coding results in 1,000 possible values for $\{T_0, \ldots, T_m\}$: a reduction by a factor of over 2,000 compared to the number of possible values for exact coding. Unfortunately, the policy found in this way could differ significantly from the optimal policy found by using exact coding. Additional research is required to understand how quantizing T_i affects the optimal policy and to find the best quantization algorithm.

In summary, eliminating the inventory constraint and clustering arrivals would reduce the computational complexity by one order of magnitude while retaining the ability to accurately model a wide range of realistic engagements. By taking these steps and doubling Δ and quantizing T_i to 10 values, we could achieve a seven order of magnitude improvement over a model with the parameters shown in Table 2. This improvement comes at the expense of some loss of accuracy that remains to be quantified, but it allows a solution to be calculated for a problem with parameters similar to those in Table 2 without placing impractical demands on computer resources.

6 FUTURE RESEARCH

Practical ship defense systems will require models that incorporate additional dynamics and a wider range of parameters. In practice, ASCM seeker parameters may not be uniform, and may not be known a priori. Other defensive systems must be integrated with the two considered here. And protracted engagements consisting of several waves of ASCMs may have to be addressed. These requirements would yield larger state spaces and greater fanout, and may require many more time steps. Even the improvements of the type just described offer no practical hope for dealing with this sort of computational complexity.

The simplifying assumptions in Section 5.5 alone will not be adequate for models of this complexity. The principal advantage of the dynamic programming algorithm described in this report is that it produces an optimal solution without considering the alternatives available in each state at each time more than once. Its principal limitation is that we must perform the computation for every state, regardless of whether the state is reachable. This limitation arises because we work backwards in time in the dynamic programming algorithm, and hence we are unable to recognize unreachable states during the computation. A forward tree search for the optimal policy avoids evaluating unreachable states at the cost of greater time or space complexity. To apply dynamic programming to these more challenging models would require the investigation of other uses of the dynamic programming equations such as policy iteration and analytic proofs of optimality. If these techniques do not prove to be feasible, it will be necessary to consider heuristic optimization techniques.

A number of heuristic techniques that have been applied to similar problems appear to offer some promise for application to the ship defense problem. SAMUEL is a computer program developed by the Navy Center for Artificial Intelligence Research that uses a genetic algorithm to develop a specification for a near-optimal policy [3]. Policies in SAMUEL are specified by a set of rules that are used to select controls based on observations. The genetic algorithm in SAMUEL adds, deletes, and changes those rules in an attempt to improve the value of the reward function. Application of SAMUEL to the ship defense problem would require the creation of a world model that is used by the simulation module in SAMUEL to evaluate candidate policy specifications.

The effectiveness of the technique used by SAMUEL depends on the suitability of a relatively small rule set for specification of near-optimal policies. Because we expect to be able to achieve an optimal dynamic programming implementation for our limited engagement model described above, evaluation of SAMUEL's performance on that model should be straightforward. Furthermore, we would hope to gain some insight into the usefulness of policy specification using small rule sets. Although the policy array is generally far too large for online storage, small rule sets would be useful in developing online decision aids and tactical guidance for defensive system operators.

Another technique worth investigating is a limited lookahead forward tree search. Rather than evaluate every path through the complete state space, the number of time steps to be considered is sharply limited and a heuristic evaluation of the partial reward is performed for each state when that limit is reached. By limiting the depth of the tree search we can avoid both consideration of unreachable states and extensive reconsideration of the same state. This approach is similar to the technique used by computer chess programs; however, the introduction of stochastic transitions could result in some significant differences in the implementation of the concept.

Selection of an appropriate heuristic for the static evaluation of terminal states is critical to the performance of the limited look ahead technique. One possible heuristic would be to use the same reward that SAMUEL would use to evaluate candidate rule sets. SAMUEL determines the performance of a set of rules from a known starting state by Monte Carlo simulation. To use SAMUEL's approach for static evaluation of an intermediate state, SAMUEL would first be used to find a good set of rules, and then SAMUEL's simulation module would be applied to the state being considered to find the expected performance of that set of rules starting from the specified state. An advantage of the limited look ahead technique is that it might be possible to apply it in real time rather than developing a policy off-line. If rules turn out to be poorly suited for specifying near-optimal policies, real-time policy development may be the only practical alternative. The availability of an optimal dynamic programming implementation for the engagement model would also facilitate performance evaluation in this case.

A third approach that could be useful, if small rule sets are not found to be effective, is to develop a neural network that could generate near-optimal controls. In one such approach, a backpropagating neural network could be trained off-line by using an optimal dynamic programming solution for a limited engagement model and then it could be used in real time to generate the optimal controls. Werbos has proposed using backpropagation through time to train the world model of a backpropagating adaptive critic [4].

Finally, Hopfield neural networks offer the possibility of directly developing an optimal policy, if a short binary specification for a policy could be developed. The rule set developed by SAMUEL may provide some insight into the development of an appropriate policy representation.

7 CONCLUSION

Because value iteration dynamic programming requires that we work backward in time, we must consider every combination of state variables at every time. Although we may be able to find an optimal solution for a small problem using dynamic programming, solving problems based on larger models requires more computer resources than can practically be provided. For this reason we are motivated to search for faster techniques. Both heuristic techniques and alternative uses of the dynamic programming equations bear further investigation. The value iteration dynamic programming solution we have developed is useful, however, since it can be used to gain insight into the design of such algorithms and to evaluate the performance of those algorithms in a restricted domain.

REFERENCES

[1] K. M. Potter, "Optimal Resource Allocation for a Controller with Two Resource Types," NRL Memorandum Report 6553, October 1989.

[2] R.A. Howard, *Dynamic Programming and Markov Processes* (Wiley and Sons, New York, N.Y., 1960), pp. 32-43.

[3] J. J. Grefenstette and H. J. Cobb, "User's Guide for SAMUEL, Version 3," NRL Memorandum Report, January 1993.

[4] P. J. Werbos, "Backpropagation and Neurocontrol: A Review and Prospectus," *International Joint Conference on Neural Networks.* Piscataway, N.J.: IEEE TAB Neural Network Committee, 1, 209-216 (1989).

[5] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models* (Prentice-Hall, Englewood Cliffs, N.J., 1987), pp. 12-22.

Appendix A

DYNAMIC PROGRAMMING WITH MULTIPLICATIVE REWARDS

A.1 GIVEN

We are given an initial time 0, a constant time quantum of 1 arbitrary unit, a nonnegative integer final time τ, random variables $\{X_0, \ldots, X_T\}$ taking values $\{x_0, \ldots, x_T\}$ that represent states, a class of control policies $\pi = \{\mu_0(\cdot), \ldots, \mu_{T-1}(\cdot)\}$, and a reward function J_π with the following properties:

(a) $x_t \in \mathcal{X}_t \; \forall(t \in \{0, \ldots, \tau\})$, where $\mathcal{X}_t$ is a finite set that represents the possible states at time t.

(b) $\mu_t : \mathcal{X}_t \to \mathcal{U} \; \forall(t \in \{0, \ldots, \tau - 1\})$, where $\mathcal{U}$ is a finite set that represents the available controls. Furthermore, $\mu_t(x_t) \in \mathcal{U}_t(x_t) \; \forall(t \in \{0, \ldots, \tau - 1\}), \forall(x_t \in \mathcal{X}_t)$, where $\mathcal{U}_t(x_t)$ is a finite set that represents the admissible controls at time t if the state is x_t.

(c) The distribution on X_0 is given and the distribution on X_{t+1} is completely determined by x_t and $\mu_t(x_t) \; \forall t \in \{0, \ldots, \tau - 1\}$:

$$\begin{aligned} &\Pr\{X_{t+1} = x_{t+1} | X_0 = x_0, \ldots, X_t = x_t; \mu_0(\cdot), \ldots, \mu_T(\cdot)\} \\ =\; &\Pr\{X_{t+1} = x_{t+1} | X_t = x_t; \mu_t(x_t)\} \end{aligned}$$

By, for example, $\Pr\{X_{t+1} = x_{t+1} | X_t = x_t; \mu_t(x_t)\}$ we mean the probability of the event $\{X_{t+1} = x_{t+1}\}$ conditioned on the event $\{X_t = x_t\}$, given the value of $\mu_t(x_t)$ as a parameter.

(d) At each time $t \in \{0, \ldots, \tau\}$, a nonnegative reward factor $g_t(x_t)$ that depends only on the present state x_t is computed.

(e) The overall reward is the expectation of the product of the reward factors (and is a function of the control policy π):

$$J_\pi = \mathop{\mathrm{E}}_{\{X_0, \ldots, X_T\}} [\prod_{t=0}^{T} g_t(X_t)]$$

Our goal is to find the maximum reward that can be earned over all admissible policies and a policy which earns that reward. Formally:

$$\begin{aligned} J^* &= \max_\pi \{ \mathop{\mathrm{E}}_{\{X_0, \ldots, X_T\}} [\prod_{t=0}^{T} g_t(X_t)]\} \\ \pi^* &= \arg\max_\pi \{ \mathop{\mathrm{E}}_{\{X_0, \ldots, X_T\}} [\prod_{t=0}^{T} g_t(X_t)]\} \end{aligned}$$

Where we maximize over π we mean maximization over $\mu_t(x_t) \in \mathcal{U}_t(x_t), \forall(t \in \{0, \ldots, \tau - 1\}), \forall(x_t \in \mathcal{X}_t)$.

A.2 CLAIM

Property (e) defines a reward function with a multiplicative structure. For reward functions with an additive cost structure, the dynamic programming algorithm described by Bertsekas in [5] provides a solution technique with a time complexity that is $O(\tau)$. Here we describe an analogous algorithm for a multiplicative reward function.

We define a set of functions $V_t(x_t)$ as:

$$\begin{aligned} V_T(x_T) &= g_T(x_T) \\ V_t(x_t) &= g_t(x_t) \cdot \max_{\mu_t(x_t)} \{ \mathop{E}_{\{X_{t+1}\}} [V_{t+1}(X_{t+1})] \}, \; (t \in \{0,\ldots,\tau-1\}), \end{aligned}$$

and then claim we can compute J_{π^*} as

$$J_{\pi^*} = \mathop{E}_{\{X_0\}} [V_0(X_0)].$$

Furthermore, we define:

$$\begin{aligned} \hat{\pi}_T &= \emptyset \\ \hat{\mu}_t(x_t) &= \arg\max_{\mu_t(x_t)} \{ \mathop{E}_{\{X_{t+1}\}} [V_{t+1}(X_{t+1})] \}, \; (t \in \{0,\ldots,\tau-1\}) \\ \hat{\pi}_t &= \{\hat{\mu}_t(\cdot)\} \cup \hat{\pi}_{t+1}, \; (t \in \{0,\ldots,\tau-1\}), \end{aligned}$$

where $\emptyset$ represents the empty set, and claim that

$$\pi^* = \hat{\pi}_0.$$

A.3 PROOF

We begin with the special case of $\tau = 0$. Here the result follows directly from the definitions

$$\begin{aligned} J_{\pi^*} &= \max_{\pi} \{ \mathop{E}_{\{X_0\}} [\prod_{t=0}^{0} g_t(X_t)] \} \\ &= \mathop{E}_{\{X_0\}} [g_0(X_0)] \\ &= \mathop{E}_{\{X_0\}} [V_0(X_0)]. \end{aligned}$$

Thus we may focus our attention on the case in which $\tau \geq 1$. We will show by induction on n that $\forall n \in \{0,\ldots,\tau-1\}$

$$\mathop{E}_{\{X_0\}} [V_0(X_0)] = \max_{\mu_0(\cdot)} \ldots \max_{\mu_n(\cdot)} \{ \mathop{E}_{\{X_0,\ldots,X_{n+1}\}} [V_{n+1}(X_{n+1}) \cdot \prod_{t=0}^{n} g_t(X_t)].$$

For the basis case we will take $n = 0$ and show that

$$\begin{aligned} \mathop{E}_{\{X_0\}} [V_0(X_0)] &= \max_{\mu_0(\cdot)} \{ \mathop{E}_{\{X_0,X_1\}} [V_1(X_1) \cdot \prod_{t=0}^{0} g_t(X_t)] \\ &= \max_{\mu_0(\cdot)} \{ \mathop{E}_{\{X_0,X_1\}} [V_1(X_1) \cdot g_0(X_0)]. \end{aligned}$$

Expanding the left side by using the definitions of V_0 and expectation, bringing two nonnegative constants inside a maximization and a summation, and then interchanging maximization and summation

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] \\
&= \mathop{\mathrm{E}}_{\{X_0\}}[g_0(X_0)\cdot \max_{\mu_0(X_0)}\{\mathop{\mathrm{E}}_{\{X_1\}}[V_1(X_1)]\}] \\
&= \sum_{x_0\in\mathcal{X}_0} \Pr\{X_0=x_0\}\cdot g_0(x_0)\cdot \max_{\mu_0(x_0)}\{\sum_{x_1\in\mathcal{X}_1}\Pr\{X_1=x_1|X_0=x_0;\mu_0(x_0)\}\cdot V_1(x_1)\} \\
&= \sum_{x_0\in\mathcal{X}_0}\max_{\mu_0(x_0)}\{\sum_{x_1\in\mathcal{X}_1}\Pr\{X_1=x_1|X_0=x_0;\mu_0(x_0)\}\cdot\Pr\{X_0=x_0\}\cdot V_1(x_1)\cdot g_0(x_0)\} \\
&= \max_{\mu_0(\cdot)}\{\sum_{x_0\in\mathcal{X}_0}\sum_{x_1\in\mathcal{X}_1}\Pr\{X_1=x_1|X_0=x_0;\mu_0(x_0)\}\cdot\Pr\{X_0=x_0\}\cdot V_1(x_1)\cdot g_0(x_0)\}
\end{aligned}$$

This last step requires some justification. With the maximization inside the summation we are separately choosing the values of $\mu_0(x_0)$ that maximize each term of the summation. Maximizing outside the summation requires us to simultaneously choose values of $\mu_0(\cdot)$ for every possible value of x_0 that maximize the sum. In this case the two operations are equivalent because each choice of $\mu_0(x_0)$ affects at most one term of the sum. In particular, property (c) ensures that $\Pr\{X_1 = x_1|X_0 = x_0;\mu_0(x_0)\}$ depends on no other choice of $\mu_0(\cdot)$. Also note that $V_1(x_1)$ does not vary with $\mu_0(x_0)$ because no information for earlier times is used in the construction of V_t. In general, we can exchange maximization and summation in this way whenever the index of the outer summation assumes values from the domain of the function over which we are maximizing; each term in the summation depends on the value of the function over which we are maximizing for at most one element of its domain, and the value of the function for each element in its domain affects at most one term in the summation.

To complete the proof of the basis case we apply the definitions of conditional probability and expectation

$$\begin{aligned}
&\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] \\
&= \max_{\mu_0(\cdot)}\{\sum_{x_0\in\mathcal{X}_0}\sum_{x_1\in\mathcal{X}_1}\Pr\{X_1=x_1,X_0=x_0;\mu_0(x_0)\}\cdot V_1(x_1)\cdot g_0(x_0)\} \\
&= \max_{\mu_0(\cdot)}\mathop{\mathrm{E}}_{\{X_0,X_1\}}[V_1(x_1)\cdot g_0(x_0)]\}.
\end{aligned}$$

We now take as our inductive hypothesis for $n \in \{1,\ldots,\tau-1\}$

$$\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] = \max_{\mu_0(\cdot)}\ldots\max_{\mu_{n-1}(\cdot)}\{\mathop{\mathrm{E}}_{\{X_0,\ldots,X_n\}}[V_n(X_n)\cdot\prod_{t=0}^{n-1}g_t(X_t)]$$

and show that

$$\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] = \max_{\mu_0(\cdot)}\ldots\max_{\mu_n(\cdot)}\{\mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n+1}\}}[V_{n+1}(X_{n+1})\cdot\prod_{t=0}^{n}g_t(X_t)].$$

We begin by separating the expectation on the right side of the inductive hypothesis, bringing a constant outside the inner expectation, expanding V_n, and bringing two nonnegative constants inside the inner maximization:

$$
\begin{aligned}
&\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] \\
&= \max_{\mu_0(\cdot)} \ldots \max_{\mu_{n-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_0,\ldots,X_n\}} [V_n(X_n) \cdot \prod_{t=0}^{n-1} g_t(X_t)]\} \\
&= \max_{\mu_0(\cdot)} \ldots \max_{\mu_{n-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [\mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [V_n(X_n) \cdot \prod_{t=0}^{n-1} g_t(X_t)]]\} \\
&= \max_{\mu_0(\cdot)} \ldots \max_{\mu_{n-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [V_n(X_n) \cdot \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [\prod_{t=0}^{n-1} g_t(X_t)]]\} \\
&= \max_{\mu_0(\cdot)} \ldots \max_{\mu_{n-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [g_n(X_n) \cdot \max_{\mu_n(X_n)} \{ \mathop{\mathrm{E}}_{\{X_{n+1}\}} [V_{n+1}(X_{n+1})])\} \cdot \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [\prod_{t=0}^{n-1} g_t(X_t)]]\} \\
&= \max_{\mu_0(\cdot)} \ldots \max_{\mu_{n-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [\max_{\mu_n(X_n)} \{ \mathop{\mathrm{E}}_{\{X_{n+1}\}} [V_{n+1}(X_{n+1})] \cdot g_n(X_n) \cdot \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [\prod_{t=0}^{n-1} g_t(X_t)]\}]\}.
\end{aligned}
$$

Now observe that the expectation over X_n is taken on an expression that depends on only one value of $\mu_n(X_n)$. Viewing the expectation as a weighted sum, the index of the summation assumes values from the domain of the function over which we are maximizing; each term in the summation depends on the value of the function over which we are maximizing for at most one element of its domain, and the value of the function for each element in its domain affects at most one term in the summation. So we can exchange that expectation with the inner maximization. Doing this and moving constants inside expectations

$$
\begin{aligned}
\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] &= \max_{\mu_0(\cdot)} \cdots \max_{\mu_n(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [\mathop{\mathrm{E}}_{\{X_{n+1}\}} [V_{n+1}(X_{n+1})] \cdot g_n(X_n) \cdot \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [\prod_{t=0}^{n-1} g_t(X_t)]]\} \\
&= \max_{\mu_0(\cdot)} \cdots \max_{\mu_n(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [\mathop{\mathrm{E}}_{\{X_{n+1}\}} [V_{n+1}(X_{n+1}) \cdot g_n(X_n) \cdot \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [\prod_{t=0}^{n-1} g_t(X_t)]]]\} \\
&= \max_{\mu_0(\cdot)} \cdots \max_{\mu_n(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_n\}} [\mathop{\mathrm{E}}_{\{X_{n+1}\}} [\mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n-1}\}} [V_{n+1}(X_{n+1}) \cdot g_n(X_n) \cdot \prod_{t=0}^{n-1} g_t(X_t)]]]\}.
\end{aligned}
$$

Combining the expectations and the product of the reward factors completes the induction

$$
\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] = \max_{\mu_0(\cdot)} \cdots \max_{\mu_n(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_0,\ldots,X_{n+1}\}} [V_{n+1}(X_{n+1}) \cdot \prod_{t=0}^{n} g_t(X_t)]\}.
$$

Choosing $n = \tau - 1$ and applying the definitions of V_τ and π

$$
\begin{aligned}
\mathop{\mathrm{E}}_{\{X_0\}}[V_0(X_0)] &= \max_{\mu_0(\cdot)} \cdots \max_{\mu_{\tau-1}(\cdot)} \{ \mathop{\mathrm{E}}_{\{X_0,\ldots,X_\tau\}} [V_\tau(X_\tau) \cdot \prod_{t=0}^{\tau-1} g_t(X_t)]\} \\
&= \max_{\pi} \{ \mathop{\mathrm{E}}_{\{X_0,\ldots,X_\tau\}} [g_\tau(X_\tau) \cdot \prod_{t=0}^{\tau-1} g_t(X_t)]\} \\
&= \max_{\pi} \{ \mathop{\mathrm{E}}_{\{X_0,\ldots,X_\tau\}} [\prod_{t=0}^{\tau} g_t(X_t)]\} \\
&= J_{\pi^*}.
\end{aligned}
$$

This proves the first claim. The proof of the second claim follows the same approach with more attention to the identity of the maximizing control functions $\mu_t(\cdot)$.

Appendix B

EQUIVALENCE OF THE SIMPLIFIED POLICY

We claim that maximizing $\hat{J}_{\hat{\pi}}$ is equivalent to maximizing J_π in the sense that for any optimal policy π^* there is an optimal policy $\hat{\pi}^*$ such that

$$\hat{J}_{\hat{\pi}^*} = J_{\pi^*}$$

To prove this, it suffices to show that $\hat{V}_0(\hat{O}_0) = V_0(I_0)\ \forall \hat{O}_0\ \forall I_0 = \hat{O}_0$. To accomplish this, we will prove by induction backwards on t from τ to 0 by Δ that

$$\hat{V}_t(\hat{O}_t) = V_t(I_t),\ \forall t \in \{0, \ldots, \tau\}\ \forall \hat{O}_t\ \forall I_t \ni \hat{O}_t = \mathcal{T}(t, I_t).$$

For the basis case, we begin with $\hat{V}_\mathcal{T}(\hat{O}_\mathcal{T})$ and apply Eqs. (26), (20), (19), (16) and (17) to get

$$\begin{aligned}\hat{V}_\mathcal{T}(\hat{O}_\mathcal{T}) &= \prod_{\{i \ni A_i(\mathcal{T})=0\}} (1 - \hat{\mathcal{T}}_i(\tau)) \\ &= V_\mathcal{T}(I_\mathcal{T})\end{aligned}$$

which proves the basis case. For the inductive step we assume

$$\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) = V_{t+\Delta}(I_{t+\Delta}),\ \forall \hat{O}_{t+\Delta}\ \forall (I_{t+\Delta} \ni \hat{O}_{t+\Delta} = \mathcal{T}(t+\Delta, I_{t+\Delta})), t \in \{\Delta, \ldots, \tau - \Delta\}$$

and must show that

$$\hat{V}_t(\hat{O}_t) = V_t(I_t),\ \forall \hat{O}_t\ \forall (I_t \ni \hat{O}_t = \mathcal{T}(t, I_t)).$$

Let $\hat{O}_t$ be arbitrary. Then, by Eqs. (18), (16), (19), (20) and the inductive hypothesis

$$\begin{aligned}V_t(I_t) &= \max_{\mu_t(I_t)}\{\underset{\{I_{t+\Delta}\}}{\mathrm{E}}[V_{t+\Delta}(I_{t+\Delta}) \cdot \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t))]\} \\ &= \max_{\mu_t(I_t)}\{\underset{\{I_{t+\Delta}\}}{\mathrm{E}}[\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) \cdot \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t))]\}.\end{aligned}$$

Observing that the function inside the expectation depends on no aspects of $I_{t+\Delta}$ other than $\hat{O}_{t+\Delta}$ and that $\mathcal{T}(t+\Delta, \cdot)$ induces a partition on $I_{t+\Delta}$, we can now write:

$$V_t(I_t) = \max_{\mu_t(I_t)}\{\underset{\{\hat{O}_{t+\Delta}\}}{\mathrm{E}}[\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) \cdot \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t))]\}.$$

When we write $\mu_t(I_t)$, we mean the control $L(t)$ that is selected at time t. Together, $\hat{O}_t$ and $L(t)$ determine the distribution on $\hat{O}_{t+\Delta}$. Observe, however, that the value of the expectation is a function of the value of $\hat{O}_t$ and the distribution on $\hat{O}_{t+\Delta}$. So, while different controls may maximize the expectation for the same $\hat{O}_t$, all must result in the same maximal value. Thus, any maximizing control may be chosen arbitrarily. Therefore we apply Eqs. (25)and (27) to write

$$\begin{aligned} V_t(I_t) &= \max_{\hat{\mu}_t(\hat{O}_t)} \{ \mathop{\mathrm{E}}_{\{\hat{O}_{t+\Delta}\}} [\hat{V}_{t+\Delta}(\hat{O}_{t+\Delta}) \cdot \prod_{i \ni A_i(t)=0} (1 - \hat{T}_i(t))] \\ &= \hat{V}_t(\hat{O}_t) \end{aligned}$$

as was to be shown.