Computers and Operations Research 117 (2020) 104890

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An approximate dynamic programming approach for comparing firing policies in a networked air defense environment

Daniel S. Summers, Matthew J. Robbins*, Brian J. Lunday

Department of Operational Sciences, Air Force Institute of Technology, Wright Patterson AFB, Ohio 45433, United States

ARTICLE INFO

Article history:
Received 14 September 2018
Revised 4 October 2019
Accepted 11 January 2020
Available online 14 January 2020

Keywords:
Military
Air and missile defense
Dynamic weapon target assignment problem
Markov decision processes
Approximate dynamic programming

ABSTRACT

An objective for effective air defense is to identify the firing policy for interceptor allocation to incoming missiles that minimizes the expected total damage to defended assets over a sequence of engagements. We formulate this dynamic weapon target assignment problem as a Markov decision process and utilize a simulation-based, approximate dynamic programming (ADP) approach to solve problem instances based on a representative scenario. Least squares policy evaluation and least squares temporal differences algorithms are developed to determine approximate solutions. A designed experiment investigates problem features such as conflict duration, attacker and defender weapon sophistication, and defended asset values. An empirical comparison of the ADP policies and two baseline policies (i.e., firing either one or two interceptors at each incoming theater ballistic missile (TBM)) yields several insights: the ADP policies outperform both baseline polices when conflict duration is short and attacker weapons are sophisticated; firing one interceptor at each TBM (regardless of inventory status) outperforms the tested ADP policies when conflict duration is long and attacker weapons are less sophisticated; and firing two interceptors at each TBM (regardless of inventory status), which is the United States Army's currently implemented policy, is never the superlative policy for the test instances investigated.

Published by Elsevier Ltd.

1. Introduction

The proliferation of Theater Ballistic Missiles (TBMs) is increasing as countries seek the ability to project power in regional and strategic contexts (U.S. Missile Defense Agency, 2016e). Some nations (e.g., North Korea, Iran) stockpile less sophisticated versions while other nations (e.g., China) continue to advance their technology to develop and acquire faster moving missiles, missiles having multiple reentry vehicles, and maneuverable missiles capable of significantly altering their ballistic trajectory (Schaffer, 2016; U.S. Missile Defense Agency, 2016e). The threat from such weapons has motivated many countries to field and operate air and missile defense systems. For example, the United States' Patriot air and missile defense system is fielded by 17 countries (Raytheon, 2019), including Germany, Israel, Japan, Kingdom of Saudi Arabia, Qatar, Republic of Korea, Sweden, and Taiwan (Raytheon, 2018). Moreover, the United States continues to develop and coordinate missile defense plans with allied countries in Europe, Africa, and Asia (Jeong, 2019; Mindock, 2015). The ongoing proliferation of both offensive and defensive missile systems speaks to the global importance of this topic.

Throughout the first half of 2016, North Korea test-launched numerous TBMs, to include a Musudan Intermediate Range Ballistic Missile in June that traveled almost 250 miles before crashing into the Sea of Japan (Copp, 2016). Secretary of Defense Ashton Carter affirmed the United States' commitment to TBM defense in response to this launch (Copp, 2016). North Korea then fired a KN-11 ballistic missile from a submarine on July 9th, further provoking tensions in the region (Kwon, 2016). In response to these launches, the United States and South Korea agreed to move a Terminal High Altitude Air Defense (THAAD) site to the peninsula on July 13th (Kim and Park, 2016). This move has been highly criticized by both Chinese and Russian officials as a destabilizing action (Kim, 2016). These events, as well as the continued military presence of the United States in the Middle East, underscore a critical need for intelligent TBM defense policies, both with regard to *deployment* and *employment*, the latter of which is the focus of the research herein.

The United States divides its TBM defense into three segments (i.e., phases): the boost defense segment, the mid-course defense segment, and the terminal defense segment (U.S. Missile Defense Agency, 2016a). The Missile Defense Agency (MDA) has expended significant resources on boost segment defense, yet it remains dif-

* Corresponding author.
E-mail address: matthew.robbins@afit.edu (M.J. Robbins).

https://doi.org/10.1016/j.cor.2020.104890
0305-0548/Published by Elsevier Ltd.

ficult to intercept TBMs with any level of accuracy when they are in the boost phase of their trajectory. Therefore, the MDA limits efforts in this segment mostly to providing early launch detection. Underscoring this policy is a 2012 National Research Council report stating, "Boost-phase missile defense is not practical [n]or cost-effective under real-world conditions for the foreseeable future" (National Research Council, 2012). The boost phase limitations reduce intercept opportunities to the mid-course and terminal phase of the TBM's trajectory.

The Aegis Ballistic Missile Defense (BMD) system provides a mid-course defense capability with the Standard Missile-3 (SM-3) interceptor. The United States Navy currently employs 33 Aegis capable platforms - five cruisers and 28 destroyers - and ongoing efforts seek to expand the Aegis ashore variant beyond its current deployment in Hungary. The Aegis BMD system began development under the Reagan administration, performing its first successful flight test intercept in 2002 and becoming fully operational in 2005 (U.S. Missile Defense Agency, 2016b).

The terminal defense segment offers the highest probability of intercept by current air and missile defense systems, but it also portends the highest threat to the defended assets. The United States currently relies on the Patriot air defense system, the THAAD system, and the Aegis with its Standard Missile-2 (SM-2) interceptor for terminal defense (U.S. Missile Defense Agency, 2016d).

The United States relies upon a segmented defense-in-depth strategy advocated by the MDA because it cannot rely solely on the Patriot or the THAAD to defend assets in the terminal phase. Paramount to this strategy is the ability to network the limited air and missile defense assets available to provide a common air picture, provide early detection, and allow for a larger and more tailored coverage area. The United States Army is currently developing, under contract with Northrop Grumman, the Integrated Air and Missile Defense Battle Command System (IBCS) (U.S. Missile Defense Agency, 2016c). Once fielded, this networked command and control system will integrate individual systems like the Patriot and the THAAD and provide a coordinated defense-in-depth.

Although a networked system of air and missile defense assets improves the ability to detect, identify, track, and engage an enemy TBM, it also induces the challenge of deciding which air and missile defense system within the system should engage each incoming TBM and with how many interceptors. A fundamental tension exists between the potential catastrophic damage caused by TBMs and the extremely limited number of air and missile defense system interceptors available. Given an air and missile defense site defending against a single salvo of incoming missiles, formulating and solving a static weapon-target assignment problem can identify the best firing solution to protect the defended asset. Unfortunately, in a high intensity conflict, the air and missile defense site can expect to encounter multiple salvos of incoming TBMs. The defender must consider how many interceptors to fire at the current salvo while anticipating future attacks. The presence of a multi-salvo missile defense situation changes the problem from a static weapon-target assignment problem to a dynamic weapon-target assignment problem. Complicating the problem further is the consideration of multiple air and missile defense batteries having overlapping target coverage and the ability to engage the same set of incoming TBMs with differing probabilities of kill.

Previous work examines situations concerning the location of integrated air and missile defense systems assets (e.g., Boardman et al., 2017; Han et al., 2016) and the control of such assets in a multi-salvo engagement (e.g., Davis et al., 2017). However, such work assumes air and missile defense systems operate in parallel (i.e., they are capable of engaging the same targets at the same time). This assumption is somewhat unrealistic due to the limited number of air defense batteries available for asset coverage. There are very few assets, if any, that would be defended by multiple air defense systems of the same type. However, it is possible for an asset to be defended by multiple air defense systems via a sequence of engagements at different segments within the TBM's trajectory. This defense-in-depth strategy assumes the individual air and missile defense systems operate in series when engaging TBMs. For example, an Aegis may have the ability to engage a TBM during the mid-course segment, and a THAAD or Patriot system may have the ability to engage the same TBM (if not previously intercepted) during the terminal phase, at a later point in its trajectory.

Due to the extremely high speed of TBMs, the air defense community generally adopts a shoot-shoot-look policy in the terminal phase (National Research Council, 2012). This policy allows air defense assets to fire two interceptors at an incoming missile before it penetrates the defended assets' "keep out zone." Such a shoot-shoot-look policy increases the probability of a kill, but it is much more resource intensive than the alternative shoot-look-shoot policy, wherein the decision maker fires one interceptor, assesses the resulting battle damage, and then, as necessary, fires a second interceptor (Glazebrook and Washburn, 2004).

This research presents two ways to model and examine the networked, defense-in-depth air and missile defense problem and examines several related questions of interest. First, it sets forth a Markov decision process (MDP) model wherein a defender makes sequential engagement decisions upon encountering a salvo of incoming TBMs, initially by a forward-deployed air and missile defense asset (during the mid-course segment), and later when another air defense asset encounters the salvo (during the terminal segment). This model allows the system to determine the optimal firing solution over an uncertain number of decision epochs representing a sequence of salvos. If the layers of air defense assets fail to destroy all TBMs in an incoming salvo, the TBMs will decrease the defended assets' health with a specified probability of hit. Although formulating and solving this MDP provides a framework for identifying an optimal solution, doing so for practically-sized problem instances is computationally intractable.

As a second way to address this problem, we utilize approximate dynamic programming (ADP) and develop intercept policies based on two approximation algorithms. This approach attains solutions to larger problem instances while addressing dimensionality issues that render an enumerative approach to an MDP untenable. To leverage ADP and develop engagement policies, we employ a least squares policy evaluation (LSPE) ADP approach and a least squares temporal difference (LSTD) ADP approach. We compare the ADP solutions for sets of test instances to solutions attained via each of two "closed loop" policies based on current doctrine. We investigate the policy of shooting one interceptor at each incoming TBM and the policy of shooting two interceptors at each incoming TBM, independent of the state of the air defense system. The research presented herein describes a methodological contribution regarding the development of modeling and solution approaches for this networked, defense-in-depth air and missile defense problem. The comprehensive design, development, testing, and comparison of two heretofore unexamined ADP solution approaches (i.e., LSPE and LSTD) for this class of defensive, asset-based dynamic weapon-target assignment problem represent novel contributions to the related literature.

The remainder of this paper is organized as follows. Section 2 presents a literature review of the dynamic weapon target assignment problem and ADP. Section 3 offers a more extensive description of the networked air and missile defense problem, and it presents the MDP model formulation as well as the two ADP solution approaches. Section 4 presents and describes the findings when applying the aforementioned methodology to designed experiments, and Section 5 provides conclusions and suggested future research efforts.

2. Literature review

Two areas of literature inform the development and analysis of the networked, defense-in-depth air and missile problem. The first area concerns the weapon-target assignment problem (WTAP), and the second area involves approximate dynamic programming (ADP).

2.1. WTAP

The WTAP dates back to the 1950s when (Manne, 1958) developed a linear programming approximation to solve the problem. He noted that simultaneous decisions are unrealistic for military applications and proposed modeling the problem in a sequential manner. This distinction led to the development of two primary classes of the WTAP: static and dynamic.

Xin et al. (2011) describe the classes as follows. In a static WTAP, all targets are known, and all weapons are assigned to the targets in a single stage. In a dynamic WTAP, the decisions occur over many sequential stages, so weapons are assigned to the currently known targets at a given decision point, after which a new set of targets is presented.

2.2. Static WTAP

The static WTAP investigates the assignment of weapons to targets without regard to the impact of time. Consider the following situation as a simple motivating example. Suppose there are 10 tanks and 15 anti-tank teams respectively representing the targets and weapons. In this class of WTAP, the battle manager (i.e., defender) selects the weapon-target assignment decision that maximizes the expected value of the destroyed tanks based on each anti-tank weapon's associated probability of kill. Before the advent of modern computers, problems like this proved difficult and time consuming to solve. Today linear programming models and either exact or heuristic algorithms can solve large-scale instances of the static WTAP rather easily. Although interesting for some application instances, this class of problem does not represent the level of detail required for realistic air defense problems. Indeed, very few situations exist in which an air defense battle manager would have the ability to consider all incoming theater ballistic missiles (TBMs) at one point in time and assign interceptors to maximize a selected optimality criterion. Instead, a battle manager will likely observe a single incoming salvo of TBMs and make a decision regarding how many interceptors to fire at the incoming salvo while knowing only that future salvos are likely. The number and size of incoming salvos can be informed by knowing the phase of a conflict in which a decision is being made and having intelligence on how many TBMs the enemy has placed within range of the defended asset. Knowing this information and seeking to formulate a more realistic problem class motivates the consideration of the dynamic WTAP.

2.3. Dynamic WTAP

Similar to the static WTAP, the dynamic class seeks to assign weapons to targets in the most effective manner to ensure the highest probability of a kill, the greatest decremented value of the target, or the least decremented value of defended assets. The dynamic class of WTAP differs from the static class by the sequential nature of decisions, iteratively occurring as more information becomes available to the defender. Consider the tank example described in the previous section. In a dynamic WTAP, the battle manager does not consider the simultaneous engagement of all 10 tanks. Instead, the battle manager might observe a group of five tanks and assign some number of the anti-tank weapons to those five tanks, knowing future tank sightings are likely. Once weapons are assigned to targets, the battle manager proceeds to the next decision epoch to observe another group of tanks and make another assignment decision. The dynamic WTAP enables a more realistic representation of combat decision making under uncertainty. However, it also renders the problem far more complex and, with each additional level of complexity, the problem becomes more computationally challenging to solve optimally.

Heuristic methods (e.g., see Hosein et al., 1988; Xin et al., 2010) can solve the dynamic WTAP. Although such methods may not identify optimal solutions to the problems they solve, this outcome is common due to both the nature of heuristics and the complexity of the problem. Genetic algorithms such as the Anytime Algorithm (Wu et al., 2008) can also solve these problems. In this algorithm, the solution evolves over time as more information becomes available. The incumbent solution improves gradually but always maintains a reasonable and feasible decision, ready for implementation. Alternatively, Karasakal (2008) formulate an integer linear program to solve the problem and consider both point and area defense strategies, but their work assumes a shoot-look-shoot policy that constrains the action space, preventing the identification of superior solutions. Chang et al. (2018) develop and test an improved artificial bee colony algorithm with rule-based heuristic factors for initialization. The authors conduct extensive computational experiments to examine the performance of their algorithm, investigating the impact of different problem features (e.g., numbers of weapons and targets) and algorithm features (e.g., heuristic factors) on performance.

Several works examine dynamic WTAP variants utilizing an ADP approach. Authors adopt such an approach to solve a dynamic WTAP because the *curse of dimensionality* renders determination of an optimal policy via classical dynamic programming approaches (e.g., value iteration and policy iteration) computationally intractable. The curse of dimensionality refers to the difficulty of identifying an optimal policy, which requires the enumeration of all states (assumed to be discrete), and which is problematic if the state variable is a vector (Powell, 2019). Indeed, when a state space is multi-dimensional, the size of the state space increases exponentially in the number of dimensions. In addition to a large state space, the respective sizes of the action and outcome spaces also impose computational challenges (Powell, 2019).

Bertsekas et al. (2000) consider a dynamic WTAP variant wherein a defender must decide how many weapons to assign to each target in the current attack salvo and how many to retain for use against successive salvos. The authors leverage an ADP approach to determine suboptimal, yet high-quality, solutions to the problem, developing four policies to approximate the solution to the dynamic WTAP. Davis et al. (2017) address the dynamic WTAP from the defender's perspective, considering a smart attacker who knows the outcome of each salvo and fires appropriately at surviving targets. The authors allow an overlapping of each air and missile defense site's coverage area so two surface-to-air missile (SAM) sites can defend one asset. This problem feature enables the investigation of optimal firing policies when one SAM site is low on interceptors and another is not, or when the relative values of defended assets differ. Their problem instance of interest is small enough to find an exact solution to the MDP model, and the authors also investigate the quality of ADP approaches. Gulpinar et al. (2018) formulate a stochastic dynamic task-resource allocation problem with retry opportunities, which generalizes many variants of the dynamic WTAP. The authors develop and test a constructive heuristic that sequentially assigns resources (e.g., interceptors) to tasks (e.g., incoming missiles). They examine the solution quality and computational efficiency of their heuristic, both alone and within an ADP algorithmic framework, over several problem instances of increasing size. Results indicate

their heuristic performs well with respect to both quality and efficiency, scaling well with respect to problem size. Compared to the aforementioned works, our paper considers a different dynamic WTAP variant and applies different ADP solution approaches.

A full survey is beyond the scope of this work. Davis et al. (2017) and Kline et al. (2019) provide extensive reviews of the recent WTAP literature for an interested reader.

2.4. ADP

Assigning interceptors to missiles in a dynamic WTAP is a decision process performed under uncertainty. Formulating an MDP model allows us to determine the optimal decision now (i.e., for the current salvo) while accounting for the uncertain future salvos. Unfortunately, because of the curse of dimensionality, we are unable to quickly determine an optimal solution for large-sized problems of this class. Therefore, for the dynamic WTAP we employ ADP techniques. Powell (2009) provides a thorough starting point for ADP procedures. Earlier works include books by Bertsekas and Tsitsiklis (1996) and Sutton and Barto (1998).

We can attain solutions using either of two different algorithmic approaches: approximate value iteration and approximate policy iteration (API). For the particular dynamic WTAP variant we examine herein, we utilize an API algorithmic strategy to map the system state (i.e., incoming salvo make up, asset health, and interceptor inventory) to the action (i.e., how many interceptors to fire at each missile) to minimize the expected value of the defender's destroyed assets.

Powell (2012) describes four different policies for solving an ADP. The first policy he addresses is a myopic cost function wherein the defender attempts to minimize damage for just one decision epoch. Next, Powell describes a look-ahead policy wherein the defender would plan decisions over a set number of epochs, but only take the prescribed action for the current period. Some problems benefit from using policy function approximations such as look-up-tables, neural networks, or linear regression. For the DWTAP, a defender might have the policy that utilizes a shoot-shoot-look policy when it is above a prescribed threshold interceptor inventory, and otherwise uses a shoot-look-shoot policy. The final policy (Powell, 2012) describes is based on value function approximations. For our problem, we utilize a value function approximation scheme, adopting a basis function approach to determine the value of the post-decision state. Van Roy et al. (1997) introduces a modified Bellman's equation that utilizes the post-decision decision state to reduce the outcome space, making large-scale problems easier to solve. As we conduct the policy evaluation part of our API algorithm in this research, we update the value function approximation using either least squares policy evaluation (LSPE) or least squares temporal difference (LSTD) learning. Bradtke and Barto (1996) showed LSTD to be an efficient algorithm to find an approximate solution to a fixed policy. Lagoudakis and Parr (2003) advanced this method as they investigated the interactions of state and action pairs. For both LSPE and LSTD, we utilize an algorithmic structure similar to those set forth in Rettke et al. (2016), Davis et al. (2017), and Jenkins et al. (2019).

3. Methodology

This section sets forth the Markov decision process (MDP) model formulation of the dynamic weapon target assignment problem (DWTAP) and provides the mathematical underpinning for the approximate dynamic programming (ADP) algorithm discussed later in this section.

3.1. Problem description

Theater ballistic missiles (TBMs) and cruise missiles (CMs) present an extremely dangerous threat to United States and allied forces in the early stages of a conflict. Although efforts may attempt to destroy enemy TBM and CM stockpiles and capabilities before moving friendly forces into an operational area, the United States cannot expect to completely prevent an adversary's use of such weapons. The United States does have several options to defend assets from TBMs and CMs, but such protective systems are available in relatively small quantities. This limitation forces the United States to leave some assets unprotected and nearly guarantees each defended asset will be protected in the terminal phase of the missile trajectory by only one air defense asset.

Over the past several years, the United States Army has worked on developing a networked air defense capability to allow available air defense assets to operate in concert, providing defense-in-depth as a TBM or CM moves through a protected area. This capability gives the defender several decisions to make when developing an air defense plan, including which assets to defend, what type of air defense system to use, how many interceptors to provide each air defense site, how many interceptors to fire at a given salvo, and what firing policy to utilize.

In our problem of interest, the defender must protect a set of assets. Co-located with each asset is an air defense system (i.e., surface-to-air missile (SAM) site) to provide terminal phase protection. Located closer to the enemy launch site is an additional air defense system providing mid-course protection. Each friendly asset has an associated value and health state. If an incoming missile hits a friendly asset, it decrements the asset's health state. Each SAM site has a predetermined, limited number of interceptors for the engagement; this work does not consider replenishment of interceptors. The destruction of a defended asset, but not the co-located air defense asset, occurs if its health state decreases to zero. The attacker has predetermined numbers of different types of TBMs that are fired in salvos. The number of salvos is uncertain from the perspective of the defender. Moreover, the attacker does not know if its previous salvos successfully destroyed the defended assets, so it can continue to fire missiles at a completely destroyed asset. For this research, the attacker's types of TBMs include a traditional TBM and a TBM with multiple reentry vehicles (MeRV). During an attack, the defender must decide how many interceptors to fire from the mid-course air defense system and, simultaneously, if it had declined to fire or if the interceptors missed, it must also decide how many interceptors to fire from the terminal phase defense systems. If the salvo contains a MeRV TBM that successfully penetrates the mid-course defense system, this TBM will split into multiple missiles (targets). The defender seeks a policy to minimize the expected loss in value of its protected assets after all incoming salvos.

3.2. MDP model formulation

The formulation of the MDP model for this DWTAP variant follows. Let $\mathcal{T} = \{1, 2, \ldots, T\}, T \leq \infty$ be the set of decision epochs. The number of decision epochs T is random and follows a geometric distribution with parameter $\gamma \in [0, 1)$.

The state space consists of three components: the status of each asset, the inventory of each SAM site, and the number of TBMs in each SAM's area of responsibility.

The asset status component is defined as $A_t = (A_{ti})_{i \in \mathcal{A}} \equiv (A_{t1}, A_{t2}, \ldots, A_{t|\mathcal{A}|})$, wherein $\mathcal{A} = \{1, 2, \ldots, |\mathcal{A}|\}$ is the set of all assets, and $A_{ti} \in \{0, 0.25, 0.5, 0.75, 1\}$. A_{ti} is the health status of asset $i \in \mathcal{A}$ at decision epoch t and shows what percentage of the asset remains operational. Motivating this modeling choice is the desire to represent an asset status with more fidelity than simply 0,

1, which would roughly correspond to an asset being respectively "destroyed" or "undamaged." The 0.25, 0.5, and 0.75 statuses respectively relate to characterizations of "mostly destroyed," "heavily damaged," and "minorly damaged" in practice.

The SAM inventory status is defined as $R_t = (R_{ti})_{i\in\mathcal{A}} \equiv (R_{t1}, R_{t2}, \ldots, R_{t|\mathcal{A}|})$, where $R_{ti} \in \{0, 1, \ldots, r_i\}$, and r_i = initial inventory of interceptors at SAM site $i \in \mathcal{A}$. R_{ti} is the number of interceptors at SAM site $i \in \mathcal{A}$ at decision epoch t.

Let $\hat{\mathcal{M}}_{tj} = \{0, 1, 2, \ldots, |\hat{\mathcal{M}}_{tj}|\}$ be the set of all inflight attacker missiles of type $j \in \mathcal{J}$ at decision epoch t, where $\mathcal{J}$ is the set of all TBM types an attacker can fire, crossed with the set of two missile defense segments. For example, $j \in \mathcal{J}$ indicates whether the missile is a traditional TBM or a MeRV and its location – midcourse segment or terminal segment. $\hat{\mathcal{M}}_{tj}$ is the collection of observed incoming TBMs of type $j \in \mathcal{J}$ a defender must target at time t. The attack salvo is expressed as $\hat{M}_t = (\hat{\mathcal{M}}_{tji})_{j\in\mathcal{J}, i\in\mathcal{A}}$, wherein $\hat{\mathcal{M}}_{tji} \subseteq \hat{\mathcal{M}}_{tj}$ is the set of missiles of type $j \in \mathcal{J}$ targeting asset $i \in \mathcal{A}$ at decision epoch t. The attack salvo information in $\hat{M}_t$ is available to the defender at time t.

Using these components, we define $S_t = (A_t, R_t, \hat{M}_t) \in \mathcal{S}$ as the state of the system at decision epoch t, wherein $\mathcal{S}$ is the set of all possible states.

At each epoch t, the defender must decide how many interceptors to assign to each TBM from among the SAM sites having the given asset within their respective protection radii. We can deduce a coverage matrix for the entire defended area from the *a priori* placement of SAM sites relative to the asset locations. From this coverage matrix, we can determine which SAM sites can intercept each incoming missile. Let $x_{tijk} \in \mathbb{N}_0$ be the number of interceptors fired by SAM site $i \in \mathcal{A}$ against missile $k \in \hat{\mathcal{M}}^A_{tij}$ at decision epoch t, where $\hat{\mathcal{M}}^A_{tij}$ is defined as the set of missiles of type $j \in \mathcal{J}$ that SAM site i can intercept at decision epoch t. Let $x_t = (x_{tijk})_{i\in\mathcal{A}, j\in\mathcal{J}, k\in\hat{\mathcal{M}}^A_{tij}}$ denote our decision vector. We define the set of all feasible defender actions (i.e., assignment of interceptors to missiles) as

$$\mathcal{X}_{S_t} = \left\{ x_t : \sum_{j\in\mathcal{J}} \sum_{k\in\hat{\mathcal{M}}^A_{tij}} x_{tijk} \le \min\{R_{ti}, x^{max}\},\ \forall\ i \in \mathcal{A} \right\},$$

wherein the constraint

$$\sum_{j\in\mathcal{J}} \sum_{k\in\hat{\mathcal{M}}^A_{tij}} x_{tijk} \le \min\{R_{ti}, x^{max}\}$$

ensures each SAM site $i \in \mathcal{A}$ cannot fire more interceptors than either its number of interceptors in inventory or $x^{\max}$, the maximum number of interceptors a single SAM site can simultaneously fire due to target tracking radar performance limitations.

The transition functions explain how the system evolves as new information becomes known. We define the asset status transition function as

$$A_{t+1,i} = \begin{cases} 0 & \text{if } A_{ti} = 0, \\ \hat{A}_{t+1,i}(x_t) & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{A}, \text{ wherein } \hat{A}_{t+1,i}(x_t) \text{ is a}$$

random variable representing the status of each asset $i \in \mathcal{A}$ after evolution of salvo $\hat{M}_t$, given the interceptor assignment decision x_t. This information depends on x_t because the number of interceptors fired at the inbound TBMs affects an asset's health status.

We define the inventory status transition function as

$R_{t+1,i} = R_{ti} - \sum_{j\in\mathcal{J}} \sum_{k\in\hat{\mathcal{M}}^A_{tij}} x_{tijk},\ \forall\ i \in \mathcal{A}$, and note the asset status transition function is stochastic, whereas the inventory status transition function is deterministic because there is no probability associated with firing the interceptor – once the decision to fire the interceptor is made, we reduce the inventory. Regarding the transition of the attacker salvo status, $\hat{M}_{t+1}(x_t)$ is a random variable representing the status of each incoming TBM given the interceptor assignment decision x_t. This information depends on x_t because the number of interceptors fired at the inbound TBMs impacts their status.

The state transition function is defined as $S_{t+1} = S^M(S_t, x_t, W_{t+1})$, wherein $W_{t+1} = (\hat{A}_{t+1}, \hat{M}_{t+1})$. The random variable W_{t+1} represents all the information (i.e., asset status and attacker salvo) that becomes known at decision epoch $t+1$.

At each decision epoch t, the defender incurs an uncertain, immediate cost as a result of its decision. We define this cost as

$$\hat{C}(S_t, x_t, \hat{A}_{t+1,i}) = \sum_{i\in\mathcal{A}} v_i(A_{ti} - \hat{A}_{t+1,i}),$$

wherein v_i is the value of asset $i \in \mathcal{A}$. We rewrite the cost function in terms of only the current state and decision by taking its expected value

$$C(S_t, x_t) = \mathbb{E}\left[\sum_{i\in\mathcal{A}} v_i(A_{ti} - \hat{A}_{t+1,i}) | S_t, x_t \right].$$

We seek to identify the policy that minimizes the expected total cost (i.e, loss). That is, the defender is trying to maintain as much value as possible in its protected assets. We let $X^\pi(S_t)$ denote a decision function (i.e., policy) that prescribes defender interceptor decisions for each state $S_t \in \mathcal{S}$. We want to determine the optimal policy π^* from the class of policies $(X^\pi(S_t))_{\pi\ \in\ \Pi}$. The objective is

$$\min_{\pi\in\Pi} \mathbb{E}^\pi \left[\mathbb{E}^T \left[\sum_{t=1}^{T} C(S_t, X^\pi(S_t)) \right] \right],$$

wherein $\mathbb{E}^\pi$ indicates the expectation depends on the defender's selected policy π, and $\mathbb{E}^T$ indicates the expectation depends on the horizon length T. The defender does not know the number of missile salvos against which it needs to defend, rendering this DWTAP more challenging than a static variant. The defender must make interceptor assignment decisions when the number of incoming salvos of TBMs is uncertain. Puterman (1994) shows the following equivalent objective may be utilized when there is a geometrically distributed horizon length

$$\min_{\pi\in\Pi} \mathbb{E}^\pi \left[\sum_{t=1}^{\infty} \gamma^{t-1} C(S_t, X^\pi(S_t)) \right].$$

In this formulation, the defender values policies according to the expected total cost optimality criterion, but the number of decision epochs T is uncertain. The parameter γ acts as a discount factor and models the uncertain number of TBM salvos fired by the attacker. We note the following relationship between γ and the expected number of salvos:

$$\mathbb{E}[T] = \frac{1}{1-\gamma}.$$

Because of intelligence estimates concerning the number of missile launchers available to the attacker as well as the attacker's doctrine concerning the employment of such weapons, it is reasonable to assume the defender can probabilistically describe both the numbers and sizes of salvos expected to be fired by the attacker for a particular conflict scenario.

To determine an optimal policy, we must find a solution to the Bellman Equation

$$J(S_t) = \min_{x_t\in\mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma \mathbb{E}[J(S_{t+1}) | S_t, x_t]), \tag{1}$$

utilizing the decision function (i.e., policy)

$$X^\pi(S_t) = \arg\min_{x_t\in\mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma \mathbb{E}[J(S_{t+1}) | S_t, x_t]).$$

3.3. ADP formulation

Although the MDP model enables the determination of an exact solution to the DWTAP, it is only computationally tractable for small-scale problems. In any instance of interest to the air defense community, the problem quickly becomes too large to solve optimally. For example, consider the size of the state space $\mathcal{S}$, where $S_t = (A_t, R_t, \hat{M}_t) \in \mathcal{S}$ is an arbitrary state. The tuples A_t, R_t, and $\hat{M}_t$ respectively represent the status of each asset, the status of each SAM site's inventory, and the attacker TBMs at decision epoch t. Since asset status can be $0, 0.25, \ldots, 1$, there are $5^{|\mathcal{A}|}$ possible states for A_t. The different SAM sites have different maximum inventories, but if they each had a maximum of 12 interceptors, then there are $13^{|\mathcal{R}|}$ possible states for R_t. If M is the maximum number of attacker missiles that can be located in any SAM's area of responsibility at any epoch t, then there are $\binom{|A|+M}{M}$ possible states for $\hat{M}_t$. This means an instance of this problem with 3 SAM sites, a maximum of 12 interceptors per site, and 12 points where missiles can be located corresponds to a state space having almost one billion different states. The computational effort necessary to exhaustively enumerate a state space of this size to find the exact solution is prohibitive. Moreover, the DWTAP suffers from the curse of dimensionality with regard to the action space as well as the state space. Since the defender can choose to fire zero, one, or two interceptors at each TBM in a corresponding SAM site's area of responsibility, the number of feasible actions can increase into the millions with only 14 available firing points. Although the state space for the DWTAP does not immediately preclude the use of enumeration methods to identify an optimal solution, its combination with the large action space renders the problem computationally intractable.

ADP offers solution strategies to address both of the aforementioned issues. The approximate policy iteration (API) algorithmic strategy is a simulation-based solution procedure that approximates solutions based on Eq. (1). Therefore, we rewrite the Bellman Equation and use the post-decision state variable convention. Letting $J^x(S_t^x)$ be the value of being in post-decision state S_t^x, we can show the relationship between $J(S_t)$ and $J^x(S_t^x)$ via the following equations

$$J(S_t) = \min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)), \tag{2}$$

$$J^x(S_t^x) = \mathbb{E}[J(S_{t+1}) | S_t^x]. \tag{3}$$

By substituting Eq. (2) into the definition of $J^x(S_{t-1}^x)$ in accordance with Eq. (3), we obtain the Bellman Equation around the post-decision state variable

$$J^x(S_{t-1}^x) = \mathbb{E}\left[\min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)) \middle| S_{t-1}^x\right]. \tag{4}$$

Using the post-decision state form instead of the standard form of the Bellman Equation requires the swapping of the expectation and minimum operators and allows us to avoid approximating the expectation inside of the optimization problem. This solution approximating decision allows us to control the structure of the problem and leverage statistical approximation techniques.

With ADP we will simulate forward in time and solve the problem approximately instead of enumerating the entire state space to solve the problem exactly (e.g., using backward induction). We are able to randomly choose a pre-decision state S_t and subsequently make a decision x_t to move to the post-decision state S_t^x.

ADP techniques can address large state spaces, but we still must contend with approximating the expectation. We can construct a *post-decision state* variable to avoid this approximation. Van Roy et al. (1997) first used this term, and Powell (2011) defines the post-decision state variable as the state of the system immediately after a decision x_t is made but before any new information W_{t+1} arrives. Now the state transition function $S_{t+1} = S^M(S_t, x_t, W_{t+1})$ can be decomposed into the following two steps

$$S_t^x = S^{M,x}(S_t, x_t), \tag{5}$$

and

$$S_{t+1} = S^{M,W}(S_t^x, W_{t+1}), \tag{6}$$

wherein S_t^x is the post-decision state variable, and S_{t+1} is the next pre-decision state variable. For this DWTAP, the post-decision state is given by $S_t^x = (A_t^x, R_t^x)$, wherein $A_t^x = (A_{ti}^x)_{i \in \mathcal{A}}$ is the component concerning asset status and $R_t^x = (R_{ti}^x)_{i \in \mathcal{A}}$ is the component concerning interceptor inventory status.

Value function approximation

Value function approximation occurs via regression methods. Similar to linear regression wherein we seek to find a vector using observations to fit a model that will predict a new, unknown observation using a set of variables, value function approximation seeks to identify a parameter vector θ using observations created from a set of basis functions $(\phi_f(S_t))_{f \in \mathcal{F}}$. The set $\mathcal{F}$ of basis functions reduces the size of the state variable to those factors with which we are most concerned. For example, a basis function $f \in \mathcal{F}$ for our problem might be the remaining value of a defended asset. Using the post-decision state, we write our value function in a manner similar to that used in linear regression

$$\bar{J}^x(S_t^x) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t^x). \tag{7}$$

The approximate Bellman Equation utilizes the value function approximation $\bar{J}^x(S_t^x)$ and is expressed as

$$\bar{J}^x(S_{t-1}^x) = \mathbb{E}\left[\min_{x_t \in \mathcal{X}_{S_t}} \left(C(S_t, x_t) + \gamma \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t^x)\right) \middle| S_{t-1}^x\right]. \tag{8}$$

The decision function (i.e., ADP policy) is expressed as

$$X^{\pi_{adp}}(S_t | \theta) = \arg\min_{x_t \in \mathcal{X}_{S_t}} \left(C(S_t, x_t) + \gamma \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t^x)\right). \tag{9}$$

Algorithmic strategy variants

API executes a sequence of inner loops to evaluate a fixed policy, embedded within an outer loop that generates an improving sequence of approximate policies. Algorithm 1 shows the application of API-LSPE to solve the air defense problem. The least squares policy evaluation (LSPE) algorithm consists of K policy evaluation loops and N policy improvement loops. Within the policy evaluation loop, we obtain a collection of K value and post-decision state pairs and use least squares regression to fit a linear model to approximate a value function. Policy improvement occurs by updating the θ-vector after each policy evaluation (i.e., inner) loop completes and using the updated θ-vector to better approximate the value function in the next policy improvement (i.e., outer) loop iteration. Each time the algorithm finishes an inner loop, it updates the θ-vector and performs another iteration of the outer loop to seek further improvement.

The policy improvement update occurs after collecting the Kth policy evaluation sample realization. We can describe the vectors of basis function evaluations and the vectors of costs in the following manner. Let

$$\Phi_{t-1} \triangleq \begin{bmatrix} \phi(S_{t-1,1}^x)^\top \\ \vdots \\ \phi(S_{t-1,K}^x)^\top \end{bmatrix} \text{ and } \quad C_t \triangleq \begin{bmatrix} C(S_{t,1}) \\ \vdots \\ C(S_{t,K}) \end{bmatrix},$$

where matrix Φ_{t-1} contains rows of basis function evaluations of the sampled post-decision states, and C_t is the cost vector. We perform a least squares regression of Φ_{t-1} against C_t. We update our estimate of θ using $\alpha_n = \frac{a}{a+n-1}, a \in (0, \infty)$ as our smoothing function. The generalized harmonic smoothing function manages the rate at which the θ-vector converges. Higher a-values slow the rate at which α_n drops to zero, allowing later outer loop iterations to have a greater impact on the θ-vector. Smoothing θ completes one policy improvement step.

Algorithm 2 shows the application of API-LSTD to solve the air defense problem. As with LSPE, the least squares temporal differences (LSTD) algorithm consists of K policy evaluation loops and N policy improvement loops. After initializing a θ-vector as the representation of an initial policy, the policy evaluation loop begins by generating a random post-decision state. Upon recording the value $\phi(S^x_{t-1,k})$, we simulate forward to the next pre-decision state and select the best decision using exhaustive enumeration. We record the cost $C(S_{t,k}, x_{t,k})$ and basis function evaluations of the post-decision state, $\phi(S^x_{t,k})$. We obtain K temporal difference sample realizations where the kth temporal difference given the parameter vector θ^n is $(C(S_{t,k}, x_{t,k}) + \gamma\phi(S^x_{t,k})^T\theta^n) - \phi(S^x_{t,k-1})^T\theta^n$.

The policy improvement loop occurs after collecting the Kth temporal difference sample realization. We describe the vectors of basis function evaluations and the vectors of costs in the following manner. Let

$$\Phi_{t-1} \triangleq \begin{bmatrix} \phi(S^x_{t-1,1})^\top \\ \vdots \\ \phi(S^x_{t-1,K})^\top \end{bmatrix}, \quad \Phi_t \triangleq \begin{bmatrix} \phi(S^x_{t,1})^\top \\ \vdots \\ \phi(S^x_{t,K})^\top \end{bmatrix}, \quad \text{and} \quad C_t \triangleq \begin{bmatrix} C(S_{t,1}) \\ \vdots \\ C(S_{t,K}) \end{bmatrix},$$

where matrices Φ_{t-1} and Φ_t contain rows of basis function evaluations of the sampled post-decision states, and C_t is the cost vector. We compute $\hat{\theta}$ by performing a least squares regression of Φ_{t-1} and Φ_t against C_t, seeking to ensure the sum of the K temporal differences tends toward zero. As in LSPE, we update our estimate of θ using $\alpha_n = \frac{a}{a+n-1}, a \in (0, \infty)$ as our smoothing function. Smoothing θ completes one policy improvement step.

The computational complexity of both algorithms is $O(NK(|\mathcal{F}|^2 + |\mathcal{X}|))$, where N indicates the total number of regressions performed, K indicates the number of samples for each regression, $|\mathcal{F}|$ indicates the number of variables for each regression, and $|\mathcal{X}|$ represents the maximum size of the decision space. Although important in a general sense, computational efficiency is not a primary concern in this work. This research focuses on a comparison of algorithm effectiveness (and policy insights) rather than algorithmic efficiency. Moreover, the two ADP algorithm variants we investigate exhibit nearly identical running times. Once an ADP policy is determined off-line via a solution approach such as our variants, weapon-target assignment recommendations based on the policy can be computed nearly instantly and would be embedded within battle management decision support software.

4. Testing, results, and analysis

4.1. Computational experiments

In this section, we examine a problem of interest to the air defense community and utilize the approximate dynamic programming (ADP) techniques described in Section 3 to seek high quality solutions to this problem. We compare policies found by our ADP algorithms to current baseline policies used by the air defense community. We construct a theater ballistic missile (TBM) defense scenario as the tactical underpinning for our analysis. From this scenario, we create 32 test instances and, for each instance, determine approximate solutions for each baseline policy using simulation techniques. We solve each instance approximately by applying each of our ADP solution methodologies. A set of designed experiments identifies which ADP algorithmic parameter-values yield the best solutions. We conduct computational experiments for both least squares policy evaluation (LSPE) and least squares temporal differences (LSTD), and we compare the two ADP algorithms to determine the better overall ADP algorithm (and corresponding policy) for each of the 32 instances. We also compare the current air defense policies and the acquired ADP policies using vignettes of interest from earlier simulation-based experiments. We utilize a dual Intel Xeon E5-2650v2 workstation having 128 GB of RAM and MATLAB's Parallel Computing Toolbox to conduct the computational experiments and analyses herein.

4.1.1. Representative scenario

We present a networked TBM defense system utilizing the Missile Defense Agency's (MDA) defense-in-depth plan for a mid-course and terminal phase defense. For this scenario the defender seeks to protect two assets. This scenario places an Aegis air defense system in the mid-course segment, a THAAD with the first defended asset in the terminal segment, and a Patriot with the second defended asset in the terminal segment. See Fig. 1 for an illustration of this scenario. All TBMs pass through the Aegis's area of responsibility, and the Aegis has an opportunity to fire up to two interceptors at each TBM. We allocate 12 interceptors to the Aegis, which constitutes half of the payload of an Aegis-equipped ship (U.S. Missile Defense Agency, 2016b). We allocate 24 interceptors each to the THAAD and Patriot, based on the number of launchers typically operated by an active Patriot battery (e.g., six 4-missile M901 launchers) (Foss and O'Halloran, 2017; Gourley, 2011). The THAAD and the Patriot can only fire at TBMs targeting the assets they respectively defend, and they have the opportunity to fire up to two interceptors at each such TBM.

The attacker fires a combination of traditional TBMs and multiple reentry vehicle (MeRV) TBMs. Each type of TBM is equally likely to be fired for any one attacker missile salvo. Informing this presumed attacker targeting distribution would be defender intelligence estimates regarding the attacker's fielded missile forces. The attacker's missile firing policy (with respect to target, but not type) depends on the initial value of each asset. We assume the likelihood an attacker targets a given asset is in relative proportion to its initial value. For example, if a defender has two assets for which Assets 1 and 2 have values of 48 and 24, respectively, we assume the attacker targets Asset 1 with probability 2/3. The defender knows the type of TBM fired by the attacker.

If the Aegis does not target (or targets and misses) a MeRV TBM, it splits into three missiles before the THAAD or Patriot have an opportunity to fire at it. Any TBMs surviving potential engagements in the terminal segment of flight have a fixed probability of hitting their intended targets. This modeling feature captures the technical sophistication of the attacker's weaponry (e.g., flight control, guidance, and warhead technology). If the TBM hits the asset it is targeting, it decrements the asset by a preassigned amount of one quarter of the asset's total value; an attacker must hit a defender's asset four times to destroy it. The discount factor γ models the expected number of salvos the defender will encounter.

From this basic scenario, we develop 32 test instances by varying four of the problem features. We first varied the number of salvos the defender can expect to engage (i.e., or the duration of the attack), as indicated by γ. Exploratory simulations based on the number of available interceptors in each phase allowed us to choose two γ-values, 0.8 and 0.9, to investigate the impact the expected number of salvos had on the ADP policy. A value of $\gamma = 0.8$ corresponds to 5 expected salvos, whereas $\gamma = 0.9$ corresponds to 10 expected salvos.

The second problem feature we varied was the enemy's level of technological sophistication – that is, the enemy may have fairly

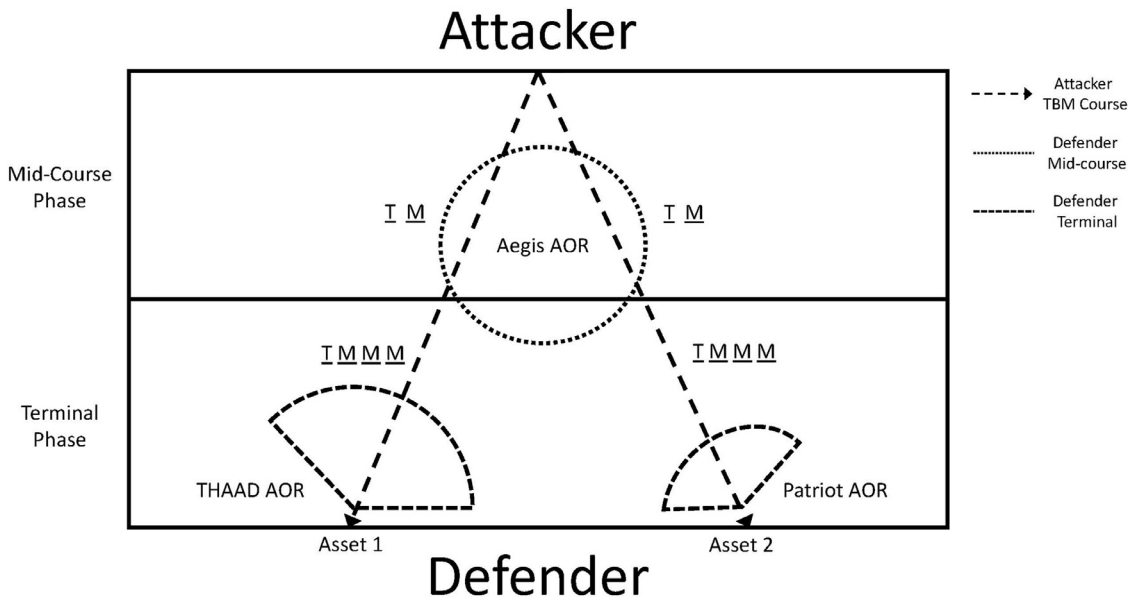

Fig. 1. Scenario Diagram.

Table 1
Test instance problem features.

Expected conflict duration	Attacker's technological sophistication	Defender's technological sophistication	Defended asset values
Short ($\mathbb{E}[T] = 5$) Long ($\mathbb{E}[T] = 10$)	Low (pH = 0.5) High (pH = 0.8)	Low (pK = 0.7, 0.8, 0.75) High (pK = 0.8, 0.9, 0.85)	Low/Low High/Low Low/High High/High

Table 2
Experimental design for Algorithmic features.

N	K	ϕ	a	η
25	1000	1	10	1
50	2000	2	100	10

accurate TBMs or inaccurate TBMs. We chose a probability of hit of 0.8 for the technologically superior attacker and a probability of hit of 0.5 for the technologically inferior attacker.

The third problem feature we varied was the defender's level of technological sophistication, seeking to capture the accuracy of the defender's interceptors to successfully engage the TBMs. We chose a probability of kill for the technologically superior defender of 0.8, 0.9, and 0.85 for the Aegis, THAAD and Patriot, respectively. We chose a probability of kill for the technologically inferior defender of 0.7, 0.8, and 0.75, for the Aegis, THAAD, and Patriot respectively.

The fourth problem feature we varied was the defended asset value. We sought to investigate how a higher, lower, or equal relative values of the defended assets protected by the THAAD and Patriot would affect the ADP policy. We assigned equal values of 24 for both Asset 1 and Asset 2 for the Low/Low case; values of 48 for Asset 1 and 24 for Asset 2 for the High/Low case; values of 24 for Asset 1 and 48 for Asset 2 in the Low/High case; and equal values of 48 for both Asset 1 and Asset 2 in the High/High case. Table 1 shows the problem feature settings utilized to construct the 32 test instances.

Algorithm 1 API-LSPE Algorithm.

Step 0: Initialize θ^0.
Step 1:
for n = 1 **do to** N **(Policy Improvement Loop)**
  Step 2:
  for k = 1 **do to** K **(Policy Evaluation Loop)**
    Generate a random post-decision state $S^x_{t-1,k}$.
    Record basis function evaluation $\phi(S^x_{t-1,k})$.
    Simulate transition to next pre-decision state $S_{t,k}$ using Equation (6).
    Determine decision $x_{t,k} = X^{\pi_{adp}}(S_{t,k}|\theta^{n-1})$ using Equations (5), (7), and (9).
    Record cost $C(S_{t,k}, x_{t,k})$.
  end for
  Update θ^n and the policy:
  $\hat{\theta} = [(\Phi_{t-1})^T(\Phi_{t-1})]^{-1}(\Phi_{t-1})^T C_t$
  $\theta^n = \alpha_n\hat{\theta} + (1-\alpha_n)\theta^{n-1}$
end for
Return $X^{\pi_{adp}}(\cdot|\theta^N)$ and θ^N.
End

Table 3
Basis function features.

ϕ	$\phi_0(S)$	$\phi_1(S)$	$\phi_2(S)$	$\phi_3(S)$	$\phi_4(S)$	$\phi_5(S)$
1	1	A_t	R^x_t	A^x_t		
2	1	R^x_t	A^x_t	$(R^x_t)^2$	$(A^x_t)^2$	$R^x_t A^x_t$

4.1.2. Experimental design

For each of the 32 test instances, we seek to determine the best parameter settings for Algorithms 1 and 2. We focus on parameters N, K, ϕ, a, and η. Table 2 shows the 2-level, 5-factor experimental design. Initial experimental runs of the model informed the selection of the levels for each factor. Table 3 shows the set of features for each design level of the ϕ-factor.

Algorithm 2 API-LSTD Algorithm.

Step 0: Initialize θ^0.
Step 1:
for n = 1 **do to** N **(Policy Improvement Loop)**
  Step 2:
  for k = 1 **do to** K **(Policy Evaluation Loop)**
    Generate a random post-decision state $S^x_{t-1,k}$.
    Record basis function evaluation $\phi(S^x_{t-1,k})$.
    Simulate transition to next pre-decision state $S_{t,k}$ using Equation (6).
    Determine decision $x_{t,k} = X^{\pi_{adp}}(S_{t,k}|\theta^{n-1})$ using Equations (5), (7), and (9).
    Record cost $C(S_{t,k}, x_{t,k})$.
    Record next post-decision state $S^x_{t,k}$ using decision $x_{t,k}$ and Equation (5).
    Record basis function evaluation $\phi(S^x_{t,k})$.
  end for
  Update θ^n and the policy:
  $\hat{\theta} = [(\Phi_{t-1} - \gamma\Phi_t)^T(\Phi_{t-1} - \gamma\Phi_t)]^{-1}(\Phi_{t-1} - \gamma\Phi_t)^T C_t$
  $\theta^n = \alpha_n\hat{\theta} + (1-\alpha_n)\theta^{n-1}$
end for
Return $X^{\pi_{adp}}(\cdot|\theta^N)$ and θ^N.
End

4.1.3. Experimental results

For each test instance, we ran a full factorial experiment for three random number seeds (i.e., three replications) for a total of 96 runs. For each run, we recorded the mean and standard deviation and calculated the difference between the ADP policy means and the means of the two baseline policies. For each scenario, we chose the ADP policy (and noted the attendant parameter settings) corresponding to the largest difference between the baseline policies and the ADP policy.

4.2. Quality of results

Utilizing the problem features and experimental design described in Section 4.1, we implemented the LSPE and LSTD algorithms respectively annotated in Algorithms 1 and 2. This required 3072 runs for each algorithm to perform the full factorial experiment for all problem and algorithmic features with three replications. The algorithms provided a θ-vector for each of these runs. We then utilized a simulation to determine the mean performance and standard deviation for each of those θ-vectors. We executed 2000 simulation runs for each θ-vector to improve the likelihood of the sample mean accurately approximating the true mean.

We compared the ADP results to two baseline policies. The first baseline policy (i.e., the Match Policy) fires one interceptor at each incoming TBM (as long as the SAM site inventory allows it), and the second baseline policy (i.e., the Overmatch Policy) fires two interceptors at each TBM (if the SAM site inventory does not allow for firing two, then the SAM will fire one, if available). We executed 2000 simulation runs for each of the two baseline policies. In exploratory runs of the simulation, we found that firing one interceptor at each incoming TBM generally outperformed firing two interceptors for the problem features being explored. We compared the means of our LSPE algorithms' policies to the two baseline policies. Table 4 reports the results for the best ADP policy versus the baseline polices for each scenario.

The LSPE policy achieves statistically significant improvement over the two baseline policies in 8 of the 32 test instances. Instances 1, 2, and 12 exhibit LSPE policy superiority at the 95% confidence level. Instances 9, 10, and 25–27 show statistical significance at the 90% confidence level. LSPE attains the best mean result in 14 of the 32 instances. We note that LSPE outperforms the baseline policies when the duration of the conflict is short or when the enemy has weapons with a high probability of hit. It is not surprising that, for circumstances in which an uninterrupted TBM has a high likelihood of damaging its targeted asset, the LSPE policy outperforms the baseline policies, but it is interesting that, in shorter duration conflicts when the two baseline policies show very similar means, the ADP policy performance is superlative at a statistically significant level.

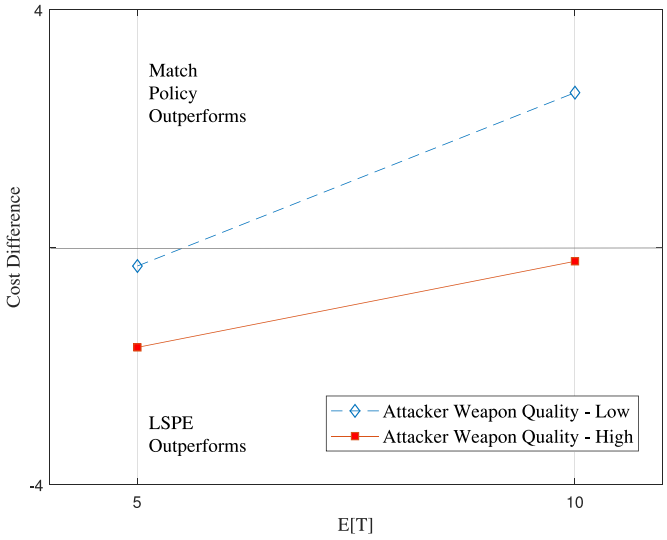

Fig. 2. Cost Difference Between LSPE and Match policies for Attacker Weapon Quality and Conflict Duration.

The Match Policy outperforms LSPE in Instances 21–24, 29, and 31 at the 95% confidence level. Examining these instances, we find common characteristics: long duration conflict wherein the attacker had lower quality weapons and the defender had higher quality weapons.

It is of further interest that the Overmatch Policy, which is currently the United States Army's implemented policy, is never significantly better than the LSPE policy or the Match Policy. This domination of the current policy by alternative policies suggests that, as the military moves to an integrated, defense-in-depth strategy, it should consider a different firing policy for networked air defense systems having both mid-course and terminal systems.

Fig. 2 illustrates the cost difference between the LSPE and Match policies when we examine the two different levels of attacker weapon quality for short and long duration conflicts. We observe that, for high quality attacker weapons, LSPE performs better than the Match Policy regardless of conflict duration, but for low quality attacker weapons, the Match Policy is superior for long duration conflicts. Of note, this graphic is not intended to imply a linear relationship that might not exist; the lines simply emphasize the respective, relative values.

Fig. 3 illustrates the cost difference between the LSPE and Match policies when we examine the two different levels of defender weapon quality for short and long duration conflicts. We observe that, for low quality defender weapons, LSPE performs better than Match regardless of conflict duration, but for high quality defender weapons, Match is always superior. Of note, this graphic does not imply a linear relationship between $\mathbb{E}[T]$ and the cost difference; the lines simply emphasize the respective, relative values.

Table 4
LSPE results - Quality of solution using best θ-vector.

Instance (γ, pH, pK, Asset Value)	Best Alg. parameters (N,K,a,η, ϕ)	ADP policy 95% CI	Match policy 95% CI	Overmatch policy 95% CI
1 (0.8, 0.5, 1, Low/Low)	25, 2000, 100, 1, 2	[a]**5.75 ± 0.71**	7.55 ± 0.64	7.59 ± 0.82
2 (0.8, 0.5, 1, High/Low)	25, 1000, 10, 10, 2	[a]**5.68 ± 0.71**	8.01 ± 0.68	7.97 ± 0.85
3 (0.8, 0.5, 1, Low/High)	25, 2000, 100, 10, 1	**6.09 ± 0.73**	7.42 ± 0.63	6.77 ± 0.78
4 (0.8, 0.5, 1, High/High)	25, 2000, 10, 1, 2	**6.05 ± 0.72**	7.55 ± 0.63	7.2 ± 0.8
5 (0.8, 0.5, 1, Low/Low)	25, 2000, 100, 1, 2	4.78 ± 0.67	**4.29 ± 0.52**	7.25 ± 0.84
6 (0.8, 0.5, 2, High/Low)	50, 2000, 100, 10, 2	4.71 ± 0.68	**4.43 ± 0.51**	7.41 ± 0.84
7 (0.8, 0.5, 2, Low/High)	25, 1000, 100, 1, 2	4.99 ± 0.7	**4.13 ± 0.46**	6.48 ± 0.78
8 (0.8, 0.5, 2, High/High)	50, 1000, 10, 10, 2	5.15 ± 0.71	**4.18 ± 0.5**	6.42 ± 0.78
9 (0.8, 0.8, 2, Low/Low)	50, 2000, 10, 1, 1	[b]**7.11 ± 0.8**	11.68 ± 0.84	8.53 ± 0.88
10 (0.8, 0.8, 1, High/Low)	25, 1000, 10, 1, 2	[b]**7.68 ± 0.83**	10.39 ± 0.77	9.15 ± 0.91
11 (0.8, 0.8, 1, Low/High)	50, 2000, 100, 10, 1	**7.49 ± 0.81**	10.8 ± 0.78	8.33 ± 0.85
12 (0.8, 0.8, 1, High/High)	25, 2000, 100, 1, 1	[a]**7.37 ± 0.81**	10.82 ± 0.79	9.36 ± 0.92
13 (0.8, 0.8, 2, Low/Low)	25, 2000, 10, 10, 2	6.07 ± 0.77	**5.79 ± 0.58**	7.98 ± 0.89
14 (0.8, 0.8, 2, High/Low)	50, 2000, 10, 10, 2	6.12 ± 0.79	**5.69 ± 0.58**	6.68 ± 0.81
15 (0.8, 0.8, 2, Low/High)	25, 1000, 10, 10, 1	5.97 ± 0.78	**5.94 ± 0.6**	8.65 ± 0.91
16 (0.8, 0.8, 2, High/High)	50, 2000, 100, 10, 2	**6.02 ± 0.76**	6.25 ± 0.62	7.31 ± 0.84
17 (0.9, 0.5, 1, Low/Low)	25, 1000, 100, 10, 1	21.11 ± 1.3	**20.88 ± 1.13**	24.46 ± 1.35
18 (0.9, 0.5, 1, High/Low)	25, 1000, 100, 10, 1	21.02 ± 1.29	**19.9 ± 1.12**	22.58 ± 1.32
19 (0.9, 0.5, 1, Low/High)	50, 1000, 100, 10, 2	20.27 ± 1.29	**20.17 ± 1.12**	23.6 ± 1.34
20 (0.9, 0.5, 1, High/High)	25, 2000, 100, 10, 1	**21.16 ± 1.3**	21.49 ± 1.15	23.4 ± 1.32
21 (0.9, 0.5, 1, Low/Low)	25, 1000, 100, 1, 2	19.46 ± 1.29	[c]**15.22 ± 1.06**	22.46 ± 1.34
22 (0.9, 0.5, 2, High/Low)	50, 2000, 10, 1, 1	19.55 ± 1.29	[c]**14.33 ± 1.02**	22.46 ± 1.35
23 (0.9, 0.5, 2, Low/High)	25, 2000, 10, 10, 2	19.6 ± 1.29	[c]**14.44 ± 1.04**	21.47 ± 1.31
24 (0.9, 0.5, 2, High/High)	25, 1000, 10, 1, 1	19.53 ± 1.28	[c]**14.5 ± 1.03**	21.6 ± 1.33
25 (0.9, 0.8, 2, Low/Low)	50, 1000, 10, 1, 2	[b]**22.97 ± 1.34**	25.83 ± 1.2	25.66 ± 1.38
26 (0.9, 0.8, 1, High/Low)	50, 1000, 100, 1, 1	[b]**23.34 ± 1.35**	26.98 ± 1.24	25.94 ± 1.38
27 (0.9, 0.8, 1, Low/High)	25, 2000, 100, 1, 1	[b]**22.85 ± 1.34**	25.16 ± 1.21	25.91 ± 1.38
28 (0.9, 0.8, 1, High/High)	50, 1000, 10, 1, 1	**23.15 ± 1.33**	25.76 ± 1.21	25 ± 1.37
29 (0.9, 0.8, 2, Low/Low)	25, 1000, 100, 10, 1	21.38 ± 1.34	[c]**17.95 ± 1.11**	24.19 ± 1.39
30 (0.9, 0.8, 2, High/Low)	50, 1000, 10, 10, 2	21.12 ± 1.32	**19.64 ± 1.15**	23.83 ± 1.38
31 (0.9, 0.8, 2, Low/High)	25, 1000, 10, 1, 2	20.88 ± 1.34	[c]**18.01 ± 1.12**	24.53 ± 1.4
32 (0.9, 0.8, 2, High/High)	50, 2000, 10, 1, 1	21.12 ± 1.34	**19.42 ± 1.17**	22.74 ± 1.36

[a] denotes statistical significance (as compared to the next best policy) with 95% confidence
[b] denotes statistical significance (as compared to the next best policy) with 90% confidence
[c] denotes statistical significance (as compared to the ADP policy) with 95% confidence⋆ denotes statistical significance (as compared to the ADP policy) with 90% confidence

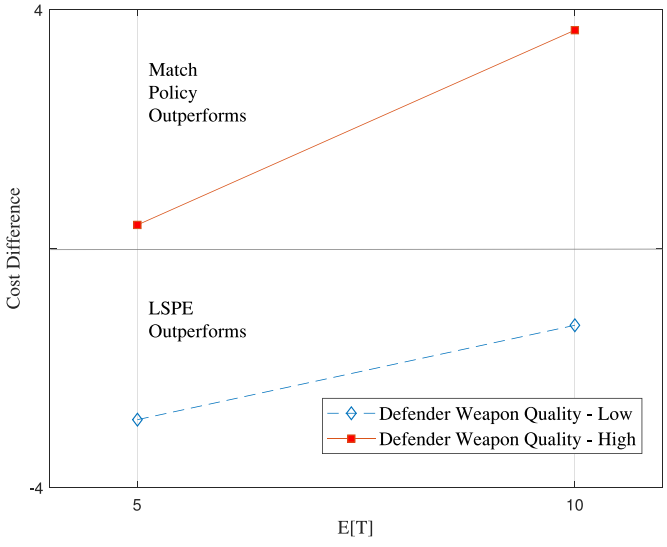

Fig. 3. Cost Difference Between LSPE and Match policies for Defender Weapon Quality and Conflict Duration.

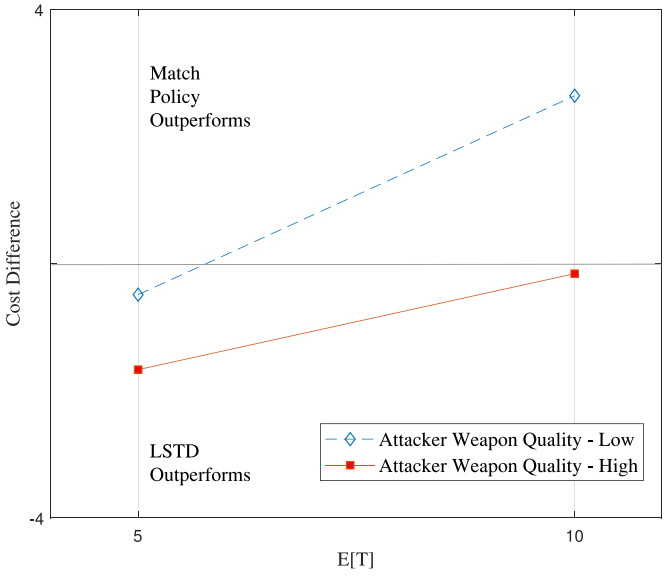

Fig. 4. Cost Difference Between LSTD and Match policies for Attacker Weapon Quality and Conflict Duration.

We proceed by examining the quality of the policies obtained by utilizing the second of our two ADP algorithms. Table 5 reports the results for the best ADP policy versus the baseline polices for each scenario. The LSTD policy achieves statistically significant improvement over the two baseline policies in 6 of the 32 test instances. Instance 2, 10, and 12 exhibit LSTD policy superiority at the 95% confidence level. Instances 1, 25, and 26 show statistical significance at the 90% confidence level. LSTD attains the best mean result in 15 of the 32 instances. LSTD outperforms the baseline policies when the duration of the conflict is short or when the enemy has weapons with a high probability of hit. Similar to the outcomes for LSPE, for circumstances in which the incoming

Table 5
LSTD results - Quality of solution using best θ-vector.

Instance (γ, pH, pK, Asset Value)	Best Alg. parameters (N,K,a,η, ϕ)	ADP policy 95% CI	Match policy 95% CI	Overmatch policy 95% CI
1 (0.8, 0.5, 1, Low/Low)	25, 2000, 10, 10, 2	[b]**6.08 ± 0.74**	7.55 ± 0.64	7.59 ± 0.82
2 (0.8, 0.5, 1, High/Low)	25, 2000, 10, 1, 2	[a]**6.15 ± 0.73**	8.01 ± 0.68	7.97 ± 0.85
3 (0.8, 0.5, 1, Low/High)	25, 2000, 100, 1, 2	**6.17 ± 0.74**	7.42 ± 0.63	6.77 ± 0.78
4 (0.8, 0.5, 1, High/High)	25, 2000, 100, 10, 2	**6.05 ± 0.73**	7.55 ± 0.63	7.2 ± 0.8
5 (0.8, 0.5, 1, Low/Low)	25, 1000, 10, 10, 2	4.83 ± 0.68	**4.29 ± 0.52**	7.25 ± 0.84
6 (0.8, 0.5, 2, High/Low)	50, 2000, 10, 10, 2	4.55 ± 0.66	**4.43 ± 0.51**	7.41 ± 0.84
7 (0.8, 0.5, 2, Low/High)	50, 1000, 100, 10, 1	4.9 ± 0.68	**4.13 ± 0.46**	6.48 ± 0.78
8 (0.8, 0.5, 2, High/High)	50, 2000, 100, 10, 2	4.88 ± 0.7	**4.18 ± 0.5**	6.42 ± 0.78
9 (0.8, 0.8, 2, Low/Low)	25, 1000, 100, 10, 2	**7.29 ± 0.81**	11.68 ± 0.84	8.53 ± 0.88
10 (0.8, 0.8, 1, High/Low)	50, 1000, 10, 1, 2	[a]**7.16 ± 0.79**	10.39 ± 0.77	9.15 ± 0.91
11 (0.8, 0.8, 1, Low/High)	50, 2000, 100, 10, 1	**7.58 ± 0.83**	10.8 ± 0.78	8.33 ± 0.85
12 (0.8, 0.8, 1, High/High)	25, 2000, 10, 10, 1	[a]**7.31 ± 0.8**	10.82 ± 0.79	9.36 ± 0.92
13 (0.8, 0.8, 2, Low/Low)	25, 1000, 100, 10, 1	6.23 ± 0.79	**5.79 ± 0.58**	7.98 ± 0.89
14 (0.8, 0.8, 2, High/Low)	25, 1000, 10, 1, 1	6.3 ± 0.79	**5.69 ± 0.58**	6.68 ± 0.81
15 (0.8, 0.8, 2, Low/High)	25, 2000, 10, 10, 2	5.98 ± 0.77	**5.94 ± 0.6**	8.65 ± 0.91
16 (0.8, 0.8, 2, High/High)	25, 1000, 10, 10, 1	**6.14 ± 0.79**	6.25 ± 0.62	7.31 ± 0.84
17 (0.9, 0.5, 1, Low/Low)	25, 1000, 100, 1, 2	**20.69 ± 1.29**	20.88 ± 1.13	24.46 ± 1.35
18 (0.9, 0.5, 1, High/Low)	25, 1000, 100, 10, 2	21.02 ± 1.29	**19.9 ± 1.12**	22.58 ± 1.32
19 (0.9, 0.5, 1, Low/High)	25, 2000, 10, 1, 1	21.11 ± 1.29	**20.17 ± 1.12**	23.6 ± 1.34
20 (0.9, 0.5, 1, High/High)	50, 2000, 10, 10, 2	**20.78 ± 1.3**	21.49 ± 1.15	23.4 ± 1.32
21 (0.9, 0.5, 1, Low/Low)	25, 2000, 10, 1, 1	19.75 ± 1.29	[c]**15.22 ± 1.06**	22.46 ± 1.34
22 (0.9, 0.5, 2, High/Low)	25, 1000, 10, 10, 1	19.4 ± 1.28	[c]**14.33 ± 1.02**	22.46 ± 1.35
23 (0.9, 0.5, 2, Low/High)	25, 1000, 100, 1, 1	19.62 ± 1.3	[c]**14.44 ± 1.04**	21.47 ± 1.31
24 (0.9, 0.5, 2, High/High)	50, 1000, 100, 10, 2	19.67 ± 1.29	[c]**14.5 ± 1.03**	21.6 ± 1.33
25 (0.9, 0.8, 2, Low/Low)	50, 1000, 10, 10, 1	[b]**23 ± 1.34**	25.83 ± 1.2	25.66 ± 1.38
26 (0.9, 0.8, 1, High/Low)	50, 2000, 10, 10, 2	[b]**23.53 ± 1.36**	26.98 ± 1.24	25.94 ± 1.38
27 (0.9, 0.8, 1, Low/High)	25, 1000, 10, 1, 2	**23.27 ± 1.36**	25.16 ± 1.21	25.91 ± 1.38
28 (0.9, 0.8, 1, High/High)	25, 1000, 100, 10, 2	**23.25 ± 1.35**	25.76 ± 1.21	25 ± 1.37
29 (0.9, 0.8, 2, Low/Low)	50, 1000, 10, 1, 2	21.22 ± 1.34	[c]**17.95 ± 1.11**	24.19 ± 1.39
30 (0.9, 0.8, 2, High/Low)	25, 1000, 10, 1, 2	21.01 ± 1.34	**19.64 ± 1.15**	23.83 ± 1.38
31 (0.9, 0.8, 2, Low/High)	25, 1000, 100, 10, 2	21.19 ± 1.35	[c]**18.01 ± 1.12**	24.53 ± 1.4
32 (0.9, 0.8, 2, High/High)	50, 1000, 100, 1, 1	21 ± 1.34	**19.42 ± 1.17**	22.74 ± 1.36

[a] denotes statistical significance (as compared to the next best policy) with 95% confidence
[b] denotes statistical significance (as compared to the next best policy) with 90% confidence
[c] denotes statistical significance (as compared to the ADP policy) with 95% confidence⋆ denotes statistical significance (as compared to the ADP policy) with 90% confidence

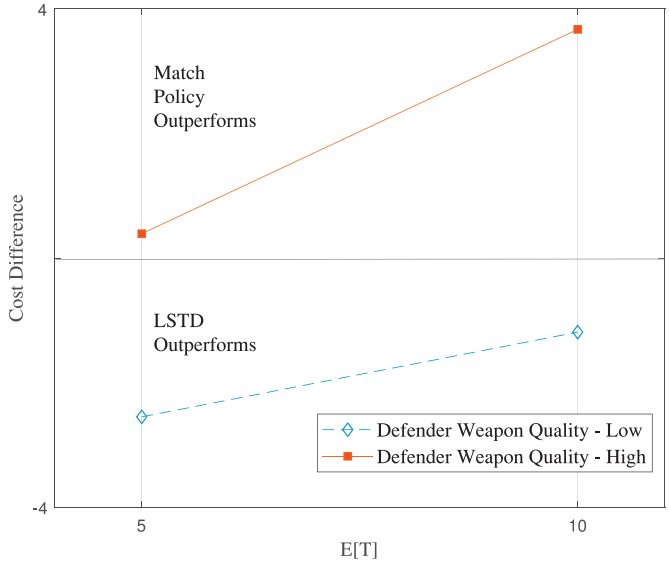

Fig. 5. Cost Difference Between LSTD and Match policies for Defender Weapon Quality and Conflict Duration.

TBM has a high likelihood of damaging its targeted asset, the LSTD policy outperforms the baseline policies, but it is interesting that, in shorter duration conflicts when the two baseline policies show very similar means, the ADP policy is able to outperform them at a statistically significant level.

Fig. 4 illustrates the cost difference between the LSTD and Match policies when we examine the two different levels of attacker weapon quality for short and long duration conflicts. We observe very similar results as those noted for LSPE.

Fig. 5 illustrates the cost difference between the LSTD and Match policies when we consider the two different levels of defender weapon quality for short and long duration conflicts. These results also follow the same trends discussed for LSPE.

4.3. Robustness of results

Table 6 shows the LSPE-determined three-run averages for the θ-vectors for each replication of the 32 instances. These three-run averages show a general robustness across the θ-vectors garnered from the given parameter settings. For most of the instances, there exists a small (i.e., < 1) cost difference between the best mean and the average mean. Although this difference may impact the statistical significance of those parameter settings versus the baseline policies, it does not indicate that any of the chosen best θ-vectors were simply outliers. Indeed, this result suggests an overall robustness with respect to the consistency of the LSPE algorithmic performance.

Table 7 shows the LSTD-determined three-run averages for the best θ-vectors for each replication of the 32 instances. These LSTD robustness results are very similar to those discussed for LSPE.

4.4. Meta analysis

When examining the parameter estimates in Table 8, we observe that conflict duration, attacker weapon quality, and defender

Table 6
LSPE results - Robustness.

	Algorithm Parameters	Run 1	Run 2	Run 3	Mean	Best	Difference
1	25, 2000, 100, 1, 2	5.75	6.62	6.23	6.20	5.75	0.45
2	25, 1000, 10, 10, 2	7.00	5.68	6.81	6.50	5.68	0.82
3	25, 2000, 100, 10, 1	6.09	6.52	6.27	6.29	6.09	0.20
4	25, 2000, 10, 1, 2	6.05	6.52	6.93	6.50	6.05	0.45
5	25, 2000, 100, 1, 2	5.69	4.78	6.03	5.50	4.78	0.72
6	50, 2000, 100, 10, 2	4.71	5.60	6.14	5.49	4.71	0.77
7	25, 1000, 100, 1, 2	6.04	4.99	5.85	5.63	4.99	0.64
8	50, 1000, 10, 10, 2	5.15	5.94	5.72	5.60	5.15	0.45
9	50, 2000, 10, 1, 1	8.27	9.43	7.11	8.27	7.11	1.16
10	25, 1000, 10, 1, 2	7.68	8.43	8.37	8.16	7.68	0.48
11	50, 2000, 100, 10, 1	8.59	7.49	8.33	8.13	7.49	0.65
12	25, 2000, 100, 1, 1	7.37	8.21	8.50	8.03	7.37	0.66
13	25, 2000, 10, 10, 2	6.07	7.07	7.47	6.87	6.07	0.80
14	50, 2000, 10, 10, 2	6.95	7.15	6.12	6.74	6.12	0.62
15	25, 1000, 10, 10, 1	7.00	5.97	7.35	6.77	5.97	0.80
16	50, 2000, 100, 10, 2	6.02	7.05	6.86	6.65	6.02	0.62
17	25, 1000, 100, 10, 1	22.46	21.11	22.85	22.14	21.11	1.03
18	25, 1000, 100, 10, 1	21.02	22.30	21.47	21.60	21.02	0.58
19	50, 1000, 100, 10, 2	21.77	20.27	22.24	21.42	20.27	1.15
20	25, 2000, 100, 10, 1	22.06	22.92	21.16	22.05	21.16	0.89
21	25, 1000, 100, 1, 2	20.90	21.27	19.46	20.54	19.46	1.09
22	50, 2000, 10, 1, 1	21.00	19.55	21.10	20.55	19.55	1.00
23	25, 2000, 10, 10, 2	19.60	21.19	20.17	20.32	19.6	0.72
24	25, 1000, 10, 1, 1	21.74	21.44	19.53	20.90	19.53	1.37
25	50, 1000, 10, 1, 2	23.56	25.45	22.97	23.99	22.97	1.02
26	50, 1000, 100, 1, 1	23.34	24.44	25.11	24.30	23.34	0.96
27	25, 2000, 100, 1, 1	25.92	22.85	25.10	24.62	22.85	1.78
28	50, 1000, 10, 1, 1	24.80	24.97	23.15	24.31	23.15	1.16
29	25, 1000, 100, 10, 1	21.38	22.06	23.68	22.37	21.38	0.99
30	50, 1000, 10, 10, 2	22.16	21.12	24.05	22.45	21.12	1.33
31	25, 1000, 10, 1, 2	23.06	20.88	22.07	22.00	20.88	1.12
32	50, 2000, 10, 1, 1	23.62	24.48	21.12	23.07	21.12	1.95

Table 7
LSTD results - Robustness.

	Algorithm parameters	Run 1	Run 2	Run 3	Mean	Best	Difference
1	25, 2000, 10, 10, 2	7.28	6.08	7.16	6.84	6.08	0.76
2	25, 2000, 10, 1, 2	6.56	6.92	6.15	6.55	6.15	0.40
3	25, 2000, 100, 1, 2	7.50	6.17	7.22	6.96	6.17	0.80
4	25, 2000, 100, 10, 2	6.71	6.05	7.01	6.59	6.05	0.54
5	25, 1000, 10, 10, 2	4.83	6.45	5.88	5.72	4.83	0.89
6	50, 2000, 10, 10, 2	5.99	5.33	4.55	5.29	4.55	0.74
7	50, 1000, 100, 10, 1	5.87	4.90	5.57	5.45	4.90	0.55
8	50, 2000, 100, 10, 2	4.88	5.19	5.41	5.16	4.88	0.28
9	25, 1000, 100, 10, 2	8.46	7.29	8.54	8.10	7.29	0.81
10	50, 1000, 10, 1, 2	8.34	9.50	7.16	8.33	7.16	1.17
11	50, 2000, 100, 10, 1	7.58	8.44	7.64	7.89	7.58	0.30
12	25, 2000, 10, 10, 1	8.51	7.31	8.50	8.10	7.31	0.80
13	25, 1000, 100, 10, 1	6.66	7.30	6.23	6.73	6.23	0.50
14	25, 1000, 10, 1, 1	6.71	7.61	6.30	6.88	6.30	0.57
15	25, 2000, 10, 10, 2	7.08	5.98	6.90	6.65	5.98	0.67
16	25, 1000, 10, 10, 1	6.14	7.02	7.13	6.76	6.14	0.63
17	25, 1000, 100, 1, 2	20.69	21.59	21.53	21.27	20.69	0.58
18	25, 1000, 100, 10, 2	21.02	22.69	22.35	22.02	21.02	1.00
19	25, 2000, 10, 1, 1	22.70	22.49	21.11	22.10	21.11	0.99
20	50, 2000, 10, 10, 2	20.78	21.86	22.28	21.64	20.78	0.86
21	25, 2000, 10, 1, 1	20.51	19.75	20.62	20.29	19.75	0.55
22	25, 1000, 10, 10, 1	20.58	19.40	20.73	20.24	19.40	0.83
23	25, 1000, 100, 1, 1	21.20	19.62	21.29	20.70	19.62	1.08
24	50, 1000, 100, 10, 2	20.91	19.67	20.85	20.48	19.67	0.81
25	50, 1000, 10, 10, 1	24.69	24.85	23.00	24.18	23.00	1.18
26	50, 2000, 10, 10, 2	26.31	23.53	24.24	24.69	23.53	1.17
27	25, 1000, 10, 1, 2	25.32	23.90	23.27	24.16	23.27	0.90
28	25, 1000, 100, 10, 2	23.43	23.89	23.25	23.52	23.25	0.27
29	50, 1000, 10, 1, 2	22.94	23.36	21.22	22.51	21.22	1.29
30	25, 1000, 10, 1, 2	21.01	22.24	23.05	22.10	21.01	1.09
31	25, 1000, 100, 10, 2	22.68	22.88	21.19	22.25	21.19	1.06
32	50, 1000, 100, 1, 1	21.00	24.84	22.60	22.81	21.00	1.81

Table 8
Parameter estimates - LSPE.

	Estimate	Standard Error	t Ratio	Probability $< \|t\|$
Intercept	−120.26	0.22	−555.56	< 0.0001
N (outer loops)	0.00	0.00	−0.79	0.43
K (inner loops)	0.00	0.00	1.47	0.14
a (smoothing)	0.00	0.00	−2.05	0.04
η (regularization)	0.00	0.00	0.61	0.54
ϕ (basis function set)	−0.01	0.02	−0.34	0.73
Conflict Duration	157.27	0.23	694.73	< 0.0001
Attacker Weapon Quality	5.74	0.08	76.05	< 0.0001
Defender Weapon Quality	−1.44	0.02	−63.43	< 0.0001
Asset 1 Value	0.00	0.00	−1.61	0.11
Asset 2 Value	0.00	0.00	−0.37	0.71
			Adj. R^2	0.99

Table 9
Parameter estimates - LSTD.

	Estimate	Standard error	t Ratio	Probability $< \|t\|$
Intercept	-122.76	0.21	-584.40	< 0.0001
N (outer loops)	0.00	0.00	0.57	0.57
K (inner loops)	0.00	0.00	0.1	0.92
a (smoothing)	0.00	0.00	-0.34	0.73
η (regularization)	0.00	0.00	-0.53	0.60
ϕ (basis function set)	0.00	0.01	-0.32	0.75
Conflict Duration	157.50	0.23	699.06	< 0.0001
Attacker Weapon Quality	5.84	0.08	77.8	< 0.0001
Defender Weapon Quality	0.73	0.01	64.44	< 0.0001
Asset 1 Value	0.00	0.00	-0.27	0.78
Asset 2 Value	0.00	0.00	0.56	0.58
			Adj. R^2	0.99

weapon quality have the largest impact on the change in the mean damage caused by the incoming TBMs. Although not statistically significant at a 95% confidence level, the Asset 1 value factor appears to explain more of the variation than the N, η, ϕ, and the Asset 2 value factors. Recall Asset 1 is protected by the THAAD air defense system, and the THAAD had the highest probability of kill across all scenarios. This result suggests that having a more effective air defense system co-located with the higher value asset leads to a greater ability to minimize the mean damage incurred, a relatively intuitive insight.

Examining the parameter settings for the LSPE ADP algorithm, we observe that the smoothing component explains a significant portion of the variation (with a 0.04 p-value). Although the number of inner loops (K) is not statistically significant, it explains more variation than the other parameter settings. The number of outer loops (N) likely did not have more impact on the mean because the smoothing coefficient did have an impact, and new information garnered from a higher number of outer loops received very little weight. We likely did not specify a large enough difference in the number of inner loops considered and, had we selected to perform 4000 inner loops, we might have observed statistical significance when varying this coefficient. Since the ϕ-value did not have an impact, future extensions to this work might benefit from searching for other sets of basis functions that perform better than the ones employed in this analysis.

Unlike the meta analysis conducted for LSPE, when we examine the parameter estimates in Table 9 for the LSTD policy performance, we find only conflict duration, attacker weapon quality, and defender weapon quality had a significant impact on the mean damage incurred. In fact, with the LSTD algorithm, none of the other terms had values anywhere close to exhibiting statistical significance. The LSTD algorithm performed nearly as well as LSPE against the two baseline policies, so this result might show that the LSTD algorithm performs well, regardless of parameter settings.

Table 10
Algorithm comparison.

Best ADP Algorithm	# of Scenarios
LSPE	19
LSTD	11
Tie	2

However, it also suggests better parameter settings might improve LSTD outcomes. Future research should include an expanded region of experimentation with respect to the LSTD algorithmic feature space.

4.5. ADP algorithm comparison

When comparing LSPE to LSTD, we observe in Table 10 that LSPE proves superior in 19 of the 32 scenarios. LSTD appears to perform better when duration is short, attacker weapon quality is low, and defender weapon quality is high. Moreover, when duration is long, attacker weapon quality is low, and defender weapon quality is high, LSTD also performs better. In most of the other problem instances, LSPE is either superior or equal to LSTD in terms of lowest mean cost, regardless of statistical significance.

5. Conclusion, recommendations, and future research

As theater ballistic missiles (TBMs) become more technologically advanced and more readily accessible to adversary nations around the world, the United States and its allies must maintain defensive capabilities with advanced air defense systems. Given limited resources, this situation requires countries to employ a networked, defense-in-depth strategy to best utilize the air and missile defense systems in the current inventory. As an integrated air and missile defense system becomes operational, the air defense community must reconsider which firing strategies are best for limited interceptor inventory.

To advance the technical modeling and analysis of this problem and garner useful insights, this research formulated a Markov decision process (MDP) model to represent an asset-based dynamic weapon target assignment problem (WTAP) in an elegant manner and obtain optimal firing decisions for small instances. This demonstrated model efficacy provided a starting point to compare the adequacy of heuristics for solving larger instances of interest to the air defense community. To address such larger instances, we developed two algorithmic variants of an approximate dynamic programming (ADP) approach to solve the MDP: a least squares policy evaluation (LSPE) and least squares temporal differences (LSTD) algorithm.

We designed and conducted computational experiments to examine the efficacy of the ADP solution variants. For comparison benchmarks, we considered a policy of only firing one interceptor at each incoming TBM (i.e., the Match Policy) and the current United States Army policy of firing two interceptors (i.e., the Overmatch Policy). Extensive testing yielded several insights. Among 32 instances representing different parametric factor levels, the LSPE and LSTD algorithms outperformed both baseline policies for 14 and 16 instances, respectively, at varying and often high (i.e., $\geq$ 90%) confidence levels. In general, the ADP policies outperform both baseline polices when conflict duration is short and attacker weapons are sophisticated; the Match Policy outperforms the tested ADP policies when conflict duration is long and attacker weapons are less sophisticated; and the Overmatch policy (i.e., the United States Army's currently implemented policy) is never the superlative policy for the test instances investigated. The domination of the Overmatch policy by alternative policies suggests that, as the military moves to an integrated, defense-in-depth strategy, it should consider a different firing policy for networked air defense systems having both mid-course and terminal systems.

Several extensions to this research can readily improve upon the MDP model by revisiting the underlying assumptions that inform it. First, this work assumed the attacker would not know the battle damage (BDA) resulting from the TBMs it fired. The model's realism can be improved if we assume the attacker is aware of the BDA, making targeting decisions that account for the remaining asset value. Second, the modeling herein assumed the traditional TBM and the TBM with multiple reentry vehicles (MeRV) cause the same amount of damage. It is unlikely that smaller MeRV warheads would cause as much damage as a traditional warhead; of interest to enhance realism would be an investigation of how different relative damage levels for the warheads impact the policies. Third, we only allowed the opportunity to fire one salvo of interceptors at TBMs during each of the mid-course and terminal phases due to computational limitations. To better investigate the firing solution utilizing either a shoot-shoot-look or shoot-look-shoot perspective, future work ought to allow two engagements of TBMs by interceptors at the mid-course and two at the terminal phase. Such an assumption requires sufficient time during those phases to fire one set of interceptors at incoming TBMs, assess which were destroyed, and then fire another set of interceptors at any remaining TBMs. From a computational perspective, this consideration notably increases the size of the state space, corresponding to 24 instead of 12 points in the TBM engagement space.

Additional enhancements may improve the ADP solution methods as well. Based on the problem instances wherein the ADP algorithms performed poorly against the Match Policy, one might consider a new basis function set that includes an indicator function to allow the firing decision to change from firing two interceptors to one based on interceptor inventory. This modification would allow the ADP policy to continue to outperform the Match Policy on the instances for which it already does so, but also perform at least as well for the instances the Match Policy currently exhibits a superlative performance.

Looking farther into the intended progress of this research thread, we intend to investigate several modifications to the problem and solution approaches. For example, including interceptor cost enables a more complete examination of an overall air defense situation and allows consideration of acquisition and system design decisions. Considering hybrid benchmark policies that switch from Overmatch (i.e., fire two interceptors) to Match (i.e., fire one interceptor) at different prescribed asset status and inventory thresholds is also of interest. Indeed, after structuring an ADP policy of such a form, the attendant task is to determine the appropriate switching threshold. Additional complications include the need for determining thresholds specific to each combination of TBM type, location, and friendly asset and weapon system statuses.

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Army, United States Air Force, Department of Defense, or United States Government. The authors thank the coordinating editor and three anonymous referees for their insightful comments and suggestions that substantively improved this manuscript.

References

Bertsekas, D.P., Homer, M.L., Logan, D.A., Patek, S.D., Sandell, N.R., 2000. Missile defense and interceptor allocation by neuro-dynamic programming. IEEE Trans. Syst. Man Cybernet. Part A 30 (1), 42–51.

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-Dynamic Programming. Athena Scientific, Belmont, MA.

Boardman, N.T., Lunday, B.J., Robbins, M.J., 2017. Heterogeneous surface-to-air missile defense battery location: a game theoretic approach. J. Heurist. 23 (6), 417–447.

Bradtke, S.J., Barto, A.G., 1996. Linear least-squares algorithms for temporal difference learning. Mach. Learn. 22 (1–3), 33–57.

Chang, T., Kong, D., Hao, N., Xu, K., Yang, G., 2018. Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization. Appl. Soft Comput. 70, 845–863. doi:10.1016/j.asoc.2018.06.014.

Copp, T., 2016. Carter: N. Korea launch shows need for robust Pacific missile defense. http://www.stripes.com/news/carter-n-korea-launch-shows-need-for-robust-pacific-missile-defense-1.415760. Accessed: 2016-07-19.

Davis, M.T., Robbins, M.J., Lunday, B.J., 2017. Approximate dynamic programming for missile defense interceptor fire control. Eur. J. Oper. Res. 259 (3), 873–886.

Foss, C.F., O'Halloran, J.C., 2017. Jane's Land Warfare Platforms: Artillery & Air Defence 2017–2018. IHS Markit, Coulsdon, United Kingdom.

Glazebrook, K., Washburn, A., 2004. Shoot-look-shoot: a review and extension. Oper. Res. 52 (3), 454–463.

Gourley, S.R., 2011. Soldier armed: PAC-3 MSE update. Army Mag. 61 (7), 65–66.

Gulpinar, N., Canakoglu, E., Branke, J., 2018. Heuristics for the stochastic dynamic task-resource allocation problem with retry opportunities. Eur. J. Oper. Res. 266 (1), 291–303. doi:10.1016/j.ejor.2017.09.006.

Han, C.Y., Lunday, B.J., Robbins, M.J., 2016. A game theoretic model for the optimal disposition of integrated air defense missile batteries. INFORMS J. Comput. 28 (3), 405–416.

Hosein, P.A., Walton, J.T., Athans, M., 1988. Dynamic weapon-target assignment problems with vulnerable C2 nodes. Technical Report. Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Jenkins, P.R., Robbins, M.J., Lunday, B.J., 2019. Approximate dynamic programming for military medical evacuation dispatching policies. INFORMS J. Comput. (in press).

Jeong, J., 2019. South Korea moves to kick its missile defense shield up a notch. *Defense News*. [Online; accessed 12-September-2019; https://www.defensenews.com/global/asia-pacific/2019/08/14/south-korea-moves-to-kick-its-missile-defense-shield-up-a-notch/].

Karasakal, O., 2008. Air defense missile-target allocation models for a naval task group. Comput. Oper. Res. 35 (6), 1759–1770.

Kim, J., 2016. China says South Korea's THAAD anti-missile decision harms foundation of trust. http://www.reuters.com/article/us-southkorea-thaad-china-idUSKCN10504Q. Accessed: 2016-07-22.

Kim, J., Park, J.-M., 2016. South Korea chooses site of THAAD U.S. missile system amid protests. http://www.reuters.com/article/us-northkorea-southkorea-thaad-idUSKCN0ZT03Fl. Accessed: 2016-07-31.

Kline, A., Ahner, D., Hill, R., 2019. The weapon-target assignment problem. Comput. Oper. Res. 105, 226–236.

Kwon, K.J., J. N., 2016. North Korea fires submarine-based ballistic missile: South Korea. http://www.military.com/daily-news/2012/09/12/nrc-dump-boost-phase-ballistic-missile-defense.html. Accessed: 2016-07-19.

Lagoudakis, M.G., Parr, R., 2003. Least-squares policy iteration. J. Mach. Learn. Res. 4, 1107–1149.
Manne, A.S., 1958. A target-assignment problem. Oper. Res. 6 (3), 346–351.
Mindock, C., 2015. US launching missile defense system in Europe, Africa that Russia hates. *International Business Times*. [Online; accessed 12-September-2019; http://www.ibtimes.com/us-launching-missile-defense-system-europe-africa-russia-hates-2028707].
National Research Council, 2012. Making sense of ballistic missile defense: An assessment of concepts and systems for US boost-Phase missile defense in comparison to other alternatives. National Academies Press.
Powell, W.B., 2009. What you should know about approximate dynamic programming. Naval Res. Logist. (NRL) 56 (3), 239–249.
Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd John Wiley & Sons, Hoboken, NJ.
Powell, W.B., 2012. Perspectives of approximate dynamic programming. Ann. Oper. Res. 13 (2), 1–38.
Powell, W.B., 2019. A unified framework for stochastic optimization. Eur. J. Oper. Res. 275 (3), 795–821.
Puterman, M.L., 1994. Markov Decision Processes: Discrete Stochastic Dynamic programming. John Wiley & Sons, Hoboken, NJ.
Raytheon, 2018. Sweden, US Sign Agreement for Patriot. [Online; accessed 12-September-2019; http://www.raytheon.com/news/feature/sweden-us-sign-agreement-patriot].
Raytheon, 2019. Global patriot solutions. [Online; accessed 12-September-2019; http://www.raytheon.com/capabilities/products/patriot].
Rettke, A.J., Robbins, M.J., Lunday, B.J., 2016. Approximate dynamic programming for the dispatch of military medical evacuation assets. Eur. J. Oper. Res. 254 (3), 824–839.
Schaffer, M.B., 2016. Boost-phase missile defense: another look at space and air-air engagements. Phalanx 49 (1), 30–37.
Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. MIT press, Cambridge, MA.
U.S. Missile Defense Agency, 2016a. The ballistic missile defense system. [Online; accessed 6-March-2017; https://www.mda.mil/system/system.html].
U.S. Missile Defense Agency, 2016b. Fact sheet: aegis ballistic missile defense. [Online; accessed 6-March-2017; https://www.mda.mil/global/documents/pdf/aegis.pdf].
U.S. Missile Defense Agency, 2016c. Fact sheet: command and control, battle management, and communications. [Online; accessed 6-March-2017; https://www.mda.mil/global/documents/pdf/c2bmc.pdf].
U.S. Missile Defense Agency, 2016d. A system of elements. [Online; accessed 6-March-2017; https://www.mda.mil/system/elements.html].
U.S. Missile Defense Agency, 2016e. The threat. [Online; accessed 6-March-2017; https://www.mda.mil/system/threat.html].
Van Roy, B., Bertsekas, D.P., Lee, Y., Tsitsiklis, J.N., 1997. A neuro-dynamic programming approach to retailer inventory management. In: Proceedings of the 36th IEEE Conference on Decision and Control, 4. IEEE, pp. 4052–4057.
Wu, L., Wang, H., Lu, F., Jia, P., 2008. An anytime algorithm based on modified GA for dynamic weapon-target allocation problem. In: IEEE Congress on Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE, pp. 2020–2025.
Xin, B., Chen, J., Peng, Z., Dou, L., Zhang, J., 2011. An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem. IEEE Trans. Syst. Man Cybernet. Part A 41 (3), 598–606.
Xin, B., Chen, J., Zhang, J., Dou, L., Peng, Z., 2010. Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics. IEEE Trans. Syst. Man Cybernet. Part C 40 (6), 649–662.