European Journal of Operational Research 259 (2017) 873–886

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Approximate dynamic programming for missile defense interceptor fire control

Michael T. Davis, Matthew J. Robbins*, Brian J. Lunday

Air Force Institute of Technology, Department of Operational Sciences, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, United States

ARTICLE INFO

Article history:
Received 27 April 2016
Accepted 10 November 2016
Available online 16 November 2016

Keywords:
Approximate dynamic programming
Least squares temporal differences
Markov decision processes
Military applications
Weapon target assignment problem

ABSTRACT

Given the ubiquitous nature of both offensive and defensive missile systems, the catastrophe-causing potential they represent, and the limited resources available to countries for missile defense, optimizing the defensive response to a missile attack is a necessary national security endeavor. For a single salvo of offensive missiles launched at a set of targets, a missile defense system protecting those targets must determine how many interceptors to fire at each incoming missile. Since such missile engagements often involve the firing of more than one attack salvo, we develop a Markov decision process (MDP) model to examine the optimal fire control policy for the defender. Due to the computational intractability of using exact methods for all but the smallest problem instances, we utilize an approximate dynamic programming (ADP) approach to explore the efficacy of applying approximate methods to the problem. We obtain policy insights by analyzing subsets of the state space that reflect a range of possible defender interceptor inventories. Testing of four instances derived from a representative planning scenario demonstrates that the ADP policy provides high-quality decisions for a majority of the state space, achieving a 7.74% mean optimality gap over all states for the most realistic instance, modeling a longer-term engagement by an attacker who assesses the success of each salvo before launching a subsequent one. Moreover, the ADP algorithm requires only a few minutes of computational effort versus hours for the exact dynamic programming algorithm, providing a method to address more complex and realistically-sized instances.

Published by Elsevier B.V.

1. Introduction

Currently, over 30 countries have inventories of theater ballistic missiles (George C. Marshall & Claremont Institutes, 2015) while an additional 50 employ multiple launch rocket systems (Global Firepower, 2015). Both of these weapon systems are capable of causing large amounts of damage and of inflicting a high number casualties on their targets. The proliferation of these weapon systems has increased their destructive potential to a worldwide scale while continued research and development on them has led to the creation of even more capable systems that can be used by their developers to threaten neighboring countries or demand concessions in exchange for halting their production. Even U.S. officials concede that, because of the country's recent focus on counter-terrorism, other world powers have closed the gap on guided munitions technology, and the U.S. is now facing the uncertainty of being able to win a "guided munitions salvo competition" (Goure, 2015).

The threat from these weapons has led to the development and spread of missile defense systems. The U.S.-developed Patriot system has been in service for over 30 years, seeing use in both Gulf Wars as well as other conflicts (Davenport, 2015), and it has been fielded by 12 other countries (Raytheon, 2015). One of the best known systems from recent conflicts is Israel's Iron Dome. Developed by Israel and funded mostly by the U.S., the Iron Dome boasts a 90% success rate of destroying incoming rockets headed towards civilian populations, intercepting over 500 rockets during Operation Protective Edge (The Jerusalem Post, 2015). Israel has exported its Iron Dome technology to Canada (Harress, 2015) and continues to work closely with India to develop cutting-edge surface-to-air missiles (SAM) (IBC News Bureau, 2015). Moreover, Israel is currently negotiating the export of the interceptor technology underlying the recent success of Iron Dome for joint production and use by the U.S. (Opall-Rome, 2016). Still more countries, like Turkey, are seeking to acquire long-range missile defense systems (Reuters, 2015), and the U.S. continues to push ahead with missile defense planning and coordination for Europe and Africa (Mindock, 2015).

The security these defense systems may provide comes at a significant financial cost. Initial acquisition costs can be billions of dollars, depending on the size and scope of the order. For example,

* Corresponding author.
E-mail addresses: matthew.robbins@afit.edu (M.J. Robbins), brian.lunday@afit.edu (B.J. Lunday).

http://dx.doi.org/10.1016/j.ejor.2016.11.023
0377-2217/Published by Elsevier B.V.

the cost to equip Qatar with the Patriot missile defense system in late 2014 was $2.4 billion (Tomkins, 2014). Once the system is in place, it must be modernized periodically to counter the evolution of missile threat systems. South Korea paid $770 million for a recent upgrade to its missile defense system (PR Newswire, 2015). Finally, the cost of the interceptor missiles themselves is a large part of the ongoing price of missile defense. The U.S. recently awarded a $1.5 billion contract to Lockheed Martin for an order of its latest interceptors (Brown, 2015), and Saudi Arabia has purchased 600 of the same missiles for $5.4 billion (Dillow, 2015).

Given the ubiquitous nature of both offensive and defensive missile systems, the catastrophe-causing potential they represent, and the limited resources available to countries for missile defense, optimizing the defensive response to a missile attack is a valuable endeavor. For a single salvo of offensive missiles launched at a set of targets, a missile defense system protecting those targets must decide how many interceptors to fire at each incoming missile. This decision is the well-studied static weapon-target assignment problem. However, missile engagements between an attacker and defender typically extend over many salvos of missile launches by the attacker. That is, the attacker does not launch all of its missiles at once. Instead, it launches subsets of its inventory at selected targets in discrete time periods. Hence, the defense cannot fire all of its interceptors at once; it must reserve some number of its inventory in consideration of subsequent attack salvos. This component of time is a distinguishing characteristic of the dynamic weapon-target assignment problem (DWTAP).

In this paper, we formulate an asset-based, defensive variant of the DWTAP wherein we explicitly model the status of the protected assets and interceptor inventory levels over time. We consider a sequence of "look-shoot" engagements in which the defender may fire only one interceptor salvo at the incoming salvo of missiles fired by the attacker. The size and intended targets of the incoming attacker salvo are not known by the defender prior to its launch by the attacker. Moreover, the total number of engagements (i.e., total number of salvos fired by the attacker) is also unknown to the defender. We develop a Markov decision process (MDP) model of this DWTAP. The large size of the motivating problem instance of interest yields a high-dimensional state space, suggesting that classical dynamic programming methods are inappropriate. As such, we apply approximate dynamic programming (ADP) methods to attain high-quality interceptor fire control policies. We develop and test an approximate policy iteration (API) algorithm that utilizes least-squares temporal differences (LSTD) for policy evaluation. We define a set of basis functions within a linear architecture to approximate the value function around the post-decision state. To demonstrate the applicability of our MDP model and to examine the efficacy of our proposed solution methodology, we construct a notional, representative missile defense planning scenario consisting of four related problem instances. We design and conduct a computational experiment to determine how selected problem features and algorithmic features affect the quality of solutions attained by our ADP policies as compared to the optimal policy.

This paper makes the following contributions. First, we examine an asset-based variant of the DWTAP in which the overlapping nature of the interceptor engagement envelopes and the presence of heterogeneous-valued assets creates an interesting, fundamental tension: should a particular SAM battery expend interceptors to protect a higher-valued asset, which is also protected by other SAM batteries, or reserve interceptors for protecting a lower-valued asset that only it can protect? Second, we propose an ADP method that represents a new solution approach for the DWTAP; the closest related work is that of Bertsekas, Homer, Logan, Patek, and Sandell (2000), who also utilize an ADP solution methodology to solve a DWTAP, but employ a neural network-based value function approximation scheme as opposed to our linear architecture approximation scheme with LSTD learning. Third, our research informs the continuing development of a larger, more comprehensive game theoretic perspective of the missile defense situation.

Previous work by Han, Lunday, and Robbins (2016) provides an initial game theoretic framework for the examination of the missile defense situation. Han et al. (2016) formulate and solve a defender-attacker-defender extensive-form game wherein the defender first decides where to locate its SAM batteries, the attacker subsequently decides which targets to engage with a *single salvo* (within which multiple missiles can target a single defender asset), and then the defender makes fire control decisions to intercept the incoming attacker missiles. In contrast to Han et al. (2016), our paper considers SAM batteries at affixed locations and provides a formulation to optimize a defender's fire control decisions in response to a *multiple-salvo* missile engagement by an attacker; thus, we decrease the emphasis on the rarely visited, long-term location decisions and improve the realism of missile engagements considered. Moreover, although our proposed MDP model is inherently a construct for a single decision maker (in this case, the defender), we incorporate a "smart" attacker into the formulation to better inform the resulting fire control policy. In a sequel to this work, we are working to address the optimization of the attacker's actions within a multiple-salvo missile engagement setting, utilizing a "smart" defender that acts according to the ADP policies developed herein. Once ADP techniques are developed for modeling both the defender's and attacker's respective policies, we intend to incorporate them within a broader, game theoretic framework to examine the larger, more realistic missile defense situation.

The remainder of this paper is organized as follows. Section 2 presents a review of pertinent literature concerning the weapon-target assignment problem. Several ADP papers that inform the development of our model and solution methodology are also reviewed. Section 3 presents a description of the DWTAP variant considered herein. Section 4 describes the MDP model formulation of the DWTAP and presents our ADP approach. In Section 5, we demonstrate the applicability of our model and examine the efficacy of our proposed solution methodology. Section 6 provides conclusions and directions for future research.

2. Literature review

Two streams of literature inform our work. The first stream of literature relates to the weapon-target assignment problem (WTAP). The second stream of literature involves selected works concerning approximate dynamic programming (ADP).

2.1. WTAP

The WTAP is a classical operations research problem of great importance to defense-related applications. Simply stated, the WTAP seeks an optimal assignment of a fixed number of weapons to a fixed number of targets to maximize the total damage inflicted on the set of targets. Research on the WTAP has increased through the years as threat systems and platforms proliferate in type and number, to the extent that a weapon-target assignment system that can efficiently solve a WTAP is now a key component of battlefield planning (Athans, 1987; Roux & Van Vuuren, 2007).

Research on the WTAP began in the 1950s. Manne (1958) developed a linear programming approximation, after which Bradford (1961) and Day (1966) studied WTAP modeling issues including its decomposition into subproblems with subsequent reconstitution.

Eckler and Burr (1972), Matlin (1970), Murphey (2000b, Chapter 3), and Cai, Liu, Chen, and Wang (2006) provide extensive surveys of the WTAP literature. Matlin (1970) reviews the literature

based on a set of five submodel characteristics. Each characteristic – the weapon system, the target complex, the engagement, the damage model, and the algorithm – is partitioned based on the complexity of the assumptions pertaining to that particular characteristic. Matlin (1970) and Murphey (2000b, Chapter 3) focus on target-based problems that apply to offensive, conventional warfare problems. In contrast, Eckler and Burr (1972) focus on asset-based problems that apply to defensive, strategic ballistic missile defense types of problems. Cai et al. (2006) focus their survey on the dynamic WTAP literature.

Exact algorithms for some WTAP formulations are proposed in the literature. The most well known is found in den Broeder, Ellison, and Emerling (1959); the authors present the minimum marginal return algorithm to solve the case of identical weapons. However, the general WTAP is NP-complete, as proven via a reduction from the exact cover problem by Lloyd and Witsenhausen (1986).

The two fundamental classes of the WTAP are the *static* and *dynamic* WTAPs. In a static WTAP, all parameters for the problem are known, and all weapons are assigned to the targets in a single stage. Comparatively, in a dynamic WTAP, weapons are assigned in stages with the assumption that the outcomes of the weapon-target engagements of the previous stage are observed before assignments for the current stage are made.

2.1.1. Static WTAP

Hosein and Athans (1990a) present the general static WTAP (SWTAP) formulation. Heuristic methods are the most common solution approach to the SWTAP. Lee, Lee, and Su (2002) and Wang, Qian, Guo, and Ma (2008) propose an ant colony optimization approach. Jaiswal, Shrotri, and Nagabhushana (1993) utilize simulated annealing to solve the SWTAP in the context of a multilayer defense scenario. Malhotra and Jain (2001) and Bisht (2004) solve a similar SWTAP scenario, using a genetic algorithm (GA) and a combined simulated annealing-GA approach, respectively. Karasakal, Özdemirel, and Kandiller (2011) examine a SWTAP in the context of protecting a naval task group, using an objective function that maximizes the probability that no missiles hit the task group; they develop and test two construction heuristics to solve their problem. Wacholder (1989) considers a neural network-based approach to solve the SWTAP. Lee, Su, and Lee (2003) follow a GA solution methodology with a novel method of gene recombination. Zeng et al. (2006) solve the SWTAP by leveraging advantages from both genetic algorithms and particle swarm optimization. Madni and Andrecut (2009) present simulated annealing and threshold accepting approaches to solve the SWTAP.

Ahuja, Kumar, Jha, and Orlin (2007) exploit the special structure of the SWTAP to formulate linear programming, mixed integer programming, network flow, and combinatorial lower-bounding schemes for proposed algorithms. The authors formulate the standard nonlinear WTAP as an integer programming problem with a convex objective function. They view this formulation as a generalized network flow problem with convex costs which they approximate in a piecewise-linear convex function so that the optimal solution to the modified problem provides a lower bound to the optimal solution to general problem. The authors propose several algorithms for solving the SWTAP as well as a construction heuristic and a very large-scale neighborhood (VLSN) search algorithm. Lee (2010) proposes a VLSN search algorithm to solve a constrained variant of the SWTAP in which the number of interceptors available to each weapon and the number of interceptors allowed to fire at each target have upper bounds. Kwon, Lee, Kang, and Park (2007) also propose a reformulation of the typical nonlinear integer programming formulation of the SWTAP and then present a branch-and-price algorithm as well as a greedy heuristic to solve the reformulated problem. Orlin (1987) transforms an integer programming model of the SWTAP into an equivalent minimum-cost network flow problem. In the author's formulation, the targets can receive at most one weapon. Green, Moore, and Borsi (1997) apply a goal programming-based approach to solve a SWTAP variant in which other realistic combat objectives are included in the formulation. Bogdanowicz, Tolano, Patel, and Coleman (2013) and Bogdanowicz (2009) solve variants of the SWTAP utilizing a modified auction algorithm approach. Menq, Tuan, and Liu (2007) consider a multilayer ballistic missile defense scenario utilizing a discrete-time Markov process model. The authors investigate the effectiveness and cost of different fire control policies. An interesting feature of their work is the ability to express the number of warheads penetrating the defenses in terms of a probability density function.

2.1.2. Dynamic WTAP

According to Murphey (2000b, Chapter 3), the dynamic WTAP (DWTAP) is formulated in one of two manners. The first model assumes that all targets are known from the start, whereas the second assumes that only a subset of targets is known while other targets may be revealed stochastically. The first formulation is also known as a shoot-look-shoot model and is the more widely studied variant. The second formulation allows for additional targets to become known as time progresses, making the decision problem one which addresses how many weapons to assign to the known targets and how many to reserve for future targets that may present themselves. This formulation is a stochastic demand problem and is introduced and studied in Murphey (2000a). Ahner and Parson (2015) exploit the structure of the stochastic demand problem to optimally solve a two-stage formulation.

As is the case for the SWTAP, heuristic methods are the most common approach to solving the DWTAP, given its computational complexity. Hosein and Athans (1990b) present the general DWTAP formulation as well as a heuristic solution method. Xin, Chen, Zhang, Dou, and Peng (2010) develop tabu search heuristics to solve an asset-based DWTAP and compare their solution procedure to other heuristics. Blodgett, Gendreau, Guertin, Potvin, and Séguin (2003) develop a construction heuristic and utilize tabu search to generate anti-ship missile defense policies. Xin, Chen, Peng, Dou, and Zhang (2011) utilize three rules based on the potential damage of an incoming missile and the potential benefit of a particular interceptor assignment to develop a heuristic that solves the asset-based DWTAP. Both Wu, Wang, Lu, and Jia (2008) and Khosla (2001) employ a GA approach, whereas Karasakal (2008) develops an integer linear programming model that addresses the defensive effectiveness of a naval task group. The formulation assumes a shoot-look-shoot policy and considers both point defenses as well as area defenses.

Simulation is frequently used to evaluate the effectiveness of air and missile defense systems. Hoyt (1985) and Li, Cong, and Xiong (2006) utilize simulation to examine variants of the DWTAP. Karasakal et al. (2011) report four further examples of naval air defense simulations in their literature review.

Soland (1987) uses stochastic dynamic programming to solve an asset-based, dynamic WTAP. The defender must protect a single asset against a simultaneous attack. The author provides numerical results as well as some extensions regarding the number of interceptors remaining for the defense. Sikanen (2008) seeks to solve an offensive variant of the DWTAP. The author employs dynamic programming to solve the problem exactly and also examines limited lookahead policies as an approximation approach.

Most DWTAP formulations assume that the defender accurately predicts the asset that an offensive missile is targeting. Leboucher, Le Menec, Kotenkoff, Shin, and Tsourdos (2013) relax this assumption so that a defender can only ascertain the particular region that is being targeted; the region may have one or more assets needing

defense. The authors propose a combined evolutionary game theory and discrete particle swarm optimization approach to solve the problem, and they provide computational results using numerical simulation.

Defended assets are typically comprised of more than a single type or purpose. Hosein, Walton, and Athans (1988) intersperses command, control, and communication (C3) nodes among the typical defended assets in their model. The authors study this structure for both dynamic and static WTAPs. The authors show that model formulation as a dynamic versus static WTAP significantly increases the effectiveness of the defense.

Bertsekas et al. (2000) model a DWTAP that is considerably more complex than the SWTAP. Instead of making one assignment of weapons to targets, the defender must decide how many weapons to employ against the current attack and how many to keep in reserve for subsequent attacks. Since Bellman's "curse of dimensionality" precludes the possibility of an exact solution for problems of even moderate size, the authors apply a class of reinforcement learning (i.e., ADP) methods called neuro-dynamic programming to address the dimensionality of the DWTAP. The neuro-dynamic programming framework for the DWTAP utilizes suboptimal solution methods to approximate the optimal function via neural networks and simulation. The authors develop four approximate policy iteration methods that generate a sequence of policies to allow for the approximate evaluation of the optimal functions.

2.2. Approximate dynamic programming

The decisions concerning which interceptors to assign to which missiles must be made sequentially over time and under uncertainty. Since fire control decisions impact the capability of the missile defense system to respond to future salvos, we must account for how current interceptor assignment decisions affect the future state of the system. As such, we formulate a Markov decision process (MDP) model of the DWTAP. Unfortunately, due to the resulting high dimensionality of such a formulation, an optimal policy cannot be identified utilizing classical exact dynamic programming algorithms. Instead, we leverage an ADP solution methodology to solve the DWTAP. Powell (2009, 2011, 2012) provides an introduction to ADP from an operations research perspective. Bertsekas and Tsitsiklis (1996) provide an engineering controls perspective and Sutton and Barto (1998) a computer science (artificial intelligence) perspective.

Two general algorithmic strategies exist for obtaining approximate solutions to our computational stochastic optimization problem: approximate value iteration (AVI) and approximate policy iteration (API). Bertsekas (2011) provides an excellent survey concerning API. We utilize an API algorithmic strategy to obtain a policy that maps the system state (e.g., asset and interceptor inventory status, incoming attacker missiles) to an action (e.g., assigning a number of interceptors from a particular SAM battery to each missile). Powell (2012) discusses four classes of policies: myopic cost function approximation, lookahead policies, policy function approximations, and policies based on value function approximations. We construct high-quality fire control policies based on value function approximations. Our approximation strategy involves the design of an appropriate set of basis functions for use within a linear architecture. Moreover, we approximate the value function around the post-decision state, enabling modification of Bellman's equation to obtain an equivalent, deterministic expression (Van Roy, Bertsekas, Lee, & Tsitsiklis, 1997). Adoption of the post-decision state convention addresses the challenge of the problem's high dimensionality with respect to the outcome space (Powell, 2011). Within the policy evaluation step of our API algorithm, we update the value function approximation for a fixed policy using least squares temporal differencing (LSTD) (Bradtke & Barto, 1996). LSTD is a computationally efficient method for estimating the adjustable parameters when using a linear architecture with fixed basis functions to approximate the value function for a fixed policy. Lagoudakis and Parr (2003) extend the LSTD algorithm to include the consideration of state-action pairs. We implement a variant of the LSTD algorithm that utilizes value function approximations around the post-decision state, as recommended by Powell (2011).

3. Problem description

U.S. military doctrine recognizes two main enemy threats to an integrated air and missile defense system: air threats (e.g., fighters and bombers) and ballistic missile (BM) threats (U.S. Joint Chiefs of Staff, 2014). Of the two, BMs are considered more difficult to counter by offensive targeting prior to their launch since, in general, they have smaller logistical footprints and are more easily maneuvered and concealed. Since an enemy's BM assets are unlikely to be completely destroyed prior to launch, it is essential to devise a defensive strategy to counter their use. U.S. military doctrine Assumes a missile defense system's ability to identify and target incoming BMs to include impact points, and it outlines planning considerations for countering a BM salvo against a set of defended assets (U.S. Joint Chiefs of Staff, 2014). These considerations – placement of surface-to-air-missile (SAM) batteries, return salvo size, interceptor inventories, and firing doctrine – attempt to enable the best possible response to an attack salvo. Since it is reasonable to assume that an attacker has a limited supply of BMs and a limited number of launchers, it is also reasonable to assume that an enemy would choose to stage an attack over several salvos to enable efficient use of limited assets via iterative battle damage assessment of the defended assets and to allow for reloading and/or repositioning of launchers. Thus, we view an attacker's BM campaign as a series of "look-shoot" engagements.

In our formulation, the defender has a set of cities, each having a quantitative value, it wishes to protect from incoming missiles using a predetermined configuration of SAM batteries with preallocated supplies of interceptors that cannot be replenished. SAM batteries are assumed to be collocated with a city (although not every city may have one) and to have a predefined protection radius that determines which cities each SAM battery can defend. Cities, but not SAM batteries, are assumed to be destroyed if at least one attacking missile targeting the city is not successfully intercepted. Implied herein is an attacker strategy that is countervalue focused rather than counter-force focused. The attacker fires salvos of missiles over time, targeting the defender's cities. We assume each missile carries only one warhead. The attacker can observe the status of each city prior to launching an attack. The attacker employs a randomized firing strategy in that the selection of targets for each salvo and the number of missiles fired at each city within a salvo is random. The total number of salvos fired is also random. This modeling convention represents the uncertainty of the engagement from the defender's perspective. A political solution (i.e., a diplomatic agreement is reached) or the overall military situation (e.g., the defender's offensive assets destroy the attacker's missile launchers) may dictate how many salvos the attacker is able to fire prior to cessation of hostilities. With respect to the attacker's targeting policy, it is reasonable to assume that it desires to be unpredictable, as defending against an unpredictable foe is more difficult. Indeed, a randomized attacker strategy provides the attacker with an element of surprise, an important principle in warfare.

Once an attack is launched, the defender can identify which city has been targeted by each missile. The defender must then decide how many interceptors to allocate from among its SAM batteries

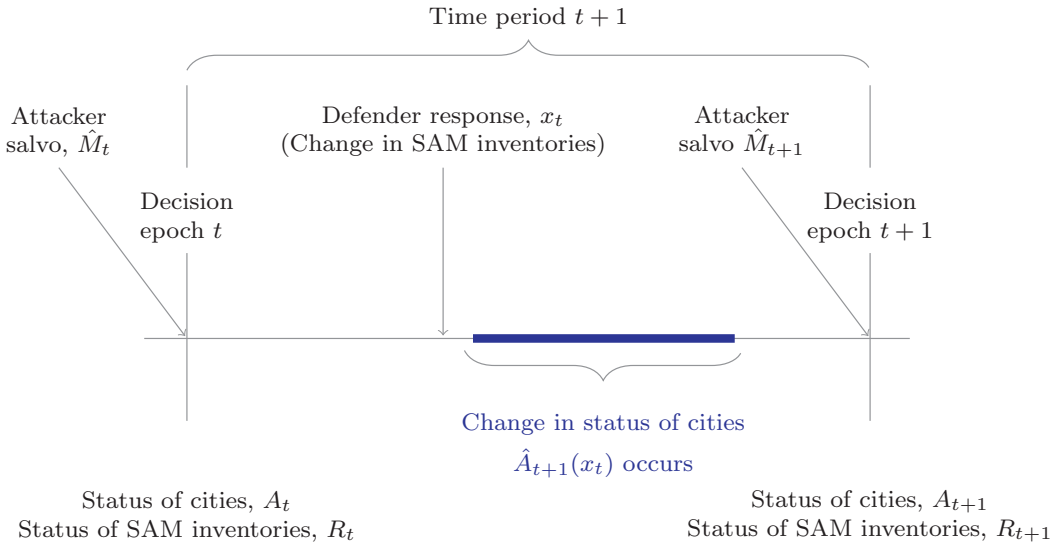

Fig. 1. Diagram outlining the timing of events for the MDP model.

to each incoming missile and how many to keep in reserve for repelling subsequent BM attacks. We wish to maximize the expected total value of the cities that remain after all attack salvos have been launched. Equivalently, we wish to minimize the expected total cost of destroyed cities over all decision epochs. Fig. 1 shows a timing diagram of the Markov decision process (MDP) model, utilizing notation introduced in Section 4.

4. Methodology

This section describes the Markov decision process (MDP) model formulation of the dynamic weapon-target assignment problem (DWTAP). The approximate dynamic programming (ADP) methodology utilized to obtain high quality solutions to the problem is also presented.

4.1. MDP formulation

The MDP model is formulated as follows. Let $\mathcal{T} = \{1, 2, ..., T\}$, $T \leq \infty$ be the set of decision epochs. The number of decision epochs T is random and follows a geometric distribution with parameter $\gamma \in [0, 1)$, independent of the interceptor fire control policy.

The state space consists of three components: the status of each city, the inventory of each SAM battery, and the vector representing the attacker's salvo, hereafter referred to as the *attack vector*.

The city status component is defined as

$$A_t = (A_{ti})_{i\in\mathcal{A}} \equiv (A_{t1}, A_{t2}, ..., A_{t|\mathcal{A}|}),$$

where $\mathcal{A} = \{1, 2, ..., |\mathcal{A}|\}$ is the set of all cities, and $A_{ti} \in \{0, 1\}$. A_{ti} is the status of city $i \in \mathcal{A}$ at decision epoch t with 1 indicating the city is alive and 0 indicating the city is destroyed.

The SAM inventory status is defined as

$$R_t = (R_{ti})_{i\in\mathcal{A}} \equiv (R_{t1}, R_{t2}, ..., R_{t|\mathcal{A}|}),$$

where $R_{ti} \in \{0, 1, ..., r_i\}$ and r_i = initial inventory of interceptors at SAM battery $i \in \mathcal{A}$. R_{ti} is the number of interceptors at SAM battery $i \in \mathcal{A}$ at decision epoch t.

Let $\hat{\mathcal{M}}_t = \{1, 2, ..., |\hat{\mathcal{M}}_t|\}$ be the set of all fired attacker missiles at decision epoch t. $\hat{\mathcal{M}}_t$ is the collection of observed incoming BMs that must be targeted by the defense at time t. The attack vector is defined as

$$\hat{M}_t = (\hat{\mathcal{M}}_{ti})_{i\in\mathcal{A}} \equiv (\hat{\mathcal{M}}_{t1}, \hat{\mathcal{M}}_{t2}, ..., \hat{\mathcal{M}}_{t|\mathcal{A}|}),$$

where $\hat{\mathcal{M}}_{ti} \subseteq \hat{\mathcal{M}}_t$ is the set of missiles fired at city i at decision epoch t, and the tuple $\hat{M}_t$ forms a disjoint set partition of $\hat{\mathcal{M}}_t$. The information provided by $\hat{M}_t$ is available to the defender at time t. However, the arrival of new information, $\hat{M}_{t+1}$, is random and can be conditioned on A_{t+1}. Let $\mathbb{P}^{\hat{M}_t}(m) = \mathbb{P}(\hat{M}_t = m|A_t)$ denote the probability distribution of the attacker BM salvo $\hat{M}_t$. This distribution is conditioned on A_t, meaning that the battle damage assessment capabilities of an attacker will determine the likelihood that a particular attack vector arrives to the system.

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, where $\mathcal{S}$ is the set of all possible states.

At each epoch t, the defender must decide how many interceptors to assign to each missile targeting a city. The defender must make this choice from among the SAM batteries that have the given city within their respective protection radii. From the *a priori* placement of SAM batteries relative to the cities, we can deduce a coverage matrix for the entire defended area. From this coverage matrix, we can determine which SAM batteries can intercept each incoming missile. Let $x_{tij} \in \mathbb{N}^0$ be the number of interceptors fired by SAM battery $i \in \mathcal{A}$ against missile $j \in \hat{\mathcal{M}}_{ti}^A$ at decision epoch t, where $\hat{\mathcal{M}}_{ti}^A$ is defined as the set of missiles that can be intercepted by SAM battery i at decision epoch t. Let $x_t = (x_{tij})_{i\in\mathcal{A}, j\in\hat{\mathcal{M}}_{ti}^A}$ denote the defender's decision vector. We define the set of all feasible defender actions (i.e., assignment of interceptors to missiles) as

$$\mathcal{X}_{S_t} = \left\{ x_t : \sum_{j\in\hat{\mathcal{M}}_{ti}^A} x_{tij} \leq \min\left(R_{ti}, x_i^{max}\right), \quad \forall\, i \in \mathcal{A} \right\},$$

where the constraint $\sum_{j\in\hat{\mathcal{M}}_{ti}^A} x_{tij} \leq \min(R_{ti}, x_i^{max})$ ensures that each SAM battery $i \in \mathcal{A}$ cannot fire more interceptors than it has in its inventory at epoch t and cannot fire more interceptors than can be simultaneously controlled (i.e., x_i^{max}) due to the performance characteristics of its target tracking radar.

We define transition functions and transition probability functions to characterize how the system evolves from one state to another as a result of decisions and information (Powell, 2011). The state transition function is defined as $S_{t+1} = S^M(S_t, x_t, W_{t+1})$, where $W_{t+1} = (\hat{A}_{t+1}, \hat{M}_{t+1})$ represents all the information (i.e., city status and attacker BM salvo) that becomes known at decision

epoch $t+1$. We define the city status transition function as

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

where $\hat{A}_{t+1,i}(x_t)$ is a random variable representing the status of city i after salvo $\hat{M}_t$ and the interceptor allocation decision x_t. This information depends on x_t since the number of interceptors fired at the inbound BMs affects a city's probability of survival. We define the inventory status transition function as

$$R_{t+1,i} = R_{ti} - \sum_{j \in \hat{\mathcal{M}}_{ti}^{A}} x_{tij}, \quad \forall\, i \in \mathcal{A},$$

and note that the city status transition function is stochastic whereas the inventory status transition function is deterministic.

The probability of transitioning from state S_t to S_{t+1} is conditioned on both the state of the system and the action chosen by the defender at decision epoch t. We assume the defender has one interceptor type, the attacker has one missile type, and that any missile that is not intercepted results in the certain destruction of the targeted city. We define $q \in (0, 1)$ to be the probability an attacking missile survives being targeted by a single interceptor. Then $\rho_{tj} = \prod_{i \in \mathcal{A}} q^{x_{tij}}$ is the probability that missile $j \in \hat{\mathcal{M}}_t$ survives being targeted by all interceptors fired against it at decision epoch t. We define

$$\psi_{ti} = \begin{cases} \prod_{j \in \hat{\mathcal{M}}_{ti}} (1 - \rho_{tj}) & \text{if } \hat{\mathcal{M}}_{ti} \neq \emptyset, \\ 1 & \text{if } \hat{\mathcal{M}}_{ti} = \emptyset, \end{cases}$$

as the probability that city $i \in \mathcal{A}$ survives to decision epoch $t+1$. Thus $\hat{A}_{t+1,i}(x_t)$ follows a Bernoulli probability distribution with parameter ψ_{ti}. Then

$$p(S_{t+1}|S_t, x_t) = \begin{cases} \mathbb{P}^{\hat{M}_{t+1}}(m) \prod_{i \in \mathcal{A}} \psi_{ti}^{A_{t+1,i}} (1 - \psi_{ti})^{A_{ti} - A_{t+1,i}} & \text{if } A_{ti} \geq A_{t+1,i} \text{ and} \\ & R_{t+1,i} = R_{ti} - \sum_{j \in \hat{\mathcal{M}}_{ti}^{A}} x_{tij}, \\ & \text{and } \hat{M}_{t+1} = m, \\ 0 & \text{otherwise,} \end{cases}$$

is the transition probability function from state S_t to S_{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 $\hat{C}(S_t, x_t, \hat{A}_{t+1}) = \sum_{i \in \mathcal{A}} v_i (A_{ti} - \hat{A}_{t+1,i})$, where v_i is the value of city $i \in \mathcal{A}$. We rewrite the cost function in terms of only the current state and decision by taking its expected value

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

We are now ready to optimize to determine the best fire control policy. Let $X^{\pi}(S_t)$ be a decision function (i.e., policy) that prescribes defender fire control actions for each state $S_t \in \mathcal{S}$. We wish to determine the optimal policy π^* from the class of policies $(X^{\pi}(S_t))_{\pi \in \Pi}$ that minimizes the expected total cost of destroyed cities over all epochs. Recall that the number of epochs T (i.e., the number of salvos launched by the attacker) is geometrically distributed with parameter γ. Thus our objective is

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

where $\mathbb{E}^T$ denotes the expectation with respect to the probability distribution of T, and where $\mathbb{E}^{\pi}$ indicates that the expectation depends on the defender's chosen policy π. Puterman (1994) shows that 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 fire control policies according to the expected total cost criterion; however, the number of decision epochs T is uncertain. The parameter γ acts as a discount factor and models an uncertain BM campaign length. We note the following relationship

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

Incorporating an uncertain horizon allows us to model the randomness of how long a multi-period missile engagement may last. The duration of a conflict is highly dependent on the overall context of the conflict situation and is therefore quite uncertain. The duration may well be dictated by a potential political resolution (i.e., a diplomatic agreement is reached) or military events exogenous to the model (e.g., the defender's offensive assets destroy the attacker's BM launchers).

To determine the 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 following decision function

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

4.2. ADP formulation

The MDP model formulation provides an elegant framework for the DWTAP. However, the application of exact dynamic programming algorithms to the problem is limited to very small instances. This limitation exists because our problem suffers from the curses of dimensionality. For example, consider the dimensionality 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 $\hat{M}_t$ represent the status of each city, the status of each SAM battery's inventory, and the attack vector at decision epoch t, respectively. Since city status is binary, there are $2^{|\mathcal{A}|}$ possibilities for A_t. Moreover, since SAM batteries are collocated at cities, there are $\prod_{i \in \mathcal{A}} (r_i + 1)$ possibilities for R_t. Let M be the maximum number of attacker missiles that can be fired at the set of defender cities at any epoch t. That is, the attacker may fire up to and including M missiles total at each decision epoch. Then there are $\binom{|A|+M}{M}$ possibilities for $\hat{M}_t$. Hence, there are $2^{|\mathcal{A}|} \cdot \prod_{i \in \mathcal{A}} (r_i + 1) \cdot \binom{|A|+M}{M}$ possibilities for S_t. This means that, given a problem instance of 10 cities and 5 SAM batteries having 10 interceptors each and an attacker firing up to 10 missiles, we would have 3×10^{13} possible states for a single epoch. Because classical dynamic programming algorithms such as policy iteration and value iteration that solve the Bellman equation exactly rely on enumeration of the state space, all but the smallest problem instances are computationally intractable for practical implementation.

ADP provides an alternative set of solution strategies that can be applied to problems that suffer from one or more curses of dimensionality. Since our use of an approximate policy iteration (API) algorithmic strategy requires the determination of approximate solutions to Eq. (1), we proceed by rewriting the Bellman equation using the post-decision state variable convention. Let

$$J^x(S_t^x) = \mathbb{E}\{J(S_{t+1}) | S_t^x\}$$

denote the value of being in post-decision state S_t^x. We can then define the relationship between $J(S_t)$ and $J^x(S_t^x)$ with the following equations

$$J^x(S_{t-1}^x) = \mathbb{E}\{J(S_t) | S_{t-1}^x\}, \tag{2}$$

$$J(S_t) = \min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)). \tag{3}$$

By substituting Eq. (3) into Eq. (2), we obtain the Bellman equation around the post-decision state variable

$$J^x(S^x_{t-1}) = \mathbb{E}\left\{ \min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma J^x(S^x_t)) \middle| S^x_{t-1} \right\}$$

The important distinction between this post-decision state form and the standard form of the Bellman equation from Eq. (1) is the swapping of the expectation and minimum operators. The swap provides computational advantages in that it lets us avoid approximating the expectation explicitly within the optimization problem, and it allows us to control the structure of our value function approximations.

4.2.1. Value function approximation

We estimate our value function using regression methods. In linear regression, the problem is one of estimating a vector to fit a model that will predict a variable using a set of observations. For our model, we wish to estimate the parameter θ using observations that are created from a set of basis functions $\phi_f(S)$, $f \in \mathcal{F}$. The set $\mathcal{F}$ of basis functions allows us to reduce the dimensionality of the state variable to a selected number of features, $|\mathcal{F}|$. For example, a basis function $f \in \mathcal{F}$ for our problem might be the interceptor inventory at a SAM battery. Using the post-decision state, we can write our value function approximation in a form similar to a standard linear regression model

$$\bar{J}^x(S^x_t) = \sum_{f \in \mathcal{F}} \theta_f \phi_f(S^x_t). \tag{4}$$

Our Bellman equation is then expressed as follows

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

We refer to the portion of the Bellman equations inside the expectation operator as the *inner minimization problem*, or IMP.

4.2.2. IMP

Consider the IMP within our formulation

$$\min_{x_t \in \mathcal{X}_{S_t}} \left(C(S_t, x_t) + \gamma \sum_{f \in \mathcal{F}} \theta_f \phi_f(S^x_t) \right). \tag{5}$$

If we assume $\theta_f = 0, \quad \forall f \in \mathcal{F}$, then our IMP is simply the minimization of the one period cost function

$\min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t))$, where $C(S_t, x_t) = \mathbb{E}\{\sum_{i \in \mathcal{A}} v_i (A_t - \hat{A}_{t+1,i})\}$. Consider a problem instance of one city and one SAM battery wherein two attacker missiles have been fired at the city. Then $C(S_t, x_t) = \mathbb{E}\{v(A_t - \hat{A}_{t+1})\} = vA_t - v\psi(x_t)$. Since vA_t is a constant, we reduce the IMP to

$$\min_{x_t \in \mathcal{X}_{S_t}} (-v(1 - q^{x_{t1}})(1 - q^{x_{t2}})).$$

Influencing our pending choice of solution methodology and its corresponding efficacy, we note the following theorem.

Theorem 1. *The integer relaxation of the IMP is not a convex optimization problem.*

Proof. By contradiction, assume that the IMP is a convex optimization problem. Consider a problem instance of one city and one SAM battery wherein two attacker missiles have been fired at the city; let $\theta_f = 0, \quad \forall f \in \mathcal{F}$. Then $\min_{x_t \in \mathcal{X}_{S_t}} (-v(1 - q^{x_{t1}})(1 - q^{x_{t2}}))$ is convex on $\mathcal{X}_{S_t} = \{x_t : x_{t1} + x_{t2} \le R_t\}$, and so the Hessian $H(x_{t1}, x_{t2})$ of $-v(1 - q^{x_{t1}})(1 - q^{x_{t2}})$ is positive definite on $\mathcal{X}_{S_t}$.

Consider the Hessian when $R_t = 10$. This instance yields

$$H(x_{t1}, x_{t2}) = \begin{bmatrix} 2.59029 \times 0.2^{x_{t1}}(1 - 0.2^{x_{t2}}) & -2.59029 \times 0.2^{x_{t1}+x_{t2}} \\ -2.59029 \times 0.2^{x_{t1}+x_{t2}} & 2.59029 \times 0.2^{x_{t2}}(1 - 0.2^{x_{t1}}) \end{bmatrix}.$$

Now, consider that a feasible solution in $\mathcal{X}_{S_t} : x_{t1} = 0, x_{t2} = 1$, results in

$$H(0, 1) = \begin{bmatrix} 2.07223 & -0.518058 \\ -0.518058 & 0 \end{bmatrix}.$$

Since $(2.07223)(0) - (-.518058)^2 < 0$, by Lemma 3.3.11 of Bazaraa, Sherali, and Shetty (2013), $H(0, 1)$ is not positive definite, a contradiction. □

Due to Theorem 1, both the IMP and its integer relaxation are nonconvex and hence lack an exact solution method other than exhaustive enumeration of $\mathcal{X}_{S_t}$.

4.2.3. Algorithmic strategy

API is an algorithmic strategy that produces a sequence of policies and associated approximate value functions through iterations with two phases. In the *policy evaluation* phase the value function approximation of a policy is evaluated. In the *policy improvement* phase a new (updated) policy is generated. We evaluate a policy by approximating the value function utilizing a linear model (see Eq. (4)). We generate an improved policy from the policy based on the current value function approximation. A temporal difference (TD) learning method is applied within our API framework. TD methods represent an important class of ADP solution techniques and have evolved to include a number of variations. The least squares temporal differences (LSTD) method collects batches of temporal differences and then applies least squares regression to determine the best fit parameters for the updated value function approximation (Bradtke & Barto, 1996). We utilize LSTD to evaluate the approximate value of a policy which we then use to improve the policy.

Algorithm 1 shows API-LSTD adapted from Powell (2011) to our problem. The 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. Once the value $\phi(S^x_{t-1,k})$ is recorded, we simulate forward to the next pre-decision state and select the best decision as per Eq. (5). We record the associated expect cost $C(S_{t,k}, x_t)$ 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 $(C(S_{t,k}, x_t) + \gamma\phi(S^x_{t,k})^T\theta^n) - \phi(S^x_{t,k-1})^T\theta^n$.

The policy improvement loop of the algorithm begins once K temporal difference sample realizations have been collected. We compactly denote basis function matrices and cost vectors as follows. Let

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

$$C_t \triangleq \begin{bmatrix} C(S_{t,1}, x_t) \\ \vdots \\ C(S_{t,K}, x_t) \end{bmatrix},$$

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 perform a least squares regression of Φ_{t-1} and Φ_t against C_t to ensure the sum of the K temporal differences equals zero and calculate $\hat{\theta}$ as per Equation (6). We update our estimate of θ according to Equation (7) where $\alpha_n = \frac{a}{a+n-1}, a \in (0, \infty)$ denotes our smoothing function. The smoothing function controls the rate of

Algorithm 1 LSTD algorithm for infinite horizon problems using basis functions.

Initialization:
Initialize θ^0.
Set $n = 1$.
Set the initial policy:
$X^\pi(S_t|\theta^{n-1}) = \arg\min_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma \phi(S^{M,x}(S_t, x_t))^T \theta^{n-1})$
Do for $n = 1, ..., N$:
Do for $k = 1, ..., K$:
Generate random post-decision state $S^x_{t-1,k}$.
Record $\phi(S^x_{t-1,k})$.
Sample W_{t+1}.
Compute the next pre-decision state $S_{t,k}$.
Compute the decision $x_t = X^\pi(S_{t,k}|\theta^{n-1})$.
Compute post-decision state $S^x_{t,k} = S^{M,x}(S_{t,k}, x_t)$.
Record $C(S_{t,k}, x_t)$.
Record $\phi(S^x_{t,k})$.
Update θ^n and the policy:

$$\hat{\theta} = [(\Phi_{t-1} - \gamma\Phi_t)^T(\Phi_{t-1} - \gamma\Phi_t)]^{-1}(\Phi_{t-1} - \gamma\Phi_t)^T C_t \quad (6)$$

$$\theta^n = \alpha_n \hat{\theta} + (1 - \alpha_n)\theta^{n-1} \quad (7)$$

Return $X^\pi(S_t|\theta^N)$ and θ^N.

convergence of the algorithm. Higher values of the parameter a slow the rate at which α_n drops to zero. Smoothing θ completes one policy improvement step.

5. Computational results

In this section, we demonstrate the applicability of our model to a problem of interest to the military missile defense community and examine the efficacy of our proposed solution methodology. We present a representative ballistic missile (BM) defense scenario (in lieu of a particular, highly sensitive, and classified scenario) as the basis for our analysis. From this representative scenario, we create four test instances. We solve each instance exactly utilizing Gauss–Seidel modified policy iteration, a classical dynamic programming algorithm, and approximately by employing the approximate dynamic programming (ADP) solution methodology described in the previous section. Moreover, for each instance, we conduct computational experiments to identify the best performance settings for the ADP algorithm. Further, we compare the optimal and ADP interceptor fire control policies for selected subsets of the state space to enhance our understanding of the proposed methodology. For the computational experiments, we utilize a dual Intel Xeon E5-2650v2 workstation having 192 gigabytes of RAM and MATLAB's Parallel Computing Toolbox.

5.1. Representative scenario

We present a BM defense scenario consisting of three cities defended by two surface-to-air-missile (SAM) batteries. The SAMs are located at the first and third cities and are positioned in such a way as to overlap the second, or "middle" city. That is, the first SAM can defend the first and second cities whereas the second SAM can defend the second and third cities. The cities' values are 1, 10, and 5 units, respectively. Each SAM battery has a preallocation of 10 interceptors and a firing limit of four interceptors per salvo. We set $q = 0.1$ to capture interceptor performance characteristics and consider an attacker that can fire up to three missiles per salvo across all cities.

Table 1
Test instance problem features.

Instance	Expected conflict Duration	Attacker BDA
I	Long ($\mathbb{E}\{T\} = 5$)	Not performed (BDA=0)
II	Long ($\mathbb{E}\{T\} = 5$)	Performed (BDA=1)
III	Short ($\mathbb{E}\{T\} = 2$)	Not performed (BDA=0)
IV	Short ($\mathbb{E}\{T\} = 2$)	Performed (BDA=1)

Table 2
Experimental design factors and levels.

N	K	$\phi(S)$	a
5	50	1	0.01
10	100	2	0.5
15	150	3	1.0

Table 3
Basis function features.

$\phi(S)$	$\phi_0(S)$	$\phi_1(S)$	$\phi_2(S)$	$\phi_3(S)$	$\phi_4(S)$	$\phi_5(S)$
1	1	A_t	R^x_t			
2	1	A_t	R^x_t	A^x_t		
3	1	R^x_t	A^x_t	$(R^x_t)^2$	$(A^x_t)^2$	$R^x_t A^x_t$

From this representative scenario, we developed four test instances by varying two of the problem features. The first feature we varied was the expected duration of the conflict, in terms of the number of expected attack salvos, $\mathbb{E}\{T\}$. Recall that T is geometrically distributed with parameter γ. We chose two γ-values, 0.5 and 0.8, corresponding to an expected number of salvos of $\mathbb{E}\{T\} = 2$ and $\mathbb{E}\{T\} = 5$, respectively. The γ-parameter variation allows us to explore the impact of the expected number of salvos on the policies and objective values. The second feature we varied was the battlefield damage assessment (BDA) capabilities of the attacker. BDA settings are either zero, indicating the attacker has no BDA capabilities, or one, indicating the attacker can identify the status of each city prior to launching a missile salvo. For both BDA settings, we utilize a multinomial probability distribution to characterize the attack salvo. We assume the attacker fires 1, 2, or 3 missiles in a salvo with equal probability; given this outcome, when no BDA capability is present (i.e., BDA = 0), the probability an attacker fires at a city is the proportion of the city's value to the total value of all cities. When BDA capability is present (i.e., BDA = 1), the probability an attacker fires at a city is the proportion of the city's value to the total value of the *remaining* cities. As an example, if City 1 and City 2 are alive (and City 3 is dead) then, when BDA = 0, the multinomial probability distribution is parameterized by the tuple (1/16, 10/16, 5/16). When BDA = 1 for the same city statuses, the multinomial probability distribution is parameterized by the tuple (1/11, 10/11, 0). Table 1 shows the problem feature settings for each test instance.

5.2. Experimental design and tuning of algorithmic parameters

For each of the four test instances, we wish to determine the best parameter settings for Algorithm 1. We focus on parameters N, K, $\phi(S)$, and a. Table 2 shows the 3-level, 4-factor experimental design, and Table 3 shows the set of features for each design level of the $\phi(S)$ factor. The levels for each factor were chosen based on initial experimental runs of the model.

For each test instance, we ran a full factorial experiment for 30 random number seeds for a total of 2430 runs per instance. For each run, we recorded the mean of the optimality gap for the states containing the full complement of cities and interceptors. For each instance, we chose the settings that yielded the lowest

Table 4
Best performing ADP algorithm parameter-values.

Instance	N	K	$\phi(S)$	a
I	10	50	1	0.01
II	15	50	2	1.0
III	15	150	3	1.0
IV	5	50	3	0.01

mean value out of all runs. The best performing algorithm parameter values for each instance are reported in Table 4.

5.3. Vignette selection and analysis

Due to the size of the state space, $|\mathcal{S}| = 16,094$, we examined subsets of $\mathcal{S}$ to gain insight into the performance of the ADP algorithm. Each of these subsets can be thought of as a vignette that represents a different system state for the defender, which could alternatively represent a state at the beginning of a missile engagement or a state realized within an engagement. Because the most interesting problem features involve the overlapping SAM coverage of the cities, we only discuss herein vignettes in which all cities are alive. Instead of varying the city status, we consider a small subset of possible interceptor inventories for both SAM batteries (located at City 1 and City 3). We consider the following five interceptor inventory levels for R_t: (10,0,10), (5,0,5), (8,0,2), (2,0,8), and (2,0,2). Thus, there are five vignettes for each instance. Each vignette represents a decision epoch of interest, a collection of system states at which all cities are alive (as indicated by A_t =(1,1,1)), a number of interceptors remaining in inventory (e.g., $R_t = (10, 0, 10)$), and all possible attack vectors (as indicated by $\hat{M}_t$).

5.3.1. Analysis of test instance II vignette results

Instance II has a longer expected conflict duration that will stress a defender's capabilities, and it models a sophisticated attacker (e.g., a more capable one against whom we would most want to be able to defend) who can observe the result of its prior attacks via its own intelligence, surveillance, and reconnaissance assets, and therefore not waste ballistic missiles by firing on cities already destroyed. For these reasons, we closely examine the differences between the optimal and ADP policies for Instance II and compare its policies across the remaining instances.

5.3.1.1. Vignette 1-full interceptor inventories-($R_t = (10, 0, 10)$). Table 5 shows policy results for both the exact and ADP algorithms for each possible attack vector when the defender has 10 interceptors available at each SAM battery. Proceeding from left to right, Column 1 indicates the probability that the attack vector (indicated in Column 2) occurs. Column 2 indicates the attack vector wherein the first element indicates the number of BMs fired at City 1, the second element indicates the number of BMs fired at City 2, and the third element indicates the number of BMs fired at City 3. Columns 3 and 4 indicate the optimal policy, $X^{\pi^*}(S_t)$. Column 3 indicates the decision vector for SAM 1 (located at City 1), wherein the first element indicates the number of interceptors to fire at the first BM, the second element indicates the number of interceptors to fire at the second BM, and the third element indicates the number of interceptors to fire at the third BM. Column 4 indicates the decision vector for SAM 2 (located at City 3) in a similar manner. Note that for attack vectors that indicate an attack against only City 1 (i.e., $\hat{M}_t = (b, 0, 0), b > 0$), only SAM 1 can respond, and therefore the SAM 2 response is a vector of zeros. Likewise, for attacks against only City 3, only SAM 2 can respond. Columns 5 and 6 indicate the ADP policy, $X^{\pi}(S_t|\theta)$, reported in the same manner as the optimal policy. Column 7 indicates the optimal expected total cost J^* attained by implementing the optimal policy in state $S_t = (R_t, A_t, \hat{M}_t)$ at time t and forward in time as the system evolves. Column 8 indicates the expected total cost $\bar{J}^*$ attained by implementing the ADP policy. Column 9 indicates the optimality gap, $(\bar{J}^* - J^*)/J^*$. Tables 6–9 are structured in the same manner, but report the optimal and ADP policies and their attendant expected total costs for different system states.

Overall, when the system is in a state with $A_t = (1, 1, 1)$ and $R_t = (10, 0, 10)$, implementation of the optimal policy results in an expected total loss of 4.32. Implementation of the ADP policy results in an expected total loss of 4.99, for an optimality gap of 15.44%, corresponding to an overall absolute gap of 0.67 from a total city value of 16 at risk.

Table 5
Policy comparison for test instance II, $R_t = (10, 0, 10)$, $A_t = (1, 1, 1)$.

$\mathbb{P}(\hat{M}_t)$	$\hat{M}_t$			Optimal policy, $X^{\pi^*}(S_t)$						ADP policy, $X^{\pi}(S_t\|\theta)$								
Attack probability	Attack vector			SAM 1 response			SAM 2 response			SAM 1 response			SAM 2 response			J^*	$\bar{J}^*$	Optimality gap
0.1042	0	0	1	0	0	0	2	0	0	0	0	0	1	0	0	3.86	4.53	17.41%
0.0326	0	0	2	0	0	0	2	2	0	0	0	0	1	1	0	4.32	5.04	16.69%
0.0102	0	0	3	0	0	0	1	1	2	0	0	0	1	1	1	5.01	5.51	9.96%
0.2083	0	1	0	**2**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	3.88	4.41	13.68%
0.1302	0	1	1	2	0	0	0	2	0	2	0	0	0	1	0	4.32	4.96	14.98%
0.0610	0	1	2	2	0	0	0	2	2	2	0	0	0	1	1	4.81	5.49	14.15%
0.1302	0	2	0	**2**	**2**	**0**	**0**	**0**	**0**	**2**	**2**	**0**	**0**	**0**	**0**	4.34	4.89	12.64%
0.1221	0	2	1	2	2	0	0	0	2	2	2	0	0	0	1	4.82	5.46	13.27%
0.0814	0	3	0	**0**	**2**	**2**	**2**	**0**	**0**	**0**	**2**	**2**	**2**	**0**	**0**	4.86	5.36	10.44%
0.0208	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	3.72	4.22	13.37%
0.0130	1	0	1	1	0	0	2	0	0	1	0	0	0	0	0	4.13	8.16	97.41%
0.0061	1	0	2	1	0	0	2	2	0	1	0	0	0	0	0	4.61	8.16	77.16%
0.0260	1	1	0	**1**	**2**	**0**	**0**	**0**	**0**	**1**	**2**	**0**	**0**	**0**	**0**	4.16	4.65	11.86%
0.0244	1	1	1	1	2	0	0	2	0	1	2	0	0	1	0	4.62	5.22	12.98%
0.0244	1	2	0	1	1	2	1	0	0	0	2	2	0	0	0	4.65	5.54	19.17%
0.0013	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	3.98	4.72	18.55%
0.0012	2	0	1	1	1	0	2	0	0	0	0	0	0	0	0	4.41	8.63	95.59%
0.0024	2	1	0	1	1	2	0	0	0	0	0	2	0	0	0	4.44	5.11	14.97%
0.0001	3	0	0	1	1	1	0	0	0	0	0	0	0	0	0	4.24	4.72	11.20%
																$\mathbb{E}\{J^*\}$	$\mathbb{E}\{\bar{J}^*\}$	$\mathbb{E}\{\text{Gap}\}$
																4.32	4.99	15.44%

Table 6
Policy comparison for test instance II, $R_t = (5, 0, 5)$, $A_t = (1, 1, 1)$.

$\mathbb{P}(\hat{M}_t)$	$\hat{M}_t$			Optimal policy, $X^{\pi^*}(S_t)$						ADP Policy, $X^{\pi}(S_t\|\theta)$								
Attack probability	Attack vector			SAM 1 response			SAM 2 response			SAM 1 response			SAM 2 response			J^*	$\bar{J}^*$	Optimality gap
0.1042	0	0	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	6.38	6.88	7.97%
0.0326	0	0	2	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	7.04	7.53	6.96%
0.0102	0	0	3	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	7.71	8.16	5.75%
0.2083	0	1	0	**2**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	6.41	6.98	8.88%
0.1302	0	1	1	**2**	**0**	**0**	**0**	**1**	**0**	**2**	**0**	**0**	**0**	**1**	**0**	7.09	7.66	8.06%
0.0610	0	1	2	**2**	**0**	**0**	**0**	**1**	**1**	**2**	**0**	**0**	**0**	**1**	**1**	7.77	8.36	7.57%
0.1302	0	2	0	**2**	**2**	**0**	**0**	**0**	**0**	**2**	**2**	**0**	**0**	**0**	**0**	7.21	7.86	9.04%
0.1221	0	2	1	**2**	**2**	**0**	**0**	**0**	**1**	**2**	**2**	**0**	**0**	**0**	**1**	7.96	8.62	8.32%
0.0814	0	3	0	0	2	2	1	0	0	0	2	2	2	0	0	8.18	8.87	8.45%
0.0208	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	6.10	6.62	8.57%
0.0130	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	6.78	9.92	46.44%
0.0061	1	0	2	1	0	0	1	1	0	1	0	0	0	0	0	7.44	9.92	33.36%
0.0260	1	1	0	**1**	**2**	**0**	**0**	**0**	**0**	**1**	**2**	**0**	**0**	**0**	**0**	6.84	7.48	9.39%
0.0244	1	1	1	**1**	**2**	**0**	**0**	**1**	**0**	**1**	**2**	**0**	**0**	**1**	**0**	7.54	8.23	9.05%
0.0244	1	2	0	1	1	2	1	0	0	0	2	2	0	0	0	7.74	8.53	10.30%
0.0013	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	6.50	6.95	6.87%
0.0012	2	0	1	1	1	0	1	0	0	0	0	0	0	0	0	7.19	10.25	42.69%
0.0024	2	1	0	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	7.23	7.68	6.20%
0.0001	3	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6.56	6.95	5.96%
																$\mathbb{E}\{J^*\}$	$\mathbb{E}\{\bar{J}^*\}$	$\mathbb{E}\{\text{Gap}\}$
																7.13	7.78	9.11%

Table 7
Policy comparison for test instance II, $R_t = (8, 0, 2)$, $A_t = (1, 1, 1)$.

$\mathbb{P}(\hat{M}_t)$	$\hat{M}_t$			Optimal policy, $X^{\pi^*}(S_t)$						ADP policy, $X^{\pi}(S_t\|\theta)$								
Attack probability	Attack vector			SAM 1 response			SAM 2 response			SAM 1 response			SAM 2 response			J^*	$\bar{J}^*$	Optimality gap
0.1042	0	0	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	6.92	7.23	4.56%
0.0326	0	0	2	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	8.01	8.25	3.02%
0.0102	0	0	3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	9.23	9.54	3.35%
0.2083	0	1	0	**2**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	6.63	7.07	6.73%
0.1302	0	1	1	**2**	**0**	**0**	**0**	**1**	**0**	**2**	**0**	**0**	**0**	**1**	**0**	7.53	7.87	4.59%
0.0610	0	1	2	**2**	**0**	**0**	**0**	**1**	**1**	**2**	**0**	**0**	**0**	**1**	**1**	8.60	8.87	3.15%
0.1302	0	2	0	**2**	**2**	**0**	**0**	**0**	**0**	**2**	**2**	**0**	**0**	**0**	**0**	7.35	7.89	7.45%
0.1221	0	2	1	**2**	**2**	**0**	**0**	**0**	**1**	**2**	**2**	**0**	**0**	**0**	**1**	8.24	8.70	5.59%
0.0814	0	3	0	1	1	2	0	0	0	0	2	2	2	0	0	8.41	9.36	11.34%
0.0208	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	6.34	6.76	6.53%
0.0130	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	7.26	9.81	35.18%
0.0061	1	0	2	1	0	0	1	1	0	1	0	0	0	0	0	8.35	9.81	17.56%
0.0260	1	1	0	**1**	**2**	**0**	**0**	**0**	**0**	**1**	**2**	**0**	**0**	**0**	**0**	7.02	7.50	6.78%
0.0244	1	1	1	**1**	**2**	**0**	**0**	**1**	**0**	**1**	**2**	**0**	**0**	**1**	**0**	7.91	8.31	5.01%
0.0244	1	2	0	1	1	2	0	0	0	0	2	2	0	0	0	7.96	8.59	7.97%
0.0013	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	6.72	7.18	6.96%
0.0012	2	0	1	1	1	0	1	0	0	0	0	0	0	0	0	7.62	10.28	34.90%
0.0024	2	1	0	1	1	2	0	0	0	0	0	2	0	0	0	7.44	7.82	5.11%
0.0001	3	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6.86	7.18	4.72%
																$\mathbb{E}\{J^*\}$	$\mathbb{E}\{\bar{J}^*\}$	$\mathbb{E}\{\text{Gap}\}$
																7.49	7.99	6.63%

The ADP policy is noticeably more conservative in assigning multiple interceptors to missiles, particularly from the second SAM battery which reflects a fundamental difference between the policies. The optimal policy for this vignette is to fire one interceptor per missile fired at City 1 and fire two interceptors per missile fired at the more valuable Cities 2 or 3. The optimal policy also assigns interceptors primarily from the first SAM battery to counter missiles fired at City 2 unless more than two missiles are inbound. In contrast, the ADP policy is to fire one interceptor per missile fired at City 1 and City 3 and fire two interceptors per missile targeting City 2. Thus, the approximate algorithm appears to undervalue City 3 as compared to the exact algorithm.

We observe that the ADP policy agrees with the optimal policy for five out of 19 states, $\hat{M}_t = (0, 1, 0)$, $(0, 2, 0)$, $(0, 3, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, resulting in optimality gaps of 13.68%, 12.64%, 10.44%, 13.37%, and 11.86%, respectively. These states are represented in bold font in Table 5. Note that our observation that the ADP policy agrees with the optimal policy for selected states in a vignette does not suggest that the agreement holds for subsequent states (and salvos). The ADP policy chooses very poorly for three out of 19 states, $\hat{M}_t = (1, 0, 1)$, $(1, 0, 2)$, $(2, 0, 1)$, resulting in optimality gaps of 97.41%, 77.16%, and 95.59%, respectively. These three states correspond to attack vectors that fire at least one missile at City 1 and City 3 but no missiles at City 2. However, the combined likelihood of these attack vectors is only 0.02, whereas the combined likelihood for the attack vectors associated with the states that are in agreement is 0.47.

One non-intuitive decision resulting from the optimal policy occurs when the attack vector is $\hat{M}_t = (0, 0, 3)$. The optimal policy chooses to counter the first two missiles with one interceptor each while firing two interceptors at the third missile. Based on

Table 8
Policy comparison for test instance II, $R_t = (2, 0, 8)$, $A_t = (1, 1, 1)$.

$\mathbb{P}(\hat{M}_t)$	$\hat{M}_t$			Optimal policy, $X^{\pi^*}(S_t)$						ADP policy, $X^{\pi}(S_t\|\theta)$								
Attack probability	Attack vector			SAM 1 response			SAM 2 response			SAM 1 response			SAM 2 response			J^*	$\bar{J}^*$	Optimality gap
0.1042	0	0	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	6.36	6.96	9.45%
0.0326	0	0	2	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	6.98	7.59	8.78%
0.0102	0	0	3	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	7.58	8.18	7.93%
0.2083	0	1	0	1	0	0	1	0	0	2	0	0	0	0	0	6.40	7.01	9.47%
0.1302	0	1	1	1	0	0	1	1	0	2	0	0	0	1	0	7.08	7.65	7.95%
0.0610	0	1	2	1	0	0	1	1	1	2	0	0	0	1	1	7.75	8.32	7.40%
0.1302	0	2	0	0	1	0	2	1	0	0	2	0	2	0	0	7.21	7.80	8.29%
0.1221	0	2	1	0	1	0	2	1	1	0	2	0	2	0	1	7.96	8.49	6.68%
0.0814	0	3	0	0	0	1	1	2	1	0	0	2	2	2	0	8.18	8.86	8.34%
0.0208	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	6.10	6.68	9.67%
0.0130	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	6.76	10.03	48.30%
0.0061	1	0	2	1	0	0	1	1	0	1	0	0	0	0	0	7.40	10.03	35.53%
0.0260	1	1	0	1	0	0	2	0	0	0	2	0	0	0	0	6.84	7.67	12.20%
0.0244	1	1	1	1	0	0	2	1	0	0	2	0	0	1	0	7.54	8.30	10.01%
0.0244	1	2	0	1	0	0	2	2	0	0	2	0	2	0	0	7.74	13.36	72.66%
0.0013	2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	6.52	6.97	6.93%
0.0012	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	7.18	10.31	43.48%
0.0024	2	1	0	**0**	**0**	**2**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**	7.22	7.67	6.29%
0.0001	3	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	6.54	6.97	6.62%
																$\mathbb{E}\{J^*\}$	$\mathbb{E}\{\bar{J}^*\}$	$\mathbb{E}\{\text{Gap}\}$
																7.12	7.89	10.89%

Table 9
Policy comparison for test instance II, $R_t = (2, 0, 2)$, $A_t = (1, 1, 1)$.

$\mathbb{P}(\hat{M}_t)$	$\hat{M}_t$			Optimal policy, $X^{\pi^*}(S_t)$						ADP policy, $X^{\pi}(S_t\|\theta)$								
Attack probability	Attack vector			SAM 1 response			SAM 2 response			SAM 1 response			SAM 2 response			J^*	$\bar{J}^*$	Optimality gap
0.1042	0	0	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	9.07	9.40	3.59%
0.0326	0	0	2	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	10.20	10.42	2.12%
0.0102	0	0	3	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	10.86	11.31	4.11%
0.2083	0	1	0	1	0	0	0	0	0	2	0	0	0	0	0	9.21	9.83	6.71%
0.1302	0	1	1	1	0	0	0	1	0	2	0	0	0	1	0	10.19	10.64	4.47%
0.0610	0	1	2	1	0	0	0	1	1	2	0	0	0	1	1	11.32	11.94	5.43%
0.1302	0	2	0	1	1	0	0	0	0	0	2	0	2	0	0	10.27	11.65	13.47%
0.1221	0	2	1	1	1	0	0	0	1	0	2	0	1	0	1	11.22	12.04	7.28%
0.0814	0	3	0	0	1	1	1	0	0	0	0	2	1	1	0	11.29	12.10	7.25%
0.0208	1	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	8.82	9.22	4.52%
0.0130	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	9.75	11.62	19.11%
0.0061	1	0	2	0	0	0	1	1	0	1	0	0	0	0	0	10.83	11.62	7.25%
0.0260	1	1	0	0	1	0	0	0	0	0	2	0	0	0	0	9.89	10.42	5.44%
0.0244	1	1	1	0	1	0	0	1	0	0	2	0	0	1	0	10.82	11.21	3.64%
0.0244	1	2	0	0	1	1	0	0	0	0	2	0	2	0	0	10.89	14.60	34.12%
0.0013	2	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	8.85	9.40	6.28%
0.0012	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	9.75	11.87	21.76%
0.0024	2	1	0	0	0	1	0	0	0	0	0	2	0	0	0	9.89	10.42	5.44%
0.0001	3	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	8.85	9.40	6.28%
																$\mathbb{E}\{J^*\}$	$\mathbb{E}\{\bar{J}^*\}$	$\mathbb{E}\{\text{Gap}\}$
																10.16	10.94	7.62%

decisions for other states, we would expect a decision at this state of defending City 3 with two interceptors per missile. This outcome most likely results from our stipulation that a SAM battery may only fire up to four interceptors per salvo.

5.3.1.2. Vignette 2-half full interceptor inventories-($R_t = (5, 0, 5)$). Table 6 shows policy results for both the exact and ADP algorithms for each attack vector when the defender has five interceptors available at each SAM battery. Overall, when the system is in a state with $A_t = (1, 1, 1)$ and $R_t = (5, 0, 5)$, implementation of the optimal policy results in an expected loss of 7.13. Implementation of the ADP policy results in an expected loss of 7.78, corresponding to a relative optimality gap of 9.11% and an absolute gap of 0.65.

With the reduced inventories, the optimal policy switches to firing one interceptor per missile fired at either City 1 or City 3 and firing two interceptors per missile fired at City 2 while still preferring to assign interceptors from the first SAM battery. This policy is the same one closely followed by the ADP policy for the first vignette, i.e., $R_t = (10, 0, 10)$. In fact, the ADP policy for the second vignette, i.e., $R_t = (5, 0, 5)$ is identical to the ADP policy from the first. This change in the optimal policy accounts for a higher number of the identical decisions between the two policies, 13 out of 19 states. These states are represented in bold font in Table 6. The ADP policy performs worst for the same attack vectors as in the previous vignette, $\hat{M}_t = (1, 0, 1), (1, 0, 2), (2, 0, 1)$.

The optimal policy exhibits counterintuitive behavior for the state containing attack vector $\hat{M}_t = (0, 3, 0)$. Instead of firing two interceptors at each missile, it fires two at the first and second missiles, but only one at the third. Conversely, the ADP policy fires two interceptors at each missile, a more intuitive decision. Also observed in this vignette, both the optimal and ADP policies do not defend City 1 against all attack vectors. Out of the 10 attack vec-

tors that target City 1, the optimal policy fires protective interceptors against all except two – $\hat{M}_t = (2, 1, 0), (3, 0, 0)$ – whereas the ADP policy fires interceptors against only five.

5.3.1.3. Vignette 3-SAM 1 inventory high, SAM 2 inventory low-($R_t = (8, 0, 2)$). Table 7 shows policy results for both the exact and ADP algorithms for each possible attack vector when the defender has eight interceptors available at the first SAM but only two available at the second. Overall, when the system is in a state with $A_t = (1, 1, 1)$ and $R_t = (8, 0, 2)$, implementation of the optimal policy results in the expected loss of 7.49. Implementation of the ADP policy results in the expected loss of 7.99, for an optimality gap of 6.63%. The overall absolute gap of 0.50 from a total city value of 16 at risk is the lowest among all vignettes.

With few exceptions, the optimal policy is the same as it was for the second vignette, one interceptor to one missile for City 1 and City 3, two interceptors to one missile for City 2. The ADP policy is identical to that of the first two vignettes with an exception for the state with attack vector $\hat{M}_t = (0, 0, 3)$. In this case, City 3 is being attacked by more missiles than the number of interceptors at the second SAM battery, and the ADP policy correctly chooses to save its interceptors for a possible future engagement.

The optimal policy again exhibits counterintuitive behavior for the state containing attack vector $\hat{M}_t = (0, 3, 0)$, choosing for this vignette to fire one interceptor each at the first two missiles but fire two interceptors at the second missile. As before, the ADP policy still fires two interceptors per missile. The optimal policy also acts counterintuitively for the attack vector $\hat{M}_t = (1, 2, 0)$. For this state, the optimal policy is to fire one interceptor at the first missile, one interceptor at the second missile, and two interceptors at the third. In the second vignette, with an inventory of five interceptors at SAM 2, the optimal policy fires an additional interceptor from SAM 2 at the second missile. With only two interceptors available at SAM 2 in this third vignette, the optimal policy does not add the second interceptor.

Also, for the attack vector $\hat{M}_t = (2, 1, 0)$, owing to an inventory of eight interceptors instead of five at SAM 1, the optimal policy chooses to defend City 1 instead of letting it be destroyed. However, the eight interceptors are not enough for the optimal policy to choose to defend City 1 when the attack vector is $\hat{M}_t = (3, 0, 0)$. This change in the optimal policy from the last vignette results in a policy-to-policy match of 12 out of 19 states (as indicated in bold font in Table 7), with the ADP still making the worst decisions for the same three attack vectors previously identified.

5.3.1.4. Vignette 4-SAM 1 inventory low, SAM 2 inventory high-($R_t = (2, 0, 8)$). Table 8 shows policy results for both the exact and ADP algorithms for each possible attack vector when the defender has two interceptors available at the first SAM battery and eight available at the second. Overall, when the system is in a state with $A_t = (1, 1, 1)$ and $R_t = (2, 0, 8)$, implementation of the optimal policy results in the expected loss of 7.12. Implementation of the ADP policy results in the expected loss of 7.89, for an optimality gap of 10.89%. The overall absolute gap of 0.77 from a total city value of 16 at risk is the highest gap so far, but still of good quality.

The optimal policy is still of the same form as the last two vignettes; however, it now assigns more interceptors from the second SAM battery in defense of City 2. In other words, the number of interceptors being fired at each missile is roughly the same; however, the split of interceptors fired from the SAM batteries is reversed. The optimal policy for this vignette remains counterintuitive for the attack vector $\hat{M}_t = (0, 3, 0)$ in a similar manner as the first three vignettes. Moreover, the optimal policy no longer defends City 1 for attack vectors $\hat{M}_t = (2, 0, 1), (2, 1, 0)$, and (3, 0, 0), allowing City 1 to be destroyed. An example of this policy is observed for attack vector $\hat{M}_t = (2, 1, 0)$. The optimal policy is to have the first SAM battery fire both of its remaining interceptors at the missile inbound to City 2 leaving City 1 undefended and letting SAM 2 conserve its inventory to protect the more valuable City 2 from expected future attack salvos.

The ADP policy observed in this vignette differs for the first time from the previous vignettes. Although it has the same general interceptor-to-missile policy, instead of utilizing SAM 2 to provide additional defense for City 2, it conserves that SAM battery's inventory at the expense of depleting SAM 1. For example, for the state with attack vector $\hat{M}_t = (0, 1, 0)$, instead of firing one interceptor from each SAM battery as the optimal policy does, the approximate policy fires two interceptors from SAM 1 and none from SAM 2, leaving City 1 defenseless against future attacks. Similarly, for attack vector $\hat{M}_t = (1, 1, 0)$, the ADP policy fires two interceptors from SAM 1 at the missile inbound to City 2 while firing none from SAM 2, thus choosing to conserve SAM 2 interceptors for future use rather than protect City 1 during the current attack.

As observed previously, the ADP policy chooses poorly for the three identified states, but for this vignette it also performs poorly for the additional attack vector of $\hat{M}_t = (1, 2, 0)$. The ADP policy fires two interceptors from both SAM batteries at only the first inbound missile to City 2, thereby allowing both City 1 and City 2 to be destroyed. The optimal and ADP policies match exactly for six out of 19 states (as indicated in bold font in Table 8).

5.3.1.5. Vignette 5-low interceptor inventories-($R_t = (2, 0, 2)$). Table 9 shows policy results for both the exact and ADP algorithms for each possible attack vector when the defender has two interceptors available at each SAM battery. When the system is in a state with $A_t = (1, 1, 1)$ and $R_t = (2, 0, 2)$, implementation of the optimal policy results in an expected loss of 10.16. Implementation of the ADP policy results in the expected loss of 10.94, for an optimality gap of 7.62%. The overall absolute gap of 0.78 from a total city value of 16 at risk is the largest gap among the vignettes.

In this vignette, the optimal policy utilizes a one-to-one interceptor-to-missile policy for all cities. Moreover, the optimal policy switches back to having the first SAM battery provide most of the defense for City 2 as is the case for Vignette 1. The optimal policy also makes decisions in this vignette similar to the decisions of the ADP policy for Vignette 4, wherein City 1 is targeted along with City 2 and City 3. The optimal policy only defends City 1 if it is the only city attacked and is attacked with one missile. In all other cases, City 1 is left undefended to provide defensive cover for City 2, either immediately or for subsequent attacks. The ADP policy remains the same as observed in earlier vignettes as much as inventories allow. That is, the ADP policy leaves City 1 unprotected so that it can fire two interceptors per inbound missile to City 2.

The optimal and ADP policies match for 6 out of 19 states (as indicated in bold font in Table 9), but the ADP policy still decides poorly for three out of four of the previously mentioned attack vectors $\hat{M}_t = (1, 0, 1), (1, 0, 2), (2, 0, 1)$. The difference in state value is not as extreme as observed in other vignettes since, with fewer interceptors at each SAM battery, the cities that are initially protected by the optimal policy will likely be destroyed in one or two more attack salvos.

5.3.2. Analysis of instances I–IV – vignette-specific and full state space results

Table 10 reports the mean optimality gaps for the optimal and ADP policies in each of the four instances for the state spaces of each of the five vignettes discussed and the entire state space, using the ADP algorithm settings from Table 4.

Considering all states, the ADP policy performs best in Instance II, attaining a 7.74% mean optimality gap. In Instance II, the attacker is expected to fire five salvos and utilize BDA so as to not waste

Table 10
Mean optimality gaps.

	Vignette					All
Instance	1	2	3	4	5	states
I	14.86%	11.67%	8.71%	11.85%	8.89%	10.96%
II	15.44%	9.11%	6.63%	10.89%	7.62%	7.74%
III	32.72%	20.13%	20.42%	31.20%	17.88%	22.10%
IV	43.40%	29.48%	33.00%	36.47%	21.52%	15.51%

salvos targeting destroyed cities. Over all instances, the mean optimality gap is generally lower when taken over the entire state space. This reflects the better performance of the ADP algorithm for states with smaller inventories and fewer surviving cities. We also observe that the mean optimality gap is higher for instances with a lower expected number of attacker salvos (i.e., Instances III and IV).

6. Conclusions

As the proliferation of offensive and defensive missile systems continues across the world, the optimization of a defensive response to a missile attack remains a valuable national security endeavor for the U.S. and its allies. Given the likelihood that a ballistic missile (BM) engagement would involve more than one missile salvo by an attacker, this paper presented approximate methods for solving a defensive, asset-based variant of the dynamic weapon-target assignment problem. Solving the Markov decision process (MDP) model for large instances requires the design, development, and implementation of an approximate dynamic programming (ADP) algorithm. To demonstrate the applicability of our MDP model and examine the efficacy of the policies produced by our ADP algorithm, we construct a notional, representative military BM defense planning scenario with overlapping interceptor engagement zones.

Across four test instances, when compared to the optimal policy, the ADP policy achieved anywhere from an 8% to 22% mean optimality gap. Moreover, for the vast majority of states in all instances, the state values for the ADP policy fell within 0.5 units of the state values for the exact policy. Analysis also showed that the γ-parameter influenced the fire control policies of both methods more than the attacker's battle damage assessment (BDA) capabilities, and that the ADP policy is invariant as compared to the optimal policy for changes to interceptor inventories, resulting in decision agreement between the two policies from four to 15 out of 19 states in vignettes within the same instance.

The research presented in this paper is of interest to military air and missile defense planners and operators. Our MDP model and ADP solution approach can be utilized to compare fire control policies for defense planning scenarios with fixed surface-to-air-missile (SAM) battery locations. Moreover, planners may evaluate different potential air and missile defense placement strategies to maximize overall performance of the integrated air and missile defense system. Military missile defense planners can make informed decisions about the value of a particular SAM battery given performance characteristics of the interceptor system (e.g., interceptor probability of kill, interceptor magazine depth, weapon engagement range). Military acquisition specialists (i.e., those personnel responsible for implementing new technology in the military) can leverage this model to examine the impact of different interceptor system performance characteristics. Such comparisons inform the future design, development, and purchase of integrated air and missile defense system assets.

Future research could explore the performance of a reasonable baseline fire control policy as compared to the ADP policy developed in this paper. Policies obtained utilizing other ADP algorithms would be of interest as well. Moreover, one could examine problem instances beyond the computational tractability of exact methods. Evaluating larger instances with more cities, greater inventories, and larger attack salvos enables the testing of scalability for the applied approximate algorithms.

Many of the assumptions in this paper could also be eliminated through the inclusion of multiple missile and interceptor types, the introduction of a SAM reload capability, and/or the addition of the partial destruction of cities (i.e., multiple states to model city health). More significant model changes could include incorporating subsequent targeting of missed missiles within the same epoch, i.e., a shoot-look-shoot policy, the development of a more complete missile defense system structure, and the targeting of missile defense system nodes by the attacker.

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 Air Force, the Department of Defense, or the United States Government. The authors would like to thank the coordinating editor and two anonymous referees for their insightful comments and suggestions that have helped improve the presentation of this article.

References

Ahner, D. K., & Parson, C. R. (2015). Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming. *Optimization Letters, 9*(8), 1689–1701.

Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2007). Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research, 55*(6), 1136–1146.

Athans, M. (1987). Command and control (c2) theory: A challenge to control science. *IEEE Transactions on Automatic Control, 32*(4), 286–293.

Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: Theory and algorithms*. John Wiley & Sons.

Bertsekas, D. P. (2011). Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications, 9*(3), 310–335.

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 Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 30*(1), 42–51.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). Neuro-dynamic programming. *Athena Scientific*.

Bisht, S. (2004). Hybrid genetic-simulated annealing algorithm for optimal weapon allocation in multilayer defence scenario. *Defence Science Journal, 54*(3), 395.

Blodgett, D. E., Gendreau, M., Guertin, F., Potvin, J.-Y., & Séguin, R. (2003). A tabu search heuristic for resource management in naval warfare. *Journal of Heuristics, 9*(2), 145–169.

Bogdanowicz, Z. R. (2009). A new efficient algorithm for optimal assignment of smart weapons to targets. *Computers & Mathematics with Applications, 58*(10), 1965–1969.

Bogdanowicz, Z. R., Tolano, A., Patel, K., & Coleman, N. P. (2013). Optimization of weapon–target pairings based on kill probabilities. *IEEE Transactions on Cybernetics, 43*(6), 1835–1844.

Bradford, J. C. (1961). Determination of the optimal assignment of a weapon system to several threats. *Vought Aeronautics, AER-EITM-9*, 14.

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

Brown, W. (2015). *Lockheed Martin wins $1.5 billion contract for Patriot missile work in Camden*. University of Arkansas at Little Rock Public Radio. [Online; accessed 4-August-2015; http://ualrpublicradio.org/post/lockheed-martin-wins-15-billion-contract-patriot-missile-work-camden].

Cai, H., Liu, J., Chen, Y., & Wang, H. (2006). Survey of the research on dynamic weapon-target assignment problem. *Journal of Systems Engineering and Electronics, 17*(3), 559–565.

Davenport, C. (2015). *Patriot, the workhorse of the Pentagon's missile defense system, to get an upgrade*. The Washington Post. [Online; Accessed 04.08.15; https://www.washingtonpost.com/news/checkpoint/wp/2015/02/20/patriot-the-workhouse-of-the-pentagons-missile-defense-system-to-get-an-upgrade/].

Day, R. H. (1966). Allocating weapons to target complexes by means of nonlinear programming. *Operations Research, 14*(6), 992–1013.

Dillow, C. (2015). *U.S. greenlights sale of 600 Patriot missiles to Saudi Arabia*. Fortune. [Online; Accessed 04.08.15; http://fortune.com/2015/08/01/u-s-patriot-missiles-saudi-arabia/].

Eckler, A. R., & Burr, S. A. (1972). Mathematical models of target coverage and missile allocation. *Technical Report*. Military Operations Research Society, Alexandria, VA.

George C. Marshall and Claremont Institutes (2015). *Missiles of the World*. [Online; Accessed 30.07.15; http://missilethreat.com/missiles-of-the-world/].

Global Firepower (2015). *MLRS (Multiple Launch Rocket System) Strength by Country*. [Online; Accessed 30.07.15; http://www.globalfirepower.com/armor-mlrs-total.asp].

Goure, D. (2015). *Directed energy can defeat massed missile salvoes*. Real Clear Defense. [Online; Accessed 30.07.15; http://www.realcleardefense.com/articles/2015/07/11/directed_energy_can_defeat_massed_missile_salvoes_108212.html].

Green, D. J., Moore, J. T., & Borsi, J. J. (1997). An integer solution heuristic for the Arsenal Exchange Model (AEM). *Military Operations Research, 3*(3), 5–15.

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 Journal on Computing, 28*(3), 405–416.

Harress, C. (2015). *Canadian military buys Israeli Iron Dome missile defense technology*. International Business Times. [Online; Accessed 30.07.15; http://www.ibtimes.com/canadian-military-buys-israeli-iron-dome-missile-defense-technology-2030063].

Hosein, P. A., & Athans, M. (1990a). Preferential Defense Strategies. Part I: The Static Case. *Technical Report LIDS-P-2002*. Cambridge, MA: Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Hosein, P. A., & Athans, M. (1990b). Preferential Defense Strategies. Part II: The Dynamic Case. *Technical Report LIDS-P-2003*. Cambridge, MA: Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Hosein, P. A., Walton, J. T., & Athans, M. (1988). Dynamic weapon-target assignment problems with vulnerable C2 nodes. *Technical Report LIDS-P-1786*. Cambridge, MA: Massachusetts Institute of Technology, Laboratory for Information and Decision Systems.

Hoyt, H. C. (1985). A simple ballistic-missile-defense model to help decision makers. *Interfaces, 15*(5), 54–62.

IBC News Bureau (2015). *Long range surface air missile to be test-fired in India*. [Online; Accessed 30.07.15; https://www.ibcworldnews.com/2015/07/25/long-range-surface-air-missile-to-be-test-fired-in-india/].

Jaiswal, N., Shrotri, P., & Nagabhushana, B. (1993). Optimal weapon mix, deployment and allocation problems in multiple layer defense. *American Journal of Mathematical and Management Sciences, 13*(1–2), 53–82.

den Broeder Jr, G., Ellison, R., & Emerling, L. (1959). On optimum target assignments. *Operations Research, 7*(3), 322–326.

Karasakal, O. (2008). Air defense missile-target allocation models for a naval task group. *Computers & Operations Research, 35*(6), 1759–1770.

Karasakal, O., Özdemirel, N. E., & Kandiller, L. (2011). Anti-ship missile defense for a naval task group. *Naval Research Logistics (NRL), 58*(3), 304–321.

Khosla, D. (2001). Hybrid genetic approach for the dynamic weapon-target allocation problem. In *Aerospace/defense sensing, simulation, and controls* (pp. 244–259). International Society for Optics and Photonics.

Kwon, O., Lee, K., Kang, D., & Park, S. (2007). A branch-and-price algorithm for a targeting problem. *Naval Research Logistics (NRL), 54*(7), 732–741.

Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *The Journal of Machine Learning Research, 4*, 1107–1149.

Leboucher, C., Le Menec, S., Kotenkoff, A., Shin, H.-S., & Tsourdos, A. (2013). Optimal weapon target assignment based on an geometric approach. In *Automatic control in aerospace: Vol. 19* (pp. 341–346).

Lee, M. (2010). Constrained weapon–target assignment: Enhanced very large scale neighborhood search algorithm. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40*(1), 198–204.

Lee, Z., Lee, C., & Su, S. (2002). An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. *Applied Soft Computing, 2*(1), 39–47.

Lee, Z., Su, S., & Lee, C. (2003). Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 33*(1), 113–121.

Li, J., Cong, R., & Xiong, J. (2006). Dynamic WTA optimization model of air defense operation of warships' formation. *Journal of Systems Engineering and Electronics, 17*(1), 126–131.

Lloyd, S. P., & Witsenhausen, H. S. (1986). Weapons allocation is NP-complete. In *Proceedings of the 1986 summer simulation conference* (pp. 1054–1058).

Madni, A. M., & Andrecut, M. (2009). Efficient heuristic approach to the weapon-target assignment problem. *Journal of Aerospace Computing, Information, and Communication, 6*(6), 405–414.

Malhotra, A., & Jain, R. (2001). Genetic algorithm for optimal weapon allocation in multilayer defence scenario. *Defence Science Journal, 51*(3), 285–293.

Manne, A. S. (1958). A target-assignment problem. *Operations Research, 6*(3), 346–351.

Matlin, S. (1970). A review of the literature on the missile-allocation problem. *Operations Research, 18*(2), 334–373.

Menq, J., Tuan, P., & Liu, T. (2007). Discrete Markov ballistic missile defense system modeling. *European Journal of Operational Research, 178*(2), 560–578.

Mindock, C. (2015). *US launching missile defense system in Europe, Africa that Russia hates*. International Business Times. [Online; Accessed 30.07.15; http://www.ibtimes.com/us-launching-missile-defense-system-europe-africa-russia-hates-2028707].

Murphey, R. A. (2000a). An approximate algorithm for a weapon target assignment stochastic program. In *Approximation and complexity in numerical optimization* (pp. 406–421). Springer.

Murphey, R. A. (2000b). Target-based weapon target assignment problems. In *Nonlinear assignment problems* (pp. 39–53). Springer.

Opall-Rome, B. (2016). *Pentagon Eyes US Iron Dome To Defend Forward-Based Forces*. Defense News. [Online; Accessed 08.08.16; http://www.defensenews.com/story/defense/international/americas/2016/08/08/skyhunter-tamir-iron-dome-raytheon-rafael-us/88290824/].

Orlin, D. (1987). Optimal weapons allocation against layered defenses. *Naval Research Logistics (NRL), 34*(5), 605–617.

Powell, W. B. (2009). What you should know about approximate dynamic programming. *Naval Research Logistics (NRL), 56*(3), 239–249.

Powell, W. B. (2011). *Approximate dynamic programming: Solving the curses of dimensionality* ((2nd ed.)). John Wiley & Sons.

Powell, W. B. (2012). Perspectives of approximate dynamic programming. *Annals of Operations Research, 13*(2), 1–38.

PR Newswire (2015). *Republic of Korea upgrades its air and missile defense*. [Online; Accessed 04.08.15; http://www.prnewswire.com/news-releases/republic-of-korea-upgrades-its-air-and-missile-defense-300057545.html].

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Hoboken, NJ: John Wiley & Sons.

Raytheon (2015). *Global Patriot Solutions*. [Online; Accessed 04.08.15; http://www.raytheon.com/capabilities/products/patriot/].

Reuters (2015). *Turkey open to new bid from China in controversial missile defense tender*. [Online; Accessed 30.07.15; http://national.bgnnews.com/turkey-open-to-new-bid-from-china-in-controversial-missile-defense-tender-haberi/8049].

Roux, J. N., & Van Vuuren, J. H. (2007). Threat evaluation and weapon assignment decision support: A review of the state of the art. *ORiON, 23*(2), 151–187.

Sikanen, T. (2008). *Solving weapon target assignment problem with dynamic programming*. Online; Accessed 31.07.16; http://salserver.org.aalto.fi/vanhat_sivut/Opinnot/Mat-2.4108/pdf-files/esik08b.pdf.

Soland, R. M. (1987). Optimal terminal defense tactics when several sequential engagements are possible. *Operations Research, 35*(4), 537–542.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.

The Jerusalem Post (2015). *Defense Ministry: Upgrades for Iron Dome's operational capabilities a success*. [Online; Accessed 30.07.15; http://www.jpost.com/Israel-News/Defense-Ministry-Upgrades-for-Iron-Domes-operational-capabilities-a-success-407827].

Tomkins, R. (2014). *Raytheon given $2.4B FMS contract for Patriot fire units*. United Press International. [Online; Accessed 04.08.15; http://www.upi.com/Business_News/Security-Industry/2014/12/23/Raytheon-given-24B-FMS-contract-for-Patriot-fire-units/8031419329356/].

U.S. Joint Chiefs of Staff (2014). *Joint Publication 3-01: Countering air and missile threats*.

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: Vol. 4* (pp. 4052–4057). IEEE.

Wacholder, E. (1989). A neural network-based optimization algorithm for the static weapon-target assignment problem. *ORSA Journal on Computing, 1*(4), 232–246.

Wang, Y., Qian, L., Guo, Z., & Ma, L. (2008). Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm. *Journal of Systems Engineering and Electronics, 19*(5), 939–944.

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. (IEEE world congress on computational intelligence)* (pp. 2020–2025). IEEE.

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 Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 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 Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 40*(6), 649–662.

Zeng, X., Zhu, Y., Nan, L., Hu, K., Niu, B., & He, X. (2006). Solving weapon-target assignment problem using discrete particle swarm optimization. In *The sixth world congress on intelligent control and automation: Vol. 1* (pp. 3562–3565). IEEE.