



Approximate dynamic programming for the military inventory routing problem

Rebekah S. McKenna¹ · Matthew J. Robbins¹ · Brian J. Lunday¹ · Ian M. McCormack¹

Published online: 2 December 2019

© This is a U.S. government work and not under copyright protection in the U.S.; foreign copyright protection may apply. 2019

Abstract

The United States Army can benefit from effectively utilizing cargo unmanned aerial vehicles (CUAVs) to perform resupply operations in combat environments to reduce the use of manned (ground and aerial) resupply that incurs risk to personnel. We formulate a Markov decision process (MDP) model of an inventory routing problem (IRP) with vehicle loss and direct delivery, which we label the military IRP (MILIRP). The objective of the MILIRP is to determine CUAV dispatching and routing policies for the resupply of geographically dispersed units operating in an austere, combat environment. The large size of the problem instance motivating this research renders dynamic programming algorithms inappropriate, so we utilize approximate dynamic programming (ADP) methods to attain improved policies (relative to a benchmark policy) via an approximate policy iteration algorithmic strategy utilizing least squares temporal differencing for policy evaluation. We examine a representative problem instance motivated by resupply operations experienced by the United States Army in Afghanistan both to demonstrate the applicability of our MDP model and to examine the efficacy of our proposed ADP solution methodology. A designed computational experiment enables the examination of selected problem features and algorithmic features vis-à-vis the quality of solutions attained by our ADP policies. Results indicate that a 4-crew, 8-CUAV unit is able to resupply 57% of the demand from an 800-person organization over a 3-month time horizon when using the ADP policy, a notable improvement over the 18% attained using a benchmark policy. Such results inform the development of procedures governing the design, development, and utilization of CUAV assets for the resupply of dispersed ground combat forces.

Keywords Inventory routing problem · Markov decision processes · Approximate dynamic programming · Least squares temporal differences · Military

✉ Matthew J. Robbins
matthew.robbins@afit.edu

¹ Department of Operational Sciences, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, USA

1 Introduction

The United States (U.S.) Army utilizes vendor-managed inventory practices to direct resupply operations when engaged in combat operations. Upper-echelon organizations monitor the inventory levels of lower-echelon organizations and manage resupply efforts, deciding when to conduct resupply operations and how to route the vehicles transporting supplies to subordinate unit locations. At the tactical level for an infantry brigade combat team (IBCT) consisting of approximately 4500 personnel, a subordinate brigade support battalion (BSB) plans and conducts all resupply operations for combat outposts (COPs) within its area of operations (AO). During combat operations in Afghanistan over the most recent decade, this typically includes the support of as many as 16 COPs, each of which serves as an operational base for a company-sized infantry unit consisting of approximately 150 personnel, and which requires approximately 25,000 pounds of supplies per day among the categories of subsistence, construction items, ammunition, medical supplies, repair parts, and fuel (General Dynamics Information Technology 2010). The BSB is kept informed of inventory levels at COPs through regular, automated and manual reporting. Thus, the relationship between an IBCT and the COPs it supports parallels the supplier-to-customer relationships seen in vendor-managed inventory replenishment practices.

Specific to our study, military resupply efforts supporting combat operations pose a significant risk to both personnel and the supplies being transported. The Army typically operates in harsh, rugged environments that include mountains, deserts, and jungles. Resupply efforts are heavily reliant on ground lines of communication (GLOC), and a lack of transportation infrastructure in austere environments combined with attacks from enemy forces make GLOC resupply inherently dangerous and difficult. Improvised explosive devices (IED) accounted for 65% of U.S. deployed fatalities between November 2002 and March 2009, with 18% occurring during sustainment operations (General Dynamics Information Technology 2010). As an alternative to ground-based resupply, manned aircraft partially fulfill the resupply role but also have risk-based limitations. Pilots cannot fly in hazardous weather conditions, and helicopters are vulnerable to man-portable air defense systems (MANPADS), especially during takeoff and landing at the COPs, closer to where more combat operations are likely to occur. Although manned cargo aircraft may be escorted by armed aircraft to mitigate the MANPADS risk, a high operational tempo of combat combined with limited air assets may result in the prioritization of armed aircraft to support combat missions over resupply missions, and a sufficiently high MANPADS risk may preclude the use of manned resupply aircraft, even with armed escort. Other factors that reduce the ability of manned cargo aircraft to provision subordinate units include weather-induced hazardous flying conditions, crew fatigue, and commander-driven operational restrictions (e.g., no nighttime aircraft resupply missions).

The U.S. military is considering the use of rotary-wing cargo unmanned aerial vehicles (CUAVs) to resupply subordinate units for the advantages they offer. Foremost, a dedicated contingent of CUAVs for resupply can offset the demand for ground-based and/or manned aerial resupply. In turn, this offset reduces the overall risk to personnel and avails manned aircraft for combat mission support. Moreover, the CUAV's higher flight ceiling compared to manned helicopters and better performance in adverse conditions reduces MANPADS threats to the resupply mission; provides a quicker, more reliable, and more flexible delivery platform; and possibly allows for shorter supply routes.

A recent study by Williams (2010) examines the use of unmanned airlift at both the theater and direct delivery levels in Department of Defense (DoD) applications, as motivated

by frequent requests from military units for unmanned aerial systems, and the study identifies combat outpost support as an application for which CUAVs can make a noteworthy impact. The DoD *Unmanned System Integrated Roadmap* outlines strategic and tactical unmanned aircraft development and acquisition goals through 2028, and it identifies resupply as a potential role for both aerial and ground unmanned systems (Department of Defense 2009). The Army also sponsored a General Dynamics study on unmanned aircraft in resupply roles (General Dynamics Information Technology 2010), resulting in a recommendation to centrally manage CUAVs within operational environments to increase effectiveness in accomplishing multiple, disparate missions. With a view toward operational testing, the U.S. Marine Corps utilized three Lockheed Martin Kaman K-MAX (Lockheed Martin 2010, 2012) CUAVs in Afghanistan between 2011 and 2014 (Lamothe 2014; Lockheed Martin 2018); the K-MAX met DoD requirements during testing and demonstrated its capability for use in a combat zone. As the technical development of capable CUAVs progresses, so too does the need to develop policies for their effective use.

However, CUAVs are not a panacea for subordinate unit resupply in a combat environment. CUAVs remain vulnerable to hostile enemy action and can be lost during a resupply effort. Whereas the loss of a CUAV is preferable to a manned resupply by ground or air, a fleet of CUAVs is a finite commodity and must be managed to balance the priorities of current and future supply operations. Moreover, the development of effective CUAV dispatching and routing policies requires that scenario-specific challenges be addressed. Policies must account for large supply quantity demands across an area of operations; the threat due to enemies; the risk incurred by weather, terrain, and poor infrastructure; the availability of distribution assets; and the flexibility to respond to changes in the operational environment.

A centralized BSB resupplying an IBCT's COPs utilizing a finite set of CUAVs (that can be destroyed) is an instantiation of a military-oriented inventory routing problem (IRP) with vehicle loss and direct delivery (denoted as an MILIRP), which we formulate and examine herein. In the MILIRP, the logistics decision-making authority must decide when to dispatch and how to route CUAVs to the respective COPs that require supplies. Routed inventory is not guaranteed to reach its destination; a BSB must consider the prospect of failed deliveries and destroyed CUAVs under evolving threat conditions. Moreover, the long-term impact of potential CUAV losses on future resupply capability must be considered when deciding to send CUAVs on resupply missions.

Our research is informed in its modeling by published literature on the IRP and in its solution methodology by work on approximate dynamic programming (ADP) methods. The IRP seeks to provide answers to three questions: (1) in which time periods should each customer be served, (2) what amount of supplies should be delivered to each of these customers, and (3) how should customers be combined into vehicle routes. Coelho et al. (2012) identify key structural components of the IRP: time horizon, supplier-customer structure, vehicle routing, inventory policy, fleet composition, fleet size, and demand type. Utilizing this taxonomy, the MILIRP has the following characteristics: an infinite time horizon, one-to-many structure, direct routing, homogeneous fleet composition, limited fleet size, deterministic demand, and stochastic supply.

A particular nuance of the MILIRP is that vehicles can be destroyed while traveling to and from the supplier, which imposes a stochastic nature on the supply. Vehicle routing problems (VRPs) with vehicle breakdown have a similar complexity in a civilian context. Mu et al. (2010) solve a variant of the VRP in which a new routing solution must be created in the event of a vehicle breakdown. The authors develop two metaheuristics that focus on rescheduling the route in an allotted time with a single extra vehicle available for use in the event of a breakdown. However, the Mu et al. (2010) formulation differs fundamentally from

the MILIRP in that the authors solve the re-optimization in only a single time period. The MILIRP must be solved over an infinite time horizon.

Our methods to solve the MILIRP are informed by approximate dynamic programming research. Inventory routing decisions must be made sequentially over time and under uncertainty. Since such resupply decisions impact the capability of the sustainment system to serve future demand, we must account for how current decisions affect the future state of the system. As such, we formulate a Markov decision process (MDP) model of the MILIRP. Unfortunately, due to the well-known curses of dimensionality, an optimal policy cannot be identified using classical exact dynamic programming algorithms. Instead, we employ an ADP solution methodology to solve the MILIRP. For an introduction to ADP, we refer the reader to Powell (2011) and Bertsekas (2012, 2017).

Two general algorithmic strategies exist for obtaining approximate solutions to our computational stochastic optimization problem: approximate value iteration (AVI) and approximate policy iteration (API). The interested reader is referred to Bertsekas (2011) for a detailed discussion concerning API. We utilize an API algorithmic strategy to obtain a policy that maps the system state (e.g., status of COP inventories, number of operational CUAVs remaining, threat map) to a decision (e.g., dispatching a number of fully loaded CUAVs to deliver supplies to COPs). API avoids some of the stability challenges that can accompany an AVI approach. For example, AVI often results in a noisy update of the coefficient vector within the approximation model. This noisy update impacts the stability of the computed policies by inducing frequent and possibly large policy changes, which subsequently contribute to further noise. API avoids this instability issue by performing batch updates after repeated simulations of a fixed policy.

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 improved routing policies (relative to a benchmark policy) based on value function approximations. Our approximation strategy involves the design of an appropriate set of basis functions for application within a linear architecture. Moreover, we approximate the value function around the post-decision state. First introduced by Van Roy et al. (1997), the post-decision state variable convention allows for modification of Bellman's equation to obtain an equivalent, deterministic expression, and it addresses the curse of dimensionality with respect to the outcome space. Within the policy evaluation step of our API algorithm, we update the value function approximation for a fixed policy utilizing least squares temporal differencing (LSTD). Introduced by Bradtke and 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.

In the intersection of inventory routing problem models and approximate dynamic programming solution methods are recent works by Kleywegt et al. (2002) and Kleywegt et al. (2004). The authors' modeling and solution methods greatly informed the development of this paper. Kleywegt et al. (2002) formulate a direct-delivery stochastic inventory routing problem as an MDP. In particular, the states of the system are the inventory levels at each customer's location, and the action space includes the amount of inventory delivered to each customer. The state of the system at a given decision epoch depends on the amount of inventory delivered, the probabilistic demand, and the supply capacity of the customer in the previous epoch. Contributions are based on the traveling costs of the vehicles, shortage costs, holding costs, and revenue. Given the large state space, Kleywegt et al. (2002) develop an ADP algorithm to solve the IRP with direct delivery and stochastic demand for an infinite horizon problem

with homogeneous vehicles and no backlogging. Kleywegt et al. (2004) extend the work of Kleywegt et al. (2002) by removing the direct delivery constraint; a vehicle can make up to three stops at different customers before returning to the supplier. Relaxation of the direct delivery constraint requires the consideration of larger state and action spaces to account for available routes and assignment of routes to each vehicle. Due to the large size of their problem, an exact solution to the MDP is computationally intractable, so Kleywegt et al. (2004) further develop the ADP from Kleywegt et al. (2002) to determine an approximate policy.

This research makes the following contributions. We formulate an MDP model of the MILIRP, a novel problem not previously studied in the literature. We apply ADP techniques to attain improved routing policies (relative to a benchmark policy) to solve the MILIRP because the high-dimensional state space of the MDP model renders classical dynamic programming methods inappropriate. This ADP approach leverages an API algorithm that utilizes LSTD for policy evaluation, and it constructs a set of basis functions within a linear architecture to approximate the value function around the post-decision state. We demonstrate the applicability of our MDP model and the efficacy of our proposed ADP solution methodology using a synthetic, representative planning scenario for contingency operations in Afghanistan, for which we conduct a designed computational experiment to determine how selected problem features and algorithmic features affect the quality of solutions attained by our ADP policies.

The remainder of this paper is organized as follows. Section 2 describes the MDP model formulation of the military inventory routing problem and presents our ADP approach. In Sect. 3, we demonstrate the applicability of our model and examine the efficacy of our proposed solution methodology to a representative instance, as well as the effect of selected parameters on solution quality. In Sect. 4, we conclude the work and indicate directions in which to extend the research.

2 Model formulation and solution methodology

This section describes the MDP model of the MILIRP, followed by the ADP methodology we utilize to obtain improved solutions to the problem.

2.1 MDP formulation

An IBCT is responsible for the COPs within its AO. The IBCT contains a BSB, which manages resupply efforts. The BSB manages a fleet of identical CUAVs to deliver supplies to the COPs.

Figure 1 provides a geographic illustration. Each COP requires a deterministic amount of supplies per time period, a demand that depends on the size of the unit at the COP. We assume that each CUAV is fully loaded when conducting resupply missions. Only direct deliveries are considered; each CUAV delivers to only one COP per resupply sortie. This formulation reduces the complexity of the problem and reflects the fact that current rotary-wing assets do not typically combine multiple deliveries. Given the austere combat environment, there is a potential for delivery failure due to factors such as hostile actions by non-friendly forces, mechanical failures, and extreme weather conditions. A set of threat maps is created to capture the inherent risk to resupply operations within the IBCT's AO. The MDP model is formulated as follows.

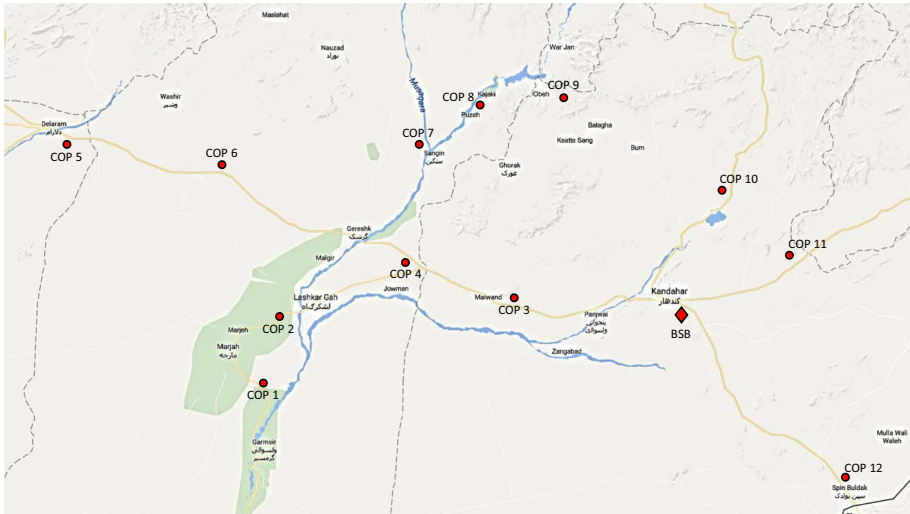


Fig. 1 Example instance wherein 12 COPs dispersed throughout southern Afghanistan require resupply from the BSB located at Kandahar Airfield

Let $\mathcal{T} = \{1, 2, \dots\}$ be the set of decision epochs. A decision epoch occurs at the beginning of a 6-h time period of resupply operations. During a single time period, a CUAV is fueled, loaded with supplies, travels from the BSB to the COP, unloads its supplies, and returns to the BSB. We assume that a fully loaded CUAV can serve any single COP within the AO and return to the BSB during a single time period.

The state space includes three components: the inventory status of each COP, the number of operational CUAVs, and the index corresponding to a given threat map, which represents the current risk level for CUAV resupply operations due to weather and enemy threats. The COP inventory status component is defined as

$$R_t = (R_{ti})_{i \in \mathcal{B}} \equiv (R_{t1}, R_{t2}, \dots, R_{tB}),$$

where $\mathcal{B} = \{1, 2, \dots, B\}$ is the set of all COPs (i.e., small, forward operating bases), $R_{ti} \in \{1, 2, \dots, R_i^{cap}\}$ is the number of tons of supplies at COP $i \in \mathcal{B}$ at time t , and $R_i^{cap} \in \mathbb{N}$ is the inventory capacity of COP $i \in \mathcal{B}$. The number of CUAVs able to perform resupply operations at time t is defined as $V_t \in \{0, 1, \dots, V^{init}\}$, where $V^{init} \in \mathbb{N}$ is the initial number of operational CUAVs. The threat map index number at time t is defined as $\hat{M}_t \in \{1, 2, \dots, M\}$, where $M \in \mathbb{N}$ is the number of threat maps utilized to model the security situation in the BCT’s AO. The threat map impacts the flight risks associated with successfully completing sorties to (from) a COP from (to) the BSB. The threat information provided by \hat{M}_t is available to the BSB at time t . However, the arrival of new information, \hat{M}_{t+1} , is unknown at time t . Moreover, the threat map at time $t + 1$ is conditioned on the threat map at time t . Utilizing these components, we define $S_t = (R_t, V_t, \hat{M}_t) \in \mathcal{S}$ as the state of the system at time t , where \mathcal{S} is the set of all possible states.

At each decision epoch $t \in \mathcal{T}$, the BSB must decide how many fully loaded CUAVs to dispatch and route to each COP. If a CUAV is destroyed by enemy combatants or otherwise malfunctions, it is no longer available for resupply operations. As such, the BSB can only route up to $\min\{V_t, \kappa\}$ CUAVs, where κ is the maximum number of CUAVs simultaneously

controllable by the BSB in a single time period. This constraint is governed by the size of the CUAV organizational element (i.e., ground station crew) within the BSB. A κ -crew element is only able to control κ CUAVs simultaneously. Let $x_{ti} \in \mathbb{N}^0$ be the number of CUAVs dispatched from the BSB to COP $i \in \mathcal{B}$ at time t , and let $x_t = (x_{ti})_{i \in \mathcal{B}}$ denote the corresponding decision vector. We define the set of all feasible BSB decisions as

$$\mathcal{X}_{S_t} = \left\{ x_t : \sum_{i \in \mathcal{B}} x_{ti} \leq \min \{V_t, \kappa\} \right\}, \forall S_t \in \mathcal{S}, t \in T$$

where the constraint $\sum_{i \in \mathcal{B}} x_{ti} \leq \min \{V_t, \kappa\}$ ensures that the total number of CUAVs routed does not exceed the number of operational CUAVs available and does not exceed the maximum number of CUAVs controllable due to CUAV ground station crew limitations.

The trajectory of the system is given by $\{(S_t, x_t) : t = 1, 2, \dots\}$. Transition functions characterize how the system evolves from one state to the next as a result of both decisions and revealed information. The state transition function is defined as $S_{t+1} = S^M(S_t, x_t, W_{t+1})$, wherein $W_{t+1} = (\hat{V}_{t+1}, \hat{M}_{t+1})$ represents the information (i.e., CUAV resupply mission results and threat map) that becomes known at time $t + 1$. We let

$$\hat{V}_{t+1} = \left(\left(\hat{V}_{t+1,i}^{BSB}(x_{ti}) \right), \left(\hat{V}_{t+1,i}^{COP}(x_{ti}) \right), \left(\hat{V}_{t+1,i}^{enroute}(x_{ti}) \right) \right), i \in \mathcal{B},$$

denote the results of the CUAV routing decision, wherein \hat{V}_{t+1} follows a multinomial distribution with parameters x_{ti} and $\left(\left(\psi_{i,\hat{M}_t} \right)^2, \psi_{i,\hat{M}_t} \left(1 - \psi_{i,\hat{M}_t} \right), 1 - \psi_{i,\hat{M}_t} \right)$. The random variable $\hat{V}_{t+1,i}^{BSB}(x_{ti})$ is a random variable representing the number of CUAV sorties that results in a CUAV successfully delivering its load of supplies to COP i and then returning to the BSB. The random variable $\hat{V}_{t+1,i}^{COP}(x_{ti})$ follows a binomial distribution with parameters x_{ti} and $(\psi_{i,\hat{M}_t})^2$. The parameter ψ_{ij} denotes the one-way probability that a single CUAV successfully travels from (to) the BSB to (from) COP i during threat conditions indicated by map $j = 1, 2, \dots, M$. The random variable $\hat{V}_{t+1,i}^{COP}(x_{ti})$ represents the number of CUAV sorties that results in a CUAV successfully delivering its load of supplies to COP i but then failing to return to the BSB. The random variable $\hat{V}_{t+1,i}^{COP}(x_{ti})$ follows a binomial distribution with parameters x_{ti} and $\psi_{i,\hat{M}_t} (1 - \psi_{i,\hat{M}_t})$. The random variable $\hat{V}_{t+1,i}^{enroute}(x_{ti})$ represents the number of CUAV sorties that results in a CUAV being destroyed (or lost due to inclement weather or mechanical failure) enroute to COP i and failing to deliver its supplies. The random variable $\hat{V}_{t+1,i}^{enroute}(x_{ti})$ follows a binomial distribution with parameters x_{ti} and $(1 - \psi_{i,\hat{M}_t})$. The information modeled by these three random variables depends on the routing decision x_t since the number of CUAVs surviving their routes, by outcome type, depends on x_{ti} , i.e., the number of CUAVs the BSB decides to route to COP i .

We define the inventory status transition function as

$$R_{t+1,i} = \begin{cases} R_i^{cap} & \text{if } R_{ti} + \eta(\hat{V}_{t+1,i}^{BSB}(x_{ti}) + \hat{V}_{t+1,i}^{COP}(x_{ti})) \leq d_i \\ \min \{R_{ti} - d_i + \eta(\hat{V}_{t+1,i}^{BSB}(x_{ti}) + \hat{V}_{t+1,i}^{COP}(x_{ti})), R_i^{cap}\} & \text{otherwise} \end{cases}$$

The parameter $\eta \in \mathbb{N}$ represents the number of tons of supplies a fully loaded CUAV carries. The parameter $d_i \in \mathbb{N}$ represents the deterministic single-period demand of COP i . The first condition captures the transition when all remaining supplies at COP i will be consumed because no resupply via CUAV is forthcoming. In such a situation, we assume the BSB orders an immediate resupply operation via ground convoy, which results in the COP receiving supplies up to its capacity R_i^{cap} . The second condition captures all other transitions. COP i 's next inventory level, $R_{t+1,i}$, results from its single-period consumption

of supplies, d_i , and the total delivery of supplies from CUAVs, $\eta(\hat{V}_{t+1,i}^{BSB}(x_{ti}) + \hat{V}_{t+1,i}^{COP}(x_{ti}))$. Only CUAVs that successfully reach the COP (as indicated by $\hat{V}_{t+1,i}^{BSB}(x_{ti})$ and $\hat{V}_{t+1,i}^{COP}(x_{ti})$) are able to deliver supplies. Moreover, the COP's inventory cannot exceed its capacity, R_i^{cap} .

We define the CUAV inventory transition function as

$$V_{t+1} = V_t - \left(\sum_{i \in \mathcal{B}} \hat{V}_{t+1,i}^{COP}(x_{ti}) + \hat{V}_{t+1,i}^{enroute}(x_{ti}) \right),$$

where the number CUAVs available at time $t + 1$ is simply the number of CUAVs available at time t less those that do not successfully return to the BSB, as captured by $\hat{V}_{t+1,i}^{COP}(x_{ti})$ and $\hat{V}_{t+1,i}^{enroute}(x_{ti})$. CUAVs that are lost cannot be used in future resupply efforts.

The threat conditions within the operational environment evolve in an uncontrolled, stochastic manner over time. The m maps capture the threat conditions in the IBCT's AO. Whereas some maps represent a low threat environment with relatively higher ψ_{ij} -values, other maps present a high threat environment with lower ψ_{ij} -values. In high threat maps, delivery of supplies via CUAV becomes increasingly risky, and the BSB must balance current resupply needs with the ability to perform future resupply operations via CUAV. The map transition represents an ever changing threat environment within the AO. We assume the next period's threat map, $\hat{M}_{t+1}(\hat{M}_t)$, depends on the current period's threat map. If the threat environment is relatively static, the transition probabilities between different maps would be relatively low whereas, if the operational environment changes rapidly between high and low threats, the transition probabilities would be relatively high. It is conceivable that the transition probabilities could be constructed in a variety of ways. For example, IBCT intelligence teams working with operational and logistical personnel within the BSB would conduct risk assessments to label subregions of the AO as high-, medium-, or low-risk based on information such as enemy disposition, weather, and season (which affects the stability of weather and both the frequency and intensity of combat operations). Information about mechanical failures of the CUAVs or ground crew station equipment reliability may also be captured in this risk assessment. Historical data from enemy engagements and weather conditions could also be leveraged to inform the development of appropriate threat maps.

At each decision epoch t , the BSB obtains an expected immediate contribution (i.e., reward) as a result of its routing decision. We define this contribution as

$$\hat{C}(S_t, x_t, \hat{V}_{t+1}) = \sum_{i \in \mathcal{B}} \min \left\{ R_i^{cap} - R_{ti} + d_i, \eta(\hat{V}_{t+1,i}^{BSB}(x_t) + \hat{V}_{t+1,i}^{COP}(x_t)) \right\}.$$

The BSB is rewarded for supplies delivered via air line of communication (ALOC). That is, each ton of supplies delivered by CUAV provides a reward. The BSB is *not* rewarded if a COP cannot receive the supplies delivered to it (i.e., a COP is at inventory capacity), *nor* is the BSB rewarded for delivering supplies via GLOC. We can express the contribution function in terms of the current state and decision as follows

$$C(S_t, x_t) = \mathbb{E} \left[\sum_{i \in \mathcal{B}} \min \left\{ R_i^{cap} - R_{ti} + d_i, \eta(\hat{V}_{t+1,i}^{BSB}(x_t) + \hat{V}_{t+1,i}^{COP}(x_t)) \right\} \middle| S_t, x_t \right].$$

The objective is to determine the policy π^* that maximizes the expected total discounted reward and is expressed as

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

Let $x_t = X^\pi(S_t)$ represent the decision function, or policy, that returns decision x_t given state S_t . The π superscript emphasizes the fact that $X^\pi(S_t)$ is one element in a family of functions, Π . To attain the optimal policy, we must determine a solution to the Bellman Equation

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

wherein $\gamma \in [0, 1)$ is the discount factor. Unfortunately, attaining such a solution is computationally intractable utilizing classical dynamic programming algorithms (e.g., value iteration and policy iteration). As such, we employ approximate dynamic programming methods to attain sub-optimal, yet improved solutions relative to those attainable via a myopic, benchmark policy.

2.2 ADP formulation

We employ an API algorithmic strategy to construct improved CUAV resupply policies based on value function approximations. To obtain such policies we must determine approximate solutions to Eq. (1). We proceed by employing a modified version of the optimality equation that uses a *post-decision state* variable convention. The post-decision state S_t^x refers to the state of the system after being in state S_t and taking action x_t . The post-decision state variable provides tremendous computational advantages, as its use eliminates the embedded expectation within the optimality equation (Powell 2011; Ruszczyński 2010). The value of being in pre-decision state S_t is denoted by $J(S_t)$, and the value of being in post-decision state S_t^x is denoted by $J^x(S_t^x)$. The relationship between $J(S_t)$ and $J^x(S_t^x)$ is given by

$$J^x(S_t^x) \triangleq \mathbb{E} [J(S_{t+1}) | S_t^x]. \tag{2}$$

The optimality equation in terms of the post-decision state variable is

$$J^x(S_{t-1}^x) = \mathbb{E} \left[\max_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)) \middle| S_{t-1}^x \right]. \tag{3}$$

Although utilization of the post-decision state variable convention provides a computational benefit, the optimality equation remains intractable due to dimensionality challenges. We proceed by defining a fixed set of basis functions to approximate the post-decision state value function, $J_t^x(S_t^x)$. Let $\phi_f(S_t^x)$ be a *basis function*, where $f \in \mathcal{F}$ is a *feature* and \mathcal{F} is the set of features. Identification of features that are important to a particular problem can be difficult but is important to obtaining an accurate approximation. Well-crafted features can capture the dominant nonlinearities of the value function, and the linear combination of the features can work well as an approximation architecture (Bertsekas 2011). Equation (4) shows the linear approximation architecture we adopt. Let

$$\bar{J}^x(S_t^x) \triangleq \sum_{f \in \mathcal{F}} \theta_f \phi_f(S_t^x) = \theta^\top \phi(S_t^x), \tag{4}$$

wherein $\phi(S_t^x)$ is a column vector with elements $\{\phi_f(S_t^x)\}_{f \in \mathcal{F}}$, and θ is a column vector of basis function weights. By substituting the value function approximation, Eq. (4), into the modified optimality equation, Equation (3), we obtain the following expression for the value function approximation

$$\bar{J}^x(S_{t-1}^x) = \mathbb{E} \left[\max_{x_t \in \mathcal{X}_{S_t}} (C(S_t, x_t) + \gamma \bar{J}^x(S_t^x)) \middle| S_{t-1}^x \right]. \tag{5}$$

We refer to the component of the optimality equation inside the expectation operator as the inner maximization problem. In terms of the basis functions, we obtain the following equivalent expression for the value function approximation

$$\theta^\top \phi(S_{t-1}^x) = \mathbb{E} \left[C(S_t, X^\pi(S_t|\theta)) + \gamma \theta^\top \phi(S_t^x) \mid S_{t-1}^x \right], \tag{6}$$

where, for a given θ -vector, the decision function is given by

$$X^\pi(S_t|\theta) = \arg \max_{x_t \in \mathcal{X}_{S_t}} \left(C(S_t, x_t) + \gamma \theta^\top \phi(S_t^x) \right). \tag{7}$$

Selection of the set of basis functions is an important aspect of our ADP approach and requires deliberate development as it directly impacts both the quality of the value function approximation and the convexity properties of the inner maximization problem. Approximating high-dimensional value functions is fundamentally intractable. As we increase the order of our value function approximation so as to obtain a higher-quality approximation, we must estimate an increasing number of parameters. For example, if we desire an n th order value function approximation with all interaction terms, we must estimate $|S|^{|S|}$ θ -component values. Moreover, the use of higher-order basis functions renders the inner maximization problem nonconvex, making it much more challenging to solve. Since we must solve the inner maximization problem many times when implementing our API algorithm, it is desirable to select basis functions that allow fast and efficient determination of a solution. Accordingly, we utilize a set of first-order basis functions with interactions, excluding interaction terms that result in a bilinear term in the objective function. This approximation approach allows us to model the inner maximization problem as an integer linear program.

The first-order basis functions are as follows. The first basis function captures the inventory level of each COP and is written as

$$\phi_{1i}(S_t^x) = R_{ti}, \quad \forall i \in \mathcal{B}.$$

The second basis function represents the number of remaining operational CUAVs and is written as

$$\phi_2(S_t^x) = V_t.$$

The third basis function indicates the current map, implicitly capturing the threat condition level associated with the map, and is written as

$$\phi_{3m}(S_t^x) = \begin{cases} 1, & \text{if } \hat{M}_t = m \\ 0, & \text{otherwise.} \end{cases}, \quad \forall m \in \{1, 2, \dots, M\}.$$

The fourth basis function represents the action taken (i.e., the number of CUAVs deployed to each COP) and is written as

$$\phi_{4i}(S_t^x) = x_{ti}, \quad \forall i \in \mathcal{B}.$$

Having stipulated the policy (i.e., decision function) and the value function approximation architecture upon which it is based, we proceed by discussing the manner in which the value function approximation is updated. We employ an API algorithmic strategy similar in structure to those utilized by Rettke et al. (2016), Davis et al. (2017), and Jenkins et al. (2019). Within the policy evaluation step of our API algorithm, we update the value function approximation for a fixed policy using LSTD. Introduced by Bradtke and Barto (1996), LSTD is a computationally efficient method for estimating the adjustable parameters (e.g.,

the θ -vector), when using a linear architecture with fixed basis functions to approximate the value function for a fixed policy. The API algorithm we employ is shown in Algorithm 1.

Algorithm 1 Approximate Policy Iteration Algorithm

```

Initialize  $\theta$ 
for  $n=1$  to  $N$  (Policy Improvement Loop) do
  for  $k=1$  to  $K$  (Policy Evaluation Loop) do
    Simulate a random post-decision state,  $S_{t-1,k}^x$ 
    Record  $\phi(S_{t-1,k}^x)$ 
    Simulate transition to next event, obtain pre-decision state  $S_{t,k}$ 
    Determine decision  $x_t = X^\pi(S_{t,k}|\theta)$  using Eq. (7)
    Record  $C(S_{t,k}, x_t)$ 
    Record  $\phi(S_{t,k}^x)$ 
  end
Apply LSTD
  Compute  $\hat{\theta}$  using either Eq. (8) or Eq. (9)
  Update  $\theta$  using Eq. (10) and Eq. (11)
end

```

The algorithm begins with an initial θ -vector, representing an initial base policy. The performance evaluation of the current policy proceeds as follows. A post-decision state is randomly sampled, and the basis function evaluation vector $\phi(S_{t-1,k}^x)$ is recorded. Next, we simulate one event forward, determine the best decision as per Eq. (7) (i.e., by solving the inner maximization problem), and record both the associated expected contribution $C(S_{t,k}, x_t)$ and the basis function evaluations of the post-decision state, $\phi(S_{t,k}^x)$. A total of K temporal difference sample realizations are collected, where $C(S_{t,k}, X^\pi(S_{t,k}|\theta)) + \gamma\theta^\top\phi(S_{t,k}^x) - \theta^\top\phi(S_{t-1,k}^x)$ is the k th temporal difference, given the parameter vector θ .

After obtaining the K temporal difference sample realizations, we conduct the policy improvement steps of the API algorithm. We compute $\hat{\theta}$, a sample estimate of θ , by regressing the $k = 1, 2, \dots, K$ basis function evaluations of the post-decision states $\phi(S_{t-1,k}^x)$ and $\phi(S_{t,k}^x)$ against the contributions $C(S_{t,k}, x_t)$. We perform a least squares regression so that the sum of the temporal differences over the K inner loop simulations (which approximates the expectation) is equal to zero.

We test and compare two methods for computing $\hat{\theta}$ in Sect. 3. The first method to compute the parameter vector $\hat{\theta}$ uses the normal equation

$$\hat{\theta} = \left[(\Phi_{t-1} - \gamma\Phi_t)^\top (\Phi_{t-1} - \gamma\Phi_t) \right]^{-1} (\Phi_{t-1} - \gamma\Phi_t)^\top C_t, \tag{8}$$

wherein the basis function matrices and reward vectors are defined as

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

and wherein the matrices Φ_{t-1} and Φ_t consist of rows of basis function evaluations of the sampled post-decision states, and C_t is the contribution vector for the sampled states.

Alternatively, Bradtke and Barto (1996) utilize instrumental variables while implementing an approximate policy iteration algorithmic strategy. An instrumental variable is correlated with the regressors, but uncorrelated with the errors in the regressors and the observations.

An instrumental variables method makes it possible to obtain consistent estimators of the regression parameters (i.e., the θ -vector). Bradtke and Barto (1996) suggest Söderström and Stoica (1983) as an appropriate reference for the interested reader. When applying an instrumental variables method, we compute the parameter vector $\hat{\theta}$ as follows.

$$\hat{\theta} = [(\Phi_{t-1})^\top (\Phi_{t-1} - \gamma \Phi_t)]^{-1} (\Phi_{t-1}^\top C_t) \quad (9)$$

The update equation for θ is given by

$$\theta \leftarrow \theta (1 - \alpha_n) + \hat{\theta} (\alpha_n). \quad (10)$$

On the right hand side of Eq. (10), θ is the previous estimate and is based on information from all previous outer loop iterations; $\hat{\theta}$ is our estimate from the current outer loop iteration. As the number of iterations n increases, we place less emphasis on any single estimate and more emphasis on the estimate based on information from the first $n - 1$ iterations.

A generalized harmonic stepsize rule is utilized to smooth in the new observation $\hat{\theta}$ with the previous estimate θ . The stepsize rule is given by

$$\alpha_n = \frac{a}{a + n - 1}. \quad (11)$$

The stepsize rule α_n greatly influences the rate of convergence of the API algorithm and therefore impacts the attendant solutions. Increasing the stepsize parameter a slows the rate at which the smoothing stepsize drops to zero. Selecting an appropriate value for a requires understanding the rate of convergence of the application. Some problems allow convergence to good solutions fairly quickly (i.e., tens to hundreds of iterations) whereas others require much more effort (i.e., thousands of iterations). We observe that relatively lower a -values work best in our application.

Upon obtaining an updated (and possibly smoothed) parameter vector θ , we have completed one policy improvement iteration of the algorithm. The parameters N and K are tunable, where N is the number of policy improvement iterations completed and K is the number of policy evaluation iterations completed.

3 Computational testing, results, and analysis

In this section, we demonstrate the applicability of our MDP model to a problem of interest to the military logistical planning community and examine the efficacy of our proposed solution methodology. For the computational experiments, we utilize a dual Intel Xeon E5-2650v2 3.6 GHz processor having 192 GB of RAM and MATLAB's parallel computing toolbox.

3.1 Representative scenario

We consider the parameterization of the MDP model that represents a MILIRP instance of interest to the military logistical planning community. Time is discretized into 6-h time periods. This discretization allows the day to be divided into four equal time periods. We assume that any single direct delivery CUAV resupply mission to a COP can be completed within a single period. In addition to travel time, a single CUAV resupply mission includes maintenance, fueling, loading, and unloading actions.

We examine an instance of the MILIRP with a BSB supporting an infantry battalion-sized element—approximately one-fourth of an IBCT—having 12 COPs requiring resupply. This

number and size of COPs corresponds to the battalion operating in an area with maximum dispersal of subordinate units, with a platoon consisting of approximately 50 personnel operating at each COP. A platoon will typically consume 8000 pounds of supplies per day during combat operations (General Dynamics Information Technology 2010) and, with four periods in 1 day, 2000 pounds (or one ton) of supplies per period are necessary for sustainment. We conservatively assume that the COP's capacity is three times the daily demand, bringing COP capacity to 12 tons. We assume that the necessary supplies to resupply all COPs are available at the BSB and that the BSB never runs out of supplies; this assumption is reasonable because the BSB is supplied via fixed wing airlift from a higher echelon of the organization, having a more robust storage and warehousing capacity.

CUAV capabilities are increasing as research and development of the systems continue. At present, Lockheed Martin's K-MAX unmanned aircraft system helicopter has successfully transported payloads of three tons at sea level and two tons at 15,000 feet (Lockheed Martin 2010). As recently as 2012, Lockheed Martin announced that the K-MAX routinely transported 4,200 pound loads in combat conditions (Lockheed Martin 2012). As a conservative estimate, we use the two-ton capacity of the K-MAX as the CUAVs' capacity in the computational experiments, and we restrict our supply increments to one-ton increments.

Parameterization of the number of CUAVs available is determined based on the Army's Tactical Unmanned Aircraft System (TUAS) platoon (Department of the Army 2010). The organizational structure of the TUAS platoon in an IBCT continues to evolve and so, in this study, we consider the organizational structure of the TUAS platoon as of 2010, parameterizing the number of TUAS crews at two and the number of CUAVs at four, where the number of crews indicates the number of CUAVs that can be routed simultaneously.

The ψ -values represent the probability a CUAV successfully travels between the BSB and a COP for a specific map. Ideally, an intelligence unit would subdivide (e.g., tessellate) the AO into subregions and assign risk levels to each subregion. This subregion risk level would take into account threats such as the probability of inclement weather, mechanical issues, and hostile enemy actions. The least risky path for each map could then be found and the attendant probability of successfully traveling the path utilized to parameterize ψ . (See McCormack (2014) for an example that tessellates an AO and, using assigned risk levels, identifies a path to maximize ψ for each COP, on each map.) For the computational experiments, we choose to explore the case of $m = 2$ threat maps, one representing a low-threat environment and one representing a high-threat environment. We create reasonable ψ -values with higher values on the low threat map and lower ψ -values on the higher threat map. The ψ -values are generated from a continuous uniform distribution that is bounded between 0.8 and 1 for the high-threat map and between 0.99 and 1 for the low-threat map. This parameterization balances the possibility of failing to make a delivery with providing a realistic risk level at which a commander would deploy a CUAV. Finally, we choose a discount factor of $\gamma = 0.98$ that successfully balances future needs with present needs. The transition probabilities between maps can be inferred based on seasons or fighting intensity; herein, we examine different values of such probabilities in our designed experiment.

3.2 Experimental design

We create a set of experiments to assess the quality of solutions, computational effort, and robustness (Barr et al. 1995) of our proposed ADP approach. To understand the effect of parameterization on the performance of the ADP algorithm, we create a design of experiments. Three response variables are considered: the number of tons delivered via ALOC (i.e.,

via CUAV), the number of required resupply missions via GLOC, and the number of vehicles that remain at the end of the simulation. It is important to note that the ALOC response variable is reported in total tons whereas the GLOC response variable is reported in the total number of missions. For each GLOC mission, up to 12 tons of supplies are delivered. We also record computation times for the ADP to determine the computational effort required to solve the MILIRP. Finally, we assess the robustness of the algorithm by experimenting with both problem factors and algorithmic factors. To report these values, a simulation is performed once the ADP policy has been created. We record the three response variables at three different simulation lengths: 1-month, 2-month, and 3-month horizons, simulating for 100 replications per treatment.

To assess the quality of solutions attained by our solution methodology, we compare the ADP policy to a benchmark policy, which is defined as follows

$$X^{\pi^{benchmark}}(S_t) = \arg \max_{x_t \in \mathcal{X}_{S_t}} C(S_t, x_t).$$

The benchmark policy is myopic in the sense that it does not account for the future state of the system that may result due to its actions. Under the benchmark policy, the BSB resupplies COPs that are relatively easier to reach (i.e., a CUAV has a higher likelihood of successfully transiting to the COP) and have spare capacity to take delivery. However, under the benchmark policy, the BSB also dispatches CUAVs for resupply even under relatively difficult threat conditions (i.e., when a CUAV has a lower likelihood of successfully transiting to the COP).

Four problem characteristics are investigated: the number of COPs (B), the number of CUAV vehicles initially available (v), and two parameters within the threat map transition probability function governing the evolution of the threat condition in the area of operations. The threat map transition probability function contains two problem factors: L and H . We denote the probability of remaining in a low threat map as L , and the probability of remaining in a high threat map as H . The probability of transitioning from a low threat map to a high threat map is represented by $1 - L$ while the probability of transitioning from a high threat map to a low threat map is $1 - H$.

Each of the four problem factors are considered to be continuous variables. We conducted preliminary experiments to determine the high and low factor levels for the number of COPs. The results indicate that the upper limit of where the ADP policy outperforms the benchmark policy in terms of supplies delivered via ALOC is 18 COPs. When we explore beyond this bound to consider 27 COPs, we observed that the benchmark policy delivers three times the supplies via ALOC than the ADP policy. Therefore, we set 9 as the low level and 15 as the high level for the number of COPs. These factor level settings allow the center factor level, 12, to represent the typical number of platoons in a maximally-dispersed battalion that is not augmented with additional units. For the number of CUAVs, we set 4 as the low level and 8 as the high level. The high factor level represents a situation in which two CUAV platoons are present. Since CUAV units are organized in a 2:1 ratio of CUAVs to crews (Department of the Army 2012), we parameterize the number of crews as half the number of CUAVS initially available. The map probability transition function parameters, L and H , are explored at the 0.2 and 0.8 levels. The lower bound, 0.2, represents a low probability of returning to the current threat map whereas the upper bound represents a high probability of returning to the current threat map.

Four algorithmic features are also explored. The number of outer loops (N) and inner loops (K) in the ADP algorithm are investigated. For the inner loop, K -values between 3000 and 7000 are considered. Initial experimentation revealed that the center value of 5000 was

Table 1 Factor settings for factorial design

	Description	Factor	Low (–)	Center (0)	High (+)
Problem factors	Number of COPs	<i>B</i>	9	12	15
	Number of CUAVs	<i>v</i>	4	6	8
	Probability of remaining in a low threat map	<i>L</i>	0.2	0.5	0.8
	Probability of remaining in a high threat map	<i>H</i>	0.2	0.5	0.8
Algorithmic factors	Number of inner loops	<i>K</i>	3000	5000	7000
	Number of outer loops	<i>N</i>	10	20	30
	Instrumental variables	<i>IV</i>	Off (L1)	–	On (L2)
	No smoothing	<i>SM</i>	Off (L1)	–	On (L2)

adequate for some parameterizations of the 12-COP problem instance; investigating smaller and larger numbers of loops provides insight into how the performance of the ADP changes for different computational efforts. For the outer loop, N -values between 10 and 30 are considered. These bounds are chosen because initial experimentation indicated that $N = 30$ provided adequate results for the ADP as compared to the benchmark policy. By investigating N -values lesser than 30, the performance of the ADP can be assessed for lower computation times. The utilization of a least squares approach (i.e., use of Eq. (8)—indicated by (L1) in our experimental results) or an instrumental variables approach (i.e., use of Eq. (9)—indicated by (L2) in our experimental results) is also considered as a two-level categorical variable. We denote this factor as *IV*. Finally, smoothing is also investigated by either applying smoothing (L1) or by not applying smoothing (L2). This final algorithmic feature is also a categorical variable and is denoted as *SM*. The problem and algorithmic factors and their associated levels are shown in Table 1.

A fractional-factorial design with center runs is implemented. We create a 2^{8-2} resolution-V design with a quarter fraction of eight factors in 64 runs. The resolution-V design dictates that some two factor interactions are aliased with three factor interactions, an acceptable structure. More importantly, the two-factor interactions are not aliased with one another, and they are not aliased with the main, first order effects. We use an additional four center points (each with one of four combinations of the two categorical variables), bringing the total number of treatment runs to 68. Using this experimental design, we create ADP policies by calculating the θ -coefficients for the basis functions. Once this is complete, we utilize simulation to obtain the response variable statistics for both the ADP policy and the benchmark policy. We conduct the two simulations per treatment (one after determining an ADP policy, and one utilizing the benchmark policy) over 100 replications. We consistently seed the experiments in both the ADP algorithm and the simulation to achieve variance reduction.

3.3 Experimental results

The fractional-factorial design is used to identify the significant factors in the experiment and provide a basis for analysis. Using this design, we estimate all eight single-factor terms as well as all 36 two-factor interaction terms and selected three-factor interaction terms. The results of the experiment for each response variable at the end of the 3-month simulation are shown in Tables 2 and 3. For each table, the first column indexes the run for subsequent discussion, and

Table 2 Experiment results, 360-period horizon

Run	Coded factor levels	Comp time (s)	ADP policy		Benchmark policy		ALOC diff. (tons)	
			ALOC (tons)	# GLOC Missions	ALOC (tons)	# GLOC Missions		
1	-----+	535.3	73.51 ± 6.56	260.35 ± 0.56	89.79 ± 7.12	259.8 ± 0.64	0 ± 0	- 16.3
2	-----++	1204.0	641.13 ± 22.22	212.66 ± 1.85	89.79 ± 7.12	259.8 ± 0.64	0 ± 0	551.3*
3	-----+++	1549.9	648.11 ± 19.8	212.07 ± 1.65	89.79 ± 7.12	259.8 ± 0.64	0 ± 0	558.3*
4	-----++++	3632.5	81.55 ± 6.48	259.8 ± 0.58	89.79 ± 7.12	259.8 ± 0.64	0 ± 0	- 8.2
5	-----+++++	517.2	91.39 ± 8.21	259.49 ± 0.77	51.31 ± 4.16	263 ± 0.37	0 ± 0	40.1*
6	-----+++++	1201.7	44.43 ± 4.1	263.13 ± 0.38	51.31 ± 4.16	263 ± 0.37	0 ± 0	- 6.9
7	-----+++++	1545.8	47.03 ± 3.91	262.87 ± 0.38	51.31 ± 4.16	263 ± 0.37	0 ± 0	- 4.3
8	-----+++++	3619.0	98.81 ± 7.97	258.48 ± 0.7	51.31 ± 4.16	263 ± 0.37	0 ± 0	47.5*
9	-----+++++	518.8	347.61 ± 28.51	237.31 ± 2.4	248.09 ± 21.65	246.65 ± 1.81	0 ± 0	99.5*
10	-----+++++	1212.8	185.31 ± 15.88	251.3 ± 1.36	248.09 ± 21.65	246.65 ± 1.81	0 ± 0	- 62.8†
11	-----+++++	1665.5	143.91 ± 12.72	254.42 ± 1.06	248.09 ± 21.65	246.65 ± 1.81	0 ± 0	- 104.2†
12	-----+++++	3927.9	512.49 ± 40.98	223.58 ± 3.39	248.09 ± 21.65	246.65 ± 1.81	0 ± 0	264.4*
13	-----+++++	558.1	59.63 ± 6.47	262.49 ± 0.53	83.03 ± 8.33	260.33 ± 0.71	0 ± 0	- 23.4†
14	-----+++++	1289.0	604.69 ± 26.64	215.82 ± 2.22	83.03 ± 8.33	260.33 ± 0.71	0 ± 0	521.7*
15	-----+++++	1556.2	155.95 ± 15.53	253.3 ± 1.33	83.03 ± 8.33	260.33 ± 0.71	0 ± 0	72.9*
16	-----+++++	3637.8	53.13 ± 5.56	262.69 ± 0.52	83.03 ± 8.33	260.33 ± 0.71	0 ± 0	- 29.9†
17	-----+++++	522.2	272.31 ± 17.36	243.49 ± 1.45	152.56 ± 9.72	253.37 ± 0.83	0 ± 0	119.8*

Table 2 continued

Run	Coded factor levels	Comp time (s)	ADP policy		Benchmark policy		ALOC diff. (tons)		
			ALOC (tons)	# GLOC Missions	# Vehicles Remaining	ALOC (tons)		# GLOC Missions	# Vehicles Remaining
18	- + - - - + - -	1228.2	119.77 ± 7.63	256.22 ± 0.65	0 ± 0	152.56 ± 9.72	253.37 ± 0.83	0 ± 0	-32.8 [†]
19	- + - - - + - -	1570.4	105.91 ± 6.75	257.19 ± 0.56	0 ± 0	152.56 ± 9.72	253.37 ± 0.83	0 ± 0	-46.7 [†]
20	- + - - - + + +	3675.2	581.15 ± 39.61	217.58 ± 3.29	0 ± 0	152.56 ± 9.72	253.37 ± 0.83	0 ± 0	428.6*
21	- + - + - - - -	521.5	65.64 ± 4.11	260.53 ± 0.31	0 ± 0	80.75 ± 5.1	259.16 ± 0.4	0 ± 0	-15.1 [†]
22	- + - + - + + +	1224.3	200.07 ± 12.24	249.62 ± 1.06	0 ± 0	80.75 ± 5.1	259.16 ± 0.4	0 ± 0	119.3*
23	- + - + - + + +	1569.5	94.75 ± 5.69	258.39 ± 0.53	0 ± 0	80.75 ± 5.1	259.16 ± 0.4	0 ± 0	14.0*
24	- + - + + + - -	3663.2	62.63 ± 4.09	260.91 ± 0.32	0 ± 0	80.75 ± 5.1	259.16 ± 0.4	0 ± 0	-18.1 [†]
25	- + - - - - + +	521.7	296.16 ± 19.9	242.19 ± 1.67	0 ± 0	373.93 ± 24.44	234.95 ± 2.05	0 ± 0	-77.8 [†]
26	- + - - - + + -	1216.4	1696.81 ± 62.52	125 ± 5.15	1.42 ± 0.19	373.93 ± 24.44	234.95 ± 2.05	0 ± 0	1322.9*
27	- + - - - + + -	1575.4	1849.93 ± 60.93	112.9 ± 5.01	2.05 ± 0.22	373.93 ± 24.44	234.95 ± 2.05	0 ± 0	1476.0*
28	- + - - - + + +	3664.4	332.75 ± 23.58	238.76 ± 1.98	0 ± 0	373.93 ± 24.44	234.95 ± 2.05	0 ± 0	-41.2
29	- + - + - - - -	522.4	1097.73 ± 47.09	174.62 ± 3.9	1.59 ± 0.24	134.25 ± 12.56	254.65 ± 1.02	0 ± 0	963.5*
30	- + - + - + + +	1218.0	116.91 ± 10.29	257.19 ± 0.87	0 ± 0	134.25 ± 12.56	254.65 ± 1.02	0 ± 0	-17.3
31	- + - + + - - +	1565.5	115.62 ± 10.38	257.05 ± 0.87	0 ± 0	134.25 ± 12.56	254.65 ± 1.02	0 ± 0	-18.6
32	- + - + + + + -	3656.2	1340.15 ± 30.36	155.05 ± 2.5	3.74 ± 0.29	134.25 ± 12.56	254.65 ± 1.02	0 ± 0	1205.9*
33	+ - - - - - - -	585.4	81.93 ± 7.08	438.77 ± 0.74	0 ± 0	161.71 ± 14.18	433.43 ± 1.19	0 ± 0	-79.8 [†]
34	+ - - - - + + +	1365.0	120.97 ± 9.62	434.23 ± 0.92	0 ± 0	161.71 ± 14.18	433.43 ± 1.19	0 ± 0	-40.7 [†]

Table 3 Experiment results (cont.), 360-period Horizon

Run	Coded factor Levels	Comp Time (sec)	ADP Policy		Benchmark Policy		ALOC Diff. (tons)
			ALOC (tons)	# GLOC Missions	ALOC (tons)	# GLOC Missions	
35	+++++	1736.6	66.15 ± 5.92	439.19 ± 0.66	161.71 ± 14.18	433.43 ± 1.19	-95.6 [†]
36	++++-	4062.1	100.57 ± 7.89	437.67 ± 0.71	161.71 ± 14.18	433.43 ± 1.19	-61.1 [†]
37	+++--	576.2	41.23 ± 3.65	442.59 ± 0.42	89.05 ± 8.11	439.35 ± 0.72	-47.8 [†]
38	+++--	1360.8	37.15 ± 3.33	443.69 ± 0.33	89.05 ± 8.11	439.35 ± 0.72	-51.9 [†]
39	+++--	1732.2	42.25 ± 3.57	441.9 ± 0.41	89.05 ± 8.11	439.35 ± 0.72	-46.8 [†]
40	+++++	4046.9	52.77 ± 4.86	441.93 ± 0.53	89.05 ± 8.11	439.35 ± 0.72	-36.3 [†]
41	++++-	576.7	201.17 ± 14.3	427.4 ± 1.24	457.79 ± 38.29	408.52 ± 3.2	-256.6 [†]
42	+++--	1367.0	198.49 ± 17.6	427.94 ± 1.55	457.79 ± 38.29	408.52 ± 3.2	-259.3 [†]
43	+++--	1868.9	171.81 ± 14.61	430.71 ± 1.29	457.79 ± 38.29	408.52 ± 3.2	-286.0 [†]
44	+++--	4372.1	785.69 ± 51.19	378.58 ± 4.3	457.79 ± 38.29	408.52 ± 3.2	327.9*
45	+++++	622.1	84.91 ± 8.2	438.12 ± 0.8	156.65 ± 14.64	433.58 ± 1.27	-71.7 [†]
46	++++-	1355.0	298.89 ± 27.76	419.32 ± 2.39	156.65 ± 14.64	433.58 ± 1.27	142.2*
47	++++-	1731.0	138.05 ± 11.56	433.84 ± 1.07	156.65 ± 14.64	433.58 ± 1.27	-18.6
48	+++++	4058.2	74.73 ± 7.52	438.91 ± 0.75	156.65 ± 14.64	433.58 ± 1.27	-81.9 [†]
49	++++-	586.5	188.99 ± 10.36	428.5 ± 0.92	255.48 ± 16.45	423.73 ± 1.34	-66.5 [†]
50	++++-	1368.2	168.9 ± 10.23	430.75 ± 0.92	255.48 ± 16.45	423.73 ± 1.34	-86.6 [†]
51	+++++	1747.8	130.53 ± 8.06	434.21 ± 0.7	255.48 ± 16.45	423.73 ± 1.34	-125.0 [†]

Table 3 continued

Run	Coded factor Levels	Comp Time (sec)	ADP Policy		Benchmark Policy		ALOC Diff. (tons)		
			ALOC (tons)	# Vehicles Remaining	ALOC (tons)	# GLOC Missions			
52	++--++++	4081.0	685.16 ± 46.06	386.81 ± 3.87	0.01 ± 0.02	255.48 ± 16.45	423.73 ± 1.34	0 ± 0	429.7*
53	++++-----	588.0	89.17 ± 5.54	437.01 ± 0.51	0 ± 0	127.78 ± 8.86	434.7 ± 0.79	0 ± 0	-38.6†
54	++++-----	1364.0	192.7 ± 11.64	428.83 ± 1.07	0 ± 0	127.78 ± 8.86	434.7 ± 0.79	0 ± 0	64.9*
55	++++-----	1762.3	89.64 ± 5.56	436.97 ± 0.54	0 ± 0	127.78 ± 8.86	434.7 ± 0.79	0 ± 0	-38.1†
56	++++-----	4089.4	83.86 ± 4.99	438.44 ± 0.47	0 ± 0	127.78 ± 8.86	434.7 ± 0.79	0 ± 0	-43.9†
57	++++-----	592.8	287.29 ± 16.51	421.51 ± 1.42	0 ± 0	676.5 ± 43.38	388.26 ± 3.63	0.03 ± 0.03	-389.2†
58	++++-----	1361.9	556.69 ± 30.65	397.5 ± 2.6	0 ± 0	676.5 ± 43.38	388.26 ± 3.63	0.03 ± 0.03	-119.8†
59	++++-----	1753.3	292.56 ± 18.94	419.43 ± 1.62	0 ± 0	676.5 ± 43.38	388.26 ± 3.63	0.03 ± 0.03	-383.9†
60	++++-----	4096.5	296.72 ± 19.44	420.68 ± 1.66	0 ± 0	676.5 ± 43.38	388.26 ± 3.63	0.03 ± 0.03	-379.8†
61	++++-----	583.6	121.49 ± 11.34	435.28 ± 0.96	0 ± 0	221.48 ± 18.07	426.75 ± 1.56	0 ± 0	-100.0†
62	++++-----	1360.1	126.25 ± 9.78	434.42 ± 0.84	0 ± 0	221.48 ± 18.07	426.75 ± 1.56	0 ± 0	-95.2†
63	++++-----	1771.9	101.52 ± 8.67	436.44 ± 0.72	0 ± 0	221.48 ± 18.07	426.75 ± 1.56	0 ± 0	-120.0†
64	++++-----	4099.5	192.75 ± 15.77	428.13 ± 1.36	0 ± 0	221.48 ± 18.07	426.75 ± 1.56	0 ± 0	-28.7
65	000000--	1816.9	87.81 ± 6.41	348.46 ± 0.58	0 ± 0	145.42 ± 10.03	343.54 ± 0.84	0 ± 0	-57.6†
66	000000--	1819.4	122.93 ± 9.32	345.5 ± 0.8	0 ± 0	145.42 ± 10.03	343.54 ± 0.84	0 ± 0	-22.5
67	000000+-	1824.9	867.18 ± 37.37	282.73 ± 3.11	1.29 ± 0.2	145.42 ± 10.03	343.54 ± 0.84	0 ± 0	721.8*
68	000000++	1816.4	143.31 ± 11.86	343.86 ± 1.02	0 ± 0	145.42 ± 10.03	343.54 ± 0.84	0 ± 0	-2.1

the second column, “Coded Factor Levels”, shows the pattern of factor levels for each factor in the treatment, in the order they are presented in Table 1. The level for each factor, in sequence, is indicated at its low (–), high (+), or center-run (0) values. The third column in Table 2 presents the required computational effort to run the ADP algorithm; overhead operations and simulation times are not included. The next six columns tabulate the mean and standard deviations for the ALOC, GLOC, and CUAV response variables, respectively for the ADP policy and the benchmark policy after 360 time periods (i.e., 3 months). The final column in Tables 2 and 3 presents the difference in the mean ALOC response variable between the ADP and the benchmark policies, with a positive value indicating a better average performance by the ADP policy. Because we are primarily interested in the policies’ respective effects on the ALOC response variable, we also compare the policies’ responses using a one-sided t test to determine whether the difference in the ALOC response variable values are significant at the 0.05 level; runs for which the ADP policy outperforms the benchmark policy by a statistically significant margin are indicated with an asterisk (*) in the final column, whereas runs for which the benchmark policy’s ALOC response variable is significantly better are annotated with a †-symbol.

By examining the results, a pattern is observed. Out of the 68 experimental runs, only 21 result in the ADP policy significantly outperforming the benchmark policy for the ALOC response variable, about 31%. However, if we consider only experimental runs that use smoothing and instrumental variables, this percentage increases to 76%, with 13 of the 17 values showing a significantly better response. Moreover, if we also only consider experiments performed with the low or center number of COPs as a factor setting, the percentage of runs for which the ADP policy is significantly better than the benchmark policy with regard to the ALOC response variable increases to 100% for all nine experiments. This result indicates that several factors significantly impact the ADP policy’s performance as compared to the benchmark policy.

Given the results of the designed experiment, we examine which problem and algorithmic factors from Table 1 are statistically significant in affecting the ALOC response variable: the number of tons that are delivered via CUAV over the 360-period simulation. Before proceeding, we check selected assumptions; we verify the equal variance and normality assumptions using the normal probability plot and a plot of the residuals versus predicted values, neither of which is graphically depicted herein for the sake of brevity. The plots confirm that the normality assumption is upheld, as is the constant variance assumption. Plots of the residuals versus the factor values also confirms that constant variance in the residuals is, for the most part, maintained.

We fit a metamodel to the first-, second-, and selected third-order effect factors and the ALOC response variable, yielding a coefficient of determination (i.e., R^2 -value) indicating that the model explains 95.5% of the variance in the ALOC response variable. An adjusted R^2_{adj} value of 0.931 and the small difference between this value and R^2 indicates that the experimental factors of the metamodel are well chosen. Table 4 provides the significant metamodel terms, their associated coefficient estimates with lower and upper bounds on a 95% confidence interval, and attendant p values, listed from increasing to decreasing levels of significance. Herein, we analyze selected terms from which we derive insights into their significance with respect to the ALOC response variable; we conducted similar analyses for the other response variables, but we omit them from this analysis in order to focus the discussion on the response variable of greatest importance.

The first term, IV , indicates that when the instrumental variables method is not used, the average ALOC response decreases by 163 tons. Thus, utilizing the instrumental variables method to update the θ -vector (instead of the normal equation) positively impacts the quality

of solutions attained by our ADP algorithm. The second term, L , indicates that a 0.3 increase in the value of L increases the ALOC response by 116.2 tons. This means that, when the probability of staying in a low threat map increases, the ALOC response variable also increases. This result is intuitive, as the simulation will remain in a low threat map for longer amounts of time allowing more CUAVs to make successful deliveries. The third term, $IV \cdot SM$, indicates that the interaction of smoothing and instrumental variables is also important. This trend can also be observed by examining the four center runs in Table 2. All other factors held constant at their mid-points, out of the four combinations of the IV and SM levels, the combination that results in the highest ALOC response is the combination of instrumental variables and smoothing.

The fourth term, $IV \cdot B$, captures the interaction between using instrumental variables and the number of COPs. Looking to the seventh term, B , we note that increasing the number of COPs decreases the response variable. This parallels the results from initial testing conducted on the number of COPs, which indicated that the ADP does not perform as well for problem instances with a higher number of COPs. However, inspecting the fourth term $IV \cdot B$ again, we observe that, when the instrumental variables method is not utilized, an additional 3 COPs actually increases the ALOC response variable by 99.7 tons. Taken together, the fourth and seventh terms indicate that the instrumental variables method is less effective as the number of COPs increases. This conclusion is an example of the difficulty in interpreting metamodels with significant interaction terms. The fifth term, H , indicates that increasing the probability of remaining in a high threat map by 0.3 decreases the ALOC response variable by 97 tons. This makes sense, as remaining in the high-threat map is more risky and results in fewer CUAVs being deployed and fewer successful deliveries. The sixth term, smoothing, indicates that when smoothing is used, the average ALOC response increases by 92.3 tons. Thus, utilizing smoothing when updating the θ -vector positively impacts the quality of solutions attained by our ADP algorithm. The estimate for the eighth term, the main factor v , indicates that increasing the number of vehicles by two increases the value of the ALOC response by 88.6 tons. This also makes sense, as more initial CUAVs allows for more potential deliveries.

Ten of the next 11 significant terms from Table 4 are two-factor interactions. Of note is the fact that, when we consider interactions with B (i.e., terms 11, 14, and 16), all the estimates for the terms are negative. This indicates that even a large number of CUAVs, a high probability of staying in the low threat map, or smoothing cannot overcome the negative effect of resupplying a large number of COPs. The 18th term, K , is the lowest significant main effect. Increasing the number of inner loops results in an increasing ALOC value. This makes sense as the higher number of inner loops should allow for the determination of improved solutions.

Terms 20 and 22 introduce the two significant three-factor interactions. It should be noted that $L \cdot v \cdot K$ is aliased with another three-factor interaction, $IV \cdot H \cdot N$. We choose $L \cdot v \cdot K$ as the significant factor because N is not found to be a significant factor in the metamodel. Since additional experimentation is not performed within the scope of this paper, it is not possible to verify this choice. The occurrence of significant three-factor interactions suggests that interactions between the variables beyond the two-factor interactions are important. Specifically, $IV \cdot H \cdot SM$ (i.e., Term 20) indicates that the combination of smoothing, instrumental variables, and probability of remaining in a low threat map are important in combination. The only occurrence of N in the metamodel is found in the 23rd term, which captures the two factor interaction of $H \cdot N$. With a p value of 0.019, this term is significant, but it is the least significant of those terms remaining in the metamodel.

Finally, we further investigate θ , the vector of weights for the basis functions, for two particular treatments from the experiment to develop further insight into the ADP. We first

Table 4 Coefficient estimates for ALOC response

#	Term	Estimate	Lower 95%	Upper 95%	Prob> t
1	IV[L1]	− 162.9	− 186.6	− 139.3	< 0.0001
2	<i>L</i>	116.2	91.8	140.6	< 0.0001
3	IV[L1]·SM[L1]	− 100.4	− 124.1	− 76.8	< 0.0001
4	IV[L1]·B	99.7	75.3	124.1	< 0.0001
5	<i>H</i>	− 97.0	− 121.4	− 72.6	< 0.0001
6	SM[L1]	92.3	68.7	116.0	< 0.0001
7	B	− 94.3	− 118.7	− 69.9	< 0.0001
8	<i>v</i>	88.6	64.2	112.9	< 0.0001
9	IV[L1]· <i>L</i>	− 75.3	− 99.7	− 50.9	< 0.0001
10	<i>L</i> · <i>v</i>	61.6	37.2	86.0	< 0.0001
11	<i>L</i> ·B	− 61.3	− 85.7	− 36.9	< 0.0001
12	SM[L1]· <i>v</i>	58.3	33.9	82.7	< 0.0001
13	IV[L1]· <i>v</i>	− 56.7	− 81.1	− 32.3	< 0.0001
14	B· <i>v</i>	− 53.9	− 78.3	− 29.6	< 0.0001
15	<i>L</i> ·SM[L1]	52.9	28.5	77.3	< 0.0001
16	SM[L1]·B	− 49.6	− 74.0	− 25.2	0.0002
17	IV[L1]· <i>H</i>	47.9	23.5	72.3	0.0003
18	K	47.7	23.3	72.1	0.0003
19	IV[L1]·K	− 41.8	− 66.2	− 17.4	0.0012
20	IV[L1]· <i>H</i> ·SM[L1]	32.6	8.2	57.0	0.0099
21	<i>H</i> ·SM[L1]	− 31.9	− 56.3	− 7.5	0.0116
22	<i>L</i> · <i>v</i> ·K	− 29.5	− 53.8	− 5.1	0.0191
23	<i>H</i> ·N	− 29.3	− 53.7	− 4.9	0.0196

consider the θ coefficients resulting from experimental Run 27, which produces the largest ALOC response. By analyzing the θ -values for this particular run, insights into *why* the ADP approach performed well are gained. First, by simply graphing the θ -values, it is evident that there is a cutoff between values near zero and those that are not. Values that are near zero indicate that the basis function corresponding to that value did not yield a change in the total discounted reward. For example, the θ -value that corresponds with the current number of vehicles remaining has a value of 48.65 for this particular experimental treatment. This means that, for each additional CUAV, there is an average increase in the total discounted reward of 48.65 tons with all the other variables held constant. Only 20 of the 166 basis functions have corresponding θ -values above 1 or below -1 , which we graph in Fig. 2. These 20 θ -values fall into four categories of term types: the intercept, number of vehicles, actions, or map-action interactions. The basis function that captures the current number of vehicles has a value of 48 and was discussed above. The basis function coefficients corresponding to the action taken at each COP have values between 43.49 and 49.71. This indicates that deploying an additional CUAV to a particular COP increases the total discounted reward by 43 to 50 tons. Finally, the θ -values for the interactions between the current map and the action taken at each COP varies between -3.33 and -17.9 . Due to the fact that the current map is modeled as a binary variable (for which the low risk map is ‘0’, and the high risk map is ‘1’), deploying CUAVs when in the high risk map decreases the total expected reward between 3.33 and 17.9 tons,

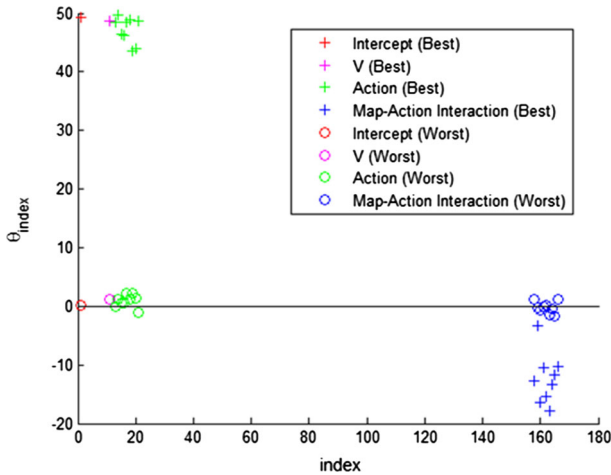


Fig. 2 θ -values for best and worst results

depending on the COP. This shows that the basis function is capturing the long-term effect of dispatching CUAVs in the high risk threat map and potentially losing the assets.

We next consider a second set of θ -values determined by choosing the experimental treatment that performed the worst, Run 6. Unlike the treatment considered for the best θ -values, this run did not include instrumental variables or smoothing, algorithmic factors that have been found to be important to the performance of the ADP. Examining the θ -values obtained for this experimental run, it is readily identifiable that the magnitude of the values is much smaller than the previous θ -values, varying between -1.77 and 2.26 . The poor performance of this particular treatment makes sense, as the basis function is not producing θ -values that address how the reward function changes. As the θ -values for the ADP approach zero, the ADP policy approaches the benchmark policy. For example, in the best case run we observed a θ -value of 48.65 for the current number of vehicles. For the worst case run, we obtain a value of 1.22 for the same parameter. Despite the small magnitudes of the θ -values for the worst run, the largest θ -values in magnitude include the same three groupings of basis functions: the number of vehicles remaining, the current action taken at each COP, and the map-action interactions. This indicates that, despite the fact that the magnitudes of the θ -values are low, the significant contributors to the total discounted reward remain the same.

It is challenging to succinctly portray ADP policy results for 68 problem instances that involve up to a 17-dimension state space and a 15-dimension action space. Nonetheless, inspection of the ADP policies reveals a general trend. Map-action combinations with larger associated ψ -values (indicating the resupply of COPs during low threat conditions) are preferred; indeed, to conserve CUAV resources, a subset of high risk COPs are not serviced at all due to their relatively smaller ψ -values. Alternative resupply delivery methods may be warranted for such COPs.

4 Conclusions

This paper examines the MILIRP. The intent of the research is to determine policies that improve the performance of a deployed military resupply system. Development of an MDP

model of the MILIRP enables examination of many different military resupply planning scenarios. However, solving the MDP for realistic-sized instances requires the design, development, and implementation of an ADP algorithm. We develop an API algorithm that utilizes LSTD learning for policy evaluation, defining a set of basis functions within a linear approximation architecture to approximate the value function around the post-decision state. 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 resupply planning scenario based on contingency operations in Afghanistan.

This research provides the military with insight into the emerging field of unmanned tactical airlift and, more specifically, CUAVs. With high casualty rates from ground resupply efforts, the Army looks to unmanned aerial resupply vehicles as a resource that could be used to supplement ground resupply efforts. Every ground convoy *not conducted* while supporting subordinate units with other means provides an opportunity to save lives. The use of CUAVs provides other benefits: the higher flight ceiling and better flight performance in adverse weather conditions make unmanned helicopters less susceptible to man-portable air defense systems, provide for greater maneuverability, and allow sorties to be scheduled in riskier environments than their manned counterparts. Additionally, a more dedicated platform may enable a more reliable, quicker, and more flexible resupply effort. The addition of a dedicated CUAV unit would also free manned rotary assets for combat missions. However, the CUAVs' key ability remains the potential to save lives by reducing the need for ground convoy missions to resupply subordinate units in a combat environment.

We look to the K-MAX as a specific testament to the Army, Navy, and Air Force's interest in unmanned tactical airlift platforms. As part of a \$45.8 million dollar contract, Lockheed Martin and Kaman Aerospace Corporation successfully deployed three optionally manned K-MAX helicopters to Afghanistan in 2011 (Lamothe 2014; Lockheed Martin 2018). During their 2-year deployment, the K-MAX helicopters were used by Marines in a tactical airlift role to decrease the number of ground convoys necessary, especially in hazardous areas. Over the duration of the deployment, the K-MAX was used by the Marines to deliver 4.5 million pounds of supplies, moving 15 tons per day over 1700 resupply missions in just 2 years of operation (Lockheed Martin 2018). The Washington Post (Lamothe 2014) reported that "the Marines raved about [the K-MAX's] utility and dependability" despite one of the three helicopters crashing (with no injuries).

The K-MAX's performance laid the groundwork for unmanned tactical airlift to become a reality in today's warfare. With this operational implementation, a capability gap exists regarding how to best apply these CUAV assets in a combat environment. This research sought to fill this gap by informing the development of tactics, techniques, and procedures for optimal utilization of CUAV resources for commanders in the field. Proper utilization of CUAVs will prolong the lifespan of the CUAV and increase its utility. By providing procedures for sustaining units via CUAV, we provide decision makers with a potentially lifesaving tool. Although no combat environment will perfectly match the computational example provided in this paper, decision makers can create their own threat maps and inputs to gain an understanding of a near-optimal policy for deployment of their CUAV resources. Even if the policy is not followed exactly, it will provide a framework for understanding how the CUAVs should be deployed and their expected lifespan, allowing these commanders to better utilize their tactical airlift capabilities.

The ADP algorithm developed to solve the MILIRP with direct delivery provides a policy for the allocation of CUAV assets to resupply a battalion-sized Army unit. The ADP policy was shown to be successful in outperforming the benchmark policy for categorical levels of selected problem features. Experimentation on algorithmic features allowed for the

conclusion that the ADP policy improves when high numbers of inner loops are utilized with instrumental variables and smoothing. In terms of problem features, the ADP policy's performance decreases when a large number of COPs is involved, but the ADP approach is robust to changes in other problem features. Specific combinations of inputs resulted in up to 71% of supplies being delivered via air line of communication over a 1-month horizon, 65% over a 2-month horizon, and 57% over a 3-month horizon.

There are several areas for future research on the MILIRP. In terms of formulating the problem, the addition of supply classes would increase the granularity of the problem and more accurately represent the Army's real-world resupply procedures. Additional insight can be gained by modeling demand differently. In lieu of a deterministic consumption rate of supply at each combat outpost, further study is warranted with demands of a stochastic nature. Finally, we explore only a single ADP algorithm for determining a resupply policy; exploration of alternative ADP algorithms may yield results that scale better than the results gained from the API LSTD algorithm implemented herein.

Acknowledgements 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, United States Army, Department of Defense, or United States Government.

References

- Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G., Stewart, J., & William, R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1), 9–32.
- 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. (2012). *Dynamic programming and optimal control* (4th ed., Vol. 2). Belmont: Athena Scientific.
- Bertsekas, D. P. (2017). *Dynamic programming and optimal control* (4th ed., Vol. 1). Belmont, MA: Athena Scientific.
- Bradtke, S. J., & Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1–3), 33–57.
- Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Thirty years of inventory routing. *Transportation Science*, 48(1), 1–19.
- Davis, M. T., Robbins, M. J., & Lunday, B. J. (2017). Approximate dynamic programming for missile defense interceptor fire control. *European Journal of Operational Research*, 259(3), 873–886.
- Department of Defense. (2009). *FY 2009–2034 Unmanned Systems Integrated Roadmap*.
- Department of the Army. (2010). *Army Field Manual: Brigade Combat Team No. 3-90.6*.
- Department of the Army. (2012). *Cargo Unmanned Aircraft System (UAS) Concept of Operations*.
- General Dynamics Information Technology. (2010). *Future modular force resupply mission for unmanned aircraft systems (UAS)*. Falls Church: General Dynamics Information Technology.
- Jenkins, P. R., Robbins, M. J., & Lunday, B. J. (2019). Approximate dynamic programming for military medical evacuation dispatching policies. *INFORMS Journal on Computing* 1–40 (in press).
- Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. P. (2002). The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1), 94.
- Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. P. (2004). Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1), 42–70.
- Lagoudakis, M. G., & Parr, R. (2003). Least-squares policy iteration. *The Journal of Machine Learning Research*, 4, 1107–1149.
- Lamothe, D. (2014). Robotic helicopter completes Afghanistan mission, back in U.S. <http://www.washingtonpost.com/news/checkpoint/wp/2014/07/25/robotic-helicopter-completes-afghanistan-mission-back-in-u-s/>. Accessed 18 Feb 2015.
- Lockheed Martin (2010). K-MAX unmanned aircraft system. <http://www.lockheedmartin.com/content/dam/lockheed/data/ms2/documents/K-MAX-brochure.pdf>. Accessed 18 Oct 2014.
- Lockheed Martin. (2012). Unmanned K-MAX operations in Afghanistan. <https://www.youtube.com/watch?v=s-mr5I657GU>. Accessed 19 Feb 2015.

- Lockheed Martin. (2018). K-MAX deployment infographic. https://www.lockheedmartin.com/us/products/kmax/infographic.html?_ga=2.196024741.809269596.1517926078-149994848.1490021743. Accessed 06 Feb 2018.
- McCormack, I. (2014). The military inventory routing problem with direct delivery. Master's thesis, Air Force Institute of Technology.
- Mu, S., Fu, Z., Lysgaard, J., & Eglese, R. (2010). Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society*, 62, 742–749.
- Powell, W. B. (2011). *Approximate dynamic programming: solving the curses of dimensionality* (2nd ed.). Hoboken, NJ: Wiley.
- Powell, W. B. (2012). Perspectives of approximate dynamic programming. *Annals of Operations Research*, 13(2), 1–38.
- Rettke, A. J., Robbins, M. J., & Lunday, B. J. (2016). Approximate dynamic programming for the dispatch of military medical evacuation assets. *European Journal of Operational Research*, 254(3), 824–839.
- Ruszczynski, A. (2010). Commentary-post-decision states and separable approximations are powerful tools of approximate dynamic programming. *INFORMS Journal on Computing*, 22(1), 20–22.
- Söderström, T. D., & Stoica, P. G. (1983). *Instrumental variable methods for system identification* (Vol. 57). Berlin: Springer.
- 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 IEEE conference on decision and control* (Vo. 4, pp. 4052–4057). IEEE.
- Williams, J. (2010). Unmanned tactical airlift: A business case study. Master's thesis, Air Force Institute of Technology.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.