

## Research Article

# Architectural Models Enabled Dynamic Optimization for System-of-Systems Evolution

Zhemei Fang ,<sup>1</sup> Xiaozhou Zhou ,<sup>1</sup> and Ani Song ,<sup>2</sup>

<sup>1</sup>School of Artificial Intelligence and Automation,  
National Key Laboratory of Science and Technology on Multispectral Information Processing,  
Huazhong University of Science and Technology, Wuhan 430074, China

<sup>2</sup>School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China

Correspondence should be addressed to Zhemei Fang; [zmfang2018@hust.edu.cn](mailto:zmfang2018@hust.edu.cn)

Received 15 October 2019; Accepted 30 January 2020; Published 21 February 2020

Academic Editor: Qingling Wang

Copyright © 2020 Zhemei Fang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

System of Systems (SoS) is designed to deliver value to participant stakeholders in a dynamic and uncertain environment where new systems are added and current systems are removed continuously and on their own volition. This requires effective evolution management at the SoS architectural level with adequate support of process, methods, and tools. This paper follows the principle of Model-Based Systems Engineering (MBSE) and develops a holistic framework integrating MBSE conceptual representations and approximate dynamic programming (ADP) to support the SoS evolution. The conceptual models provide a common architectural representation to improve communication between various decision makers while the dynamic optimization method suggests evolution planning decisions from the analytical perspective. The Department of Defense Architecture Framework (DoDAF) models using Systems Modeling Language (SysML) are used as MBSE artifacts to connect with ADP modeling elements through DoDAF metamodels to increase information traceability and reduce unnecessary information loss. Using a surface warfare SoS as an example, this paper demonstrates and explains the procedures of developing DoDAF models, mapping DoDAF models to ADP elements, formulating ADP formulation, and generating evolutionary decisions. The effectiveness of using ADP in supporting evolution to achieve a near-optimal solution that can maximize the SoS capability over time is illustrated by comparing ADP solution to other alternative solutions. The entire framework also sheds light on bridging the DoDAF-based conceptual models and other mathematical optimization methods.

## 1. Introduction

System-of-Systems (SoS) problems have received increased attention in the aerospace industry in recent years with the advancement of communication and information technologies. In these problems, multiple heterogeneous, geographically distributed systems interact and collaborate for an overarching purpose while maintaining managerial and operational independence [1, 2]. Integrated defense system is a typical combat SoS consisting of various kinds of platforms, aircraft, sensors, weapons, and so on. Compared to traditional monolithic systems, an SoS appears more dynamic and evolutionary because it is almost impossible to have a complete and fully formed

SoS. Over time, new systems can be added, current systems can be replaced or removed, and the interdependency network can switch to a more efficient structure. The change of the command and control structure in the defense community from hierarchy to multidomain illustrates a typical evolution process that is challenging to manage. Evolution is often considered as a major challenge in the SoS area [3] due to its complexity and difficulty resulting from the SoS characteristics including system autonomy, system diversity, complicated interactions, uncertainties, and dynamicity.

The U.S. Department of Defense (DoD) has made a significant effort to address the complexity of SoS evolution management from the Systems Engineering (SE) perspective

for enhancing defense acquisition success. SE Guide for SoS [4] and Defense Acquisition Guidebook (DAG) [5] provided useful guidance on streamlining the evolutionary acquisition process. Dahmann [6] further unwinded the “trapeze model” in the SE guide for SoS to an intuitive time-sequenced “wave model” that is well recognized by the acquisition and SoS community. The model consists of four key decision points—“conduct SoS analysis,” “develop/evolve SoS architecture,” “plan SoS update,” and “implement SoS update,” as shown in Figure 1. The “wave model” itself does not specify any particular methods or tools to support the decision-making process for SoS development; hence, this paper employs it as an overarching framework for developing effective methods to support the SoS evolution.

Following the procedures of the “wave model,” this paper uses a Model-Based Systems Engineering (MBSE) approach to identify the operational tasks, system alternatives, interfaces, connectivity, and exchanges within and between the constituent systems for an SoS problem. MBSE is a new paradigm in SE that formalizes the practice of system development through the use of system models that are usually depicted with Systems Modeling Language (SysML) [7]. To further manage the complexity of SoS problems, this paper brings in the U.S. DoD Architecture Framework (DoDAF), supported by SysML, to describe the models of the target problem from different stakeholders’ viewpoints. The conceptual architecture description is important because it allows the SoS practitioners to organize all the information in a structured way and allows all the associated stakeholders to have a common understanding of the problem.

While the conceptual architectural models demonstrate how different elements fit together into a consistent whole, analytical methods are also in need to provide quantitative solutions for decision makers. At the architectural level, the evolutionary process of system addition, removal, and modification within an SoS can be addressed as a special multistage portfolio planning problem where decisions occur at the “develop/evolve SoS architecture” block in the “wave model.” Decisions, subject to various constraints, at the current stage could have an impact on the parameters, decisions, and capabilities in the future. For instance, new technology investment decisions at present that decrease the budget for current acquisition might increase technology readiness level and in turn improve future system capability or lower future acquisition cost. This paper adopts approximate dynamic programming (ADP) [8] to solve this multistage decision-making problem based on three major considerations: (1) the decision-tree-like representation of ADP shares similarity with the SoS evolution process; (2) ADP leverages various approximation strategies and other techniques to address the explosion issue of state space, decision space, and sample space that commonly occurs in a complex SoS; (3) ADP is able to capture the interaction between decisions and exogenous parameters.

In the current literature, the conceptual models and analytical methods were often studied separately, and thus this paper proposes to integrate the architectural models with the dynamic optimization method for better supporting

SoS evolution. On the one hand, the DoDAF-based conceptual models help to identify the key elements determining the consistent performance delivery during SoS evolution; on the other hand, the ADP method provides sequential decision support from the analytical perspective based on the key information extracted from the DoDAF models. Initial attempts to connect the conceptual models and optimization methods have been published in some recent papers [9–11], but significant efforts, such as a clearer mapping between DoDAF models and mathematical elements, are still required. Overall, this paper aims to contribute to the SoS research by developing a comprehensive framework that links the architectural models and dynamic optimization method based on the steps in the “wave model” to generate effective decision support for SoS evolution. More specifically, this paper (1) develops the ADP formulation and solution approach for the evolutionary multistage decision-making process of an SoS problem, (2) links the DoDAF models and ADP formulation through the DoDAF metamodel (DM2) that can enhance the traceability between different phases of analysis, and (3) sheds light on how to build the connections between DoDAF conceptual models and other mathematical optimization methods.

The rest of this paper consists of four sections. Section 2 introduces the background and state of the art of relevant research. Section 3 employs a surface warfare SoS as a motivating example to explain the process of building DoDAF models, mapping DoDAF models to ADP elements, building ADP formulation for SoS evolution, and solving the ADP problem. Section 4 generates and explains the results of the example in Section 3. Section 5 concludes the paper and indicates future directions.

## 2. Background and Related Work

**2.1. System-of-Systems Evolution.** Evolutionary development is one of the five distinguishing SoS features that are commonly accepted by the SoS community. Evolution generally reflects a process of progressive change in the attributes of the evolving entity or that of one or more of its constituent elements [12]. Factors that can influence the SoS evolution are quite diverse, and thus organizations and researchers have studied the subject from a wide range of perspectives and some of them are not even under the name of SoS evolution. In the defense domain, official documents such as SE Guide for SoS [4] and DAG [5] as well as DoDAF and MBSE relevant methodologies aim to formalize the development and evolution process to improve efficiency and reduce risks. The U.S. DoD emphasizes the criticality of modularity and openness [13] in military system design and acquisition to better deal with unexpected changes. Maier [1] also articulated the importance of “stable intermediate form” and “leverage at the interfaces” for successfully architecting an SoS. Lock [14] proposed a qualitative analysis method to identify, organize, and discuss the information from the aspects of target, context, keyword, risk, consequences, and actions to manage SoS evolution risks. Carney et al. [15] discussed the motivation, locality, and outcome of SoS evolution by borrowing ideas from software evolution.

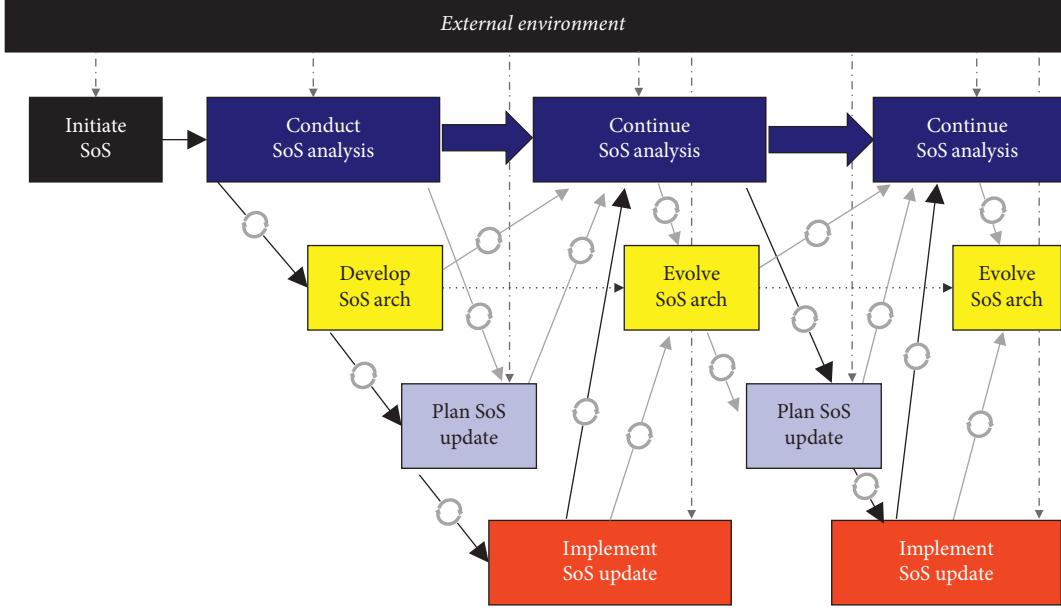


FIGURE 1: Wave model [6].

Lehman [16, 17], as a pioneer in software evolution, developed eight important laws of software evolution from the 1970s to 1990s, including continuing change, increasing complexity, self-regulation, conservation of organizational stability, conservation of familiarity, continuing growth, declining quality, and feedback system. These laws are also applicable to SoS evolution, but with necessary shift from code to SoS elements.

An important line of research that is quite related to the quantitative and analytical support for SoS evolution is dynamic strategic planning (DSP) [18]. A variety of methods are available for addressing the DSP problem or more specifically multistage decision-making problem under uncertainty. Real options analysis (ROA) [19, 20] has been employed in evaluating the value of the flexibility-enabling staged deployment of SoSs such as low Earth orbit communications satellite constellation, maritime domain protection system, and so on. Epoch-Era Analysis (EEA) [21, 22] is another method for analyzing uncertain future contexts on the value of a system in a special structure where an epoch is a time period of fixed contexts and an era is a sequence of epochs that create a potential lifecycle. The time-expanded decision network (TDN) [23] method aims to select the development path with minimum cost under different operating scenarios. Davison et al. [24] also proposed a graph-theory-based method for charting development pathways with a trade space of potential architectures, with a view to enabling robustness to technology portfolio realization and later architectural changes. Tan et al. [25] proposed a multi-objective evolutionary algorithm (EA) to obtain system development plans informing which system components should be advanced to which maturity levels within resource limit. Acheson et al. [26] used genetic algorithm (GA) to populate initial SoS meta-architecture and formulate the trade space of possible architectures in a “wave

model” driven agent-based simulation of SoS evolution. Some of the methods such as ROA, EEA, and TDNs are intuitive to understand, but suffer from the computational complexity when the problem size increases; while others such as EA and GA are not quite intuitive for decision makers to understand in the SoS context and cannot capture the impact of decisions on parameters in the future. Plus, most of them focus on the value calculation of flexibility rather than the evolutionary decision-making support of system addition, removal, and modification.

Approximate dynamic programming that combines dynamic programming with stochastic approximations illustrates great potential in representing the evolutionary process intuitively, along with the capability to alleviate the computational complexity resulted from the large number of systems, interactions, and uncertainties in an SoS. Maier [27] has proposed dynamic programming as one of the most promising methods to formulate the SoS management problem. ADP is favorable in addressing multistage decision-making problems under uncertainty by leveraging proper value function approximations. ADP applications are available in various examples, such as military airlift operations [28], energy resource allocation problem [29], and carbon reduction strategy analysis [30], to name a few. In the SoS context, Nusawardhana [31] employed ADP to a class of SoS problems that involves simultaneous design of a new system (e.g., aircraft) and allocation of the new system along with existing systems (e.g., airline fleet allocation) over a finite horizon. Fang and her co-authors [32–34] have also demonstrated the ability of ADP in supporting SoS evolution decision making for defense acquisition with proper assumptions. This paper continues leveraging the advantages of ADP in addressing complex multistage decision-making problems under uncertainty in the SoS context with more elements incorporated in the formulation.

*2.2. MBSE and Architectural Models Enabled Analysis.* The International Council on Systems Engineering (INCOSE) defined MBSE in their SE Vision 2020 as “the formalized application of modeling to support system requirements, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life-cycle phases” [35]. As stated previously, MBSE relevant methodologies can support the SoS evolution from the perspective of standardization and formalization. The goal of MBSE is to replace the traditional document-centric approach by model-centric approach to achieve improved design quality, enhanced communications, reduced development risks, and increased productivity [36]. SysML that emerges from Unified Modeling Language (UML) has been adopted widely in supporting MBSE. Although MBSE and SysML were not initially proposed for SoS development, they have received great attention from the SoS community. For example, Mori et al. [37] proposed a SysML profile for SoS and demonstrated how to use the profile in a model-driven engineering process to support different types of analysis. Lane and Bohn [38] demonstrated that the SysML models can capture information distilled from a multitude of constituent system documents and engineering experts and then integrate them to better facilitate SoS engineering. While SysML provides graphical representations for modeling, DoDAF provides different viewpoints to view and analyze an SoS, primarily including capability viewpoint (CV), operational viewpoint (OV), system viewpoint (SV), services viewpoint (SvCv), and data and information viewpoint (DIV). A successful example of integrating DoDAF and SysML is the Unified Profile for DoDAF /MODAF (UPDM) that was started by members of INCOSE and OMG [39]. Hayden and Jeffries [40] employed SysML, DoDAF 2.0, and UPDM to model the architecture for the joint polar satellite system ground system. Mitchell [41] used UPDM with SysML and DoDAF for modeling a common submarine combat system. This paper also uses SysML to develop DoDAF models for understanding and analyzing an SoS problem.

Conceptual DoDAF models and quantitative methods need to be closely connected to reduce unnecessary information loss. In the past, the integration of conceptual models and simulation models has been paid closer attention. For example, researchers have developed architectural models enabled executable simulation models based on colored Petri net [42, 43], agent-based modeling [44], and network-based simulation [45, 46]. However, not much work has focused on building the connection between DoDAF models and optimization-based models or analytical models [10]. A few researchers have started exploring the integration of analytical methods and model-based architectures. For instance, LaSorda et al. [9] combined model-based systems engineering with programmatic optimization for satellite SoS architecting. Guariniello et al. [10] took the initial step to connect a few analytical methods for SoS development and DoDAF models. Huff et al. [11] explored the links between DoDAF-based models and an integer linear program for rigorously evaluating system vulnerability. This paper aims

to build a bridge between the DoDAF models and ADP-focused mathematical formulation to reduce the information loss between different phases in the “wave model” and provide more informed decision support for SoS evolution.

### 3. Proposed Framework: Architectural Models Enabled Dynamic Optimization

The proposed methodology package includes DoDAF modeling, mathematical formulation, and solution generation. Figure 2 provides an overview of the steps involved in the method and their connections to the steps in the “wave model.” In the phase of “conduct SoS analysis,” DoDAF models are developed for clarifying the mission objectives, requirements, operational tasks, existing and potential systems, exchanged resources, and future possibilities. The primary elements required for ADP formulation are also identified in this phase. The SoS evolutionary solutions are then generated in the phase of “develop/evolve SoS architecture.” Although we can generate the entire evolutionary path, only the architectural decisions at the current stage will be implemented. When the “wave model” moves to the next “Vee,” both internal and external changes could occur, and another round of decision making begins.

To clearly illustrate the implementation procedures, this paper employs a Surface Warfare (SUW) system as an SoS example to describe the method step by step. SUW is a littoral combat ship (LCS) centered mission, tasked with conducting detect-to-engage operations against surface contacts. LCS uses open architecture design that allows to be equipped with modular “plug-and-fight” mission packages. Based on references [47, 48], the SUW mission package primarily contains helicopters (e.g., MH-60R), unmanned aerial vehicles (UAV, e.g., MQ-8), unmanned surface vehicles (USV), and other supporting systems. The managerial independence and operational independence of the geographically distributed participant systems characterize the SUW system as an SoS. To satisfy the user needs continuously as time proceeds and prevent chaos when various changes occur, a visible and formalized model-based approach is important to drive the SUW architecture towards desired outcomes. For simplicity, we limit the scope of SoS evolution as in the area inside the rectangle with dashed line in Figure 3. That is, this paper only considers evolution decisions in the form of system addition in a stable operational context.

*3.1. Step I: Developing Supportive DoDAF Models.* “Fit-for-Purpose” is the key principle of DoDAF 2.02. Following this principle, the paper only builds necessary DoDAF models to understand the SoS problem. Table 1 lists the DoDAF views used in this paper and their purposes, including OV-5a (operational activity decomposition tree), OV-5b (operational activity model), OV-6a (operational rules model), SV-4 (systems functionality description), SV-5b (operational activity to systems traceability matrix), SV-7 (systems measures matrix), SV-8 (systems evolution

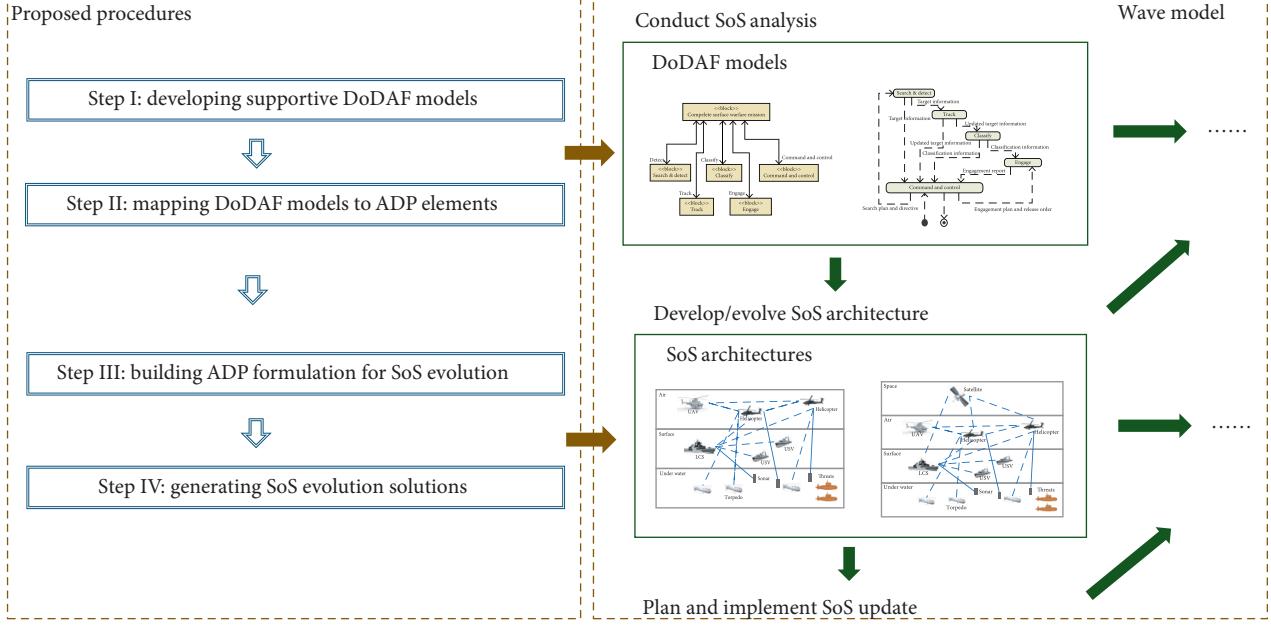


FIGURE 2: Procedures of the proposed method and relevance to the “wave model.”

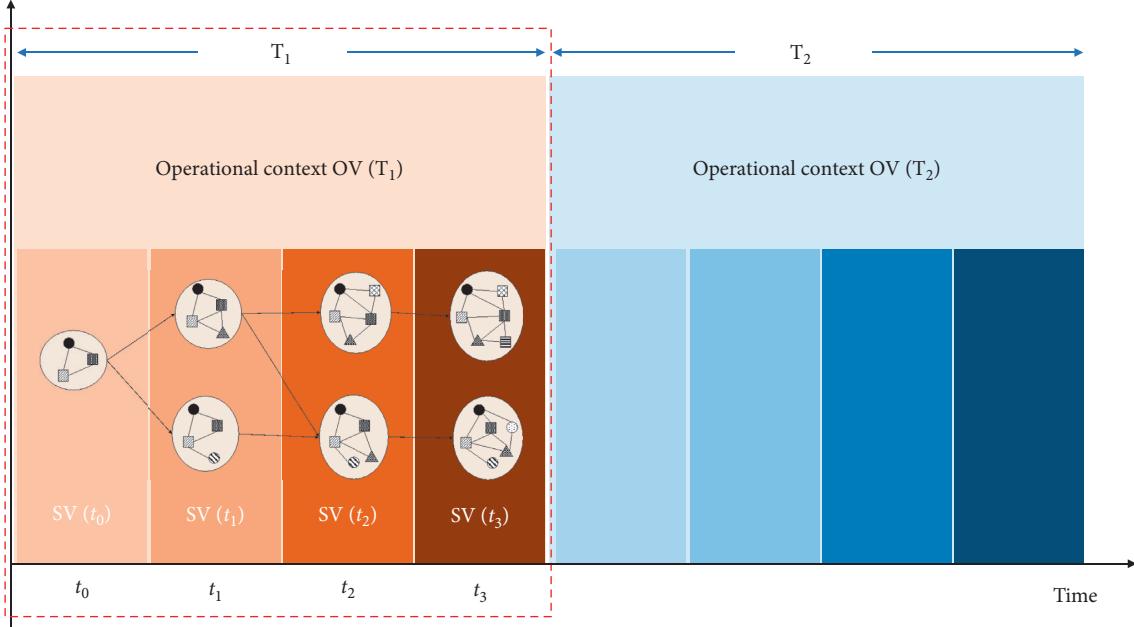


FIGURE 3: Scope of SoS evolution in this paper (OV: operational viewpoint; SV: systems viewpoint).

description), and SV-10a (systems rules model). The DoDAF models are developed with SysML using MagicDraw software and a couple of supplementary matrices and graphs. In SysML, block definition diagram (BDD) represents structural elements called blocks and their interconnections; activity diagram represents behaviors in terms of the order in which actions execute based on the availability of their inputs, outputs, and control and how the actions transform the inputs to outputs. As shown in Table 1, we use BDD for building OV-5a and SV-4 models and activity diagram for

building OV-5b models. Other forms including graphs, matrices, and texts are also used for developing OV-6a, SV-5b, SV-7, SV-8, and SV-10a models.

OV-5a decomposes the mission objective of clearing surface threats into several operational activities including search and detect, track, classify, engage, and command and control. The granularity is kept at a coarse level for the ease of understanding the procedures. Based on OV-5a, OV-5b further clarifies the interdependency relationship and information exchange between the operational activities. As

TABLE 1: Adopted DoDAF views and SysML diagrams.

Purposes	DoDAF views	SysML diagrams	Other forms
Identify operational tasks and activities	OV-5a: operational activity decomposition tree OV-5b: operational activity model	Block definition diagram Activity diagram	
Identify systems and their functionalities	SV-4: systems functionality description	Block definition diagram	
Connect systems with operational activities	SV-5b: operational activity to systems traceability matrix		Allocation matrix
Identify key performance measures of systems	SV-7: systems measures matrix		Matrix
Identify the evolution process	SV-8: systems evolution description		Graph
Identify operational/systems constraints	OV-6a: operational rules model SV-10a: systems rules model		Text description Text description

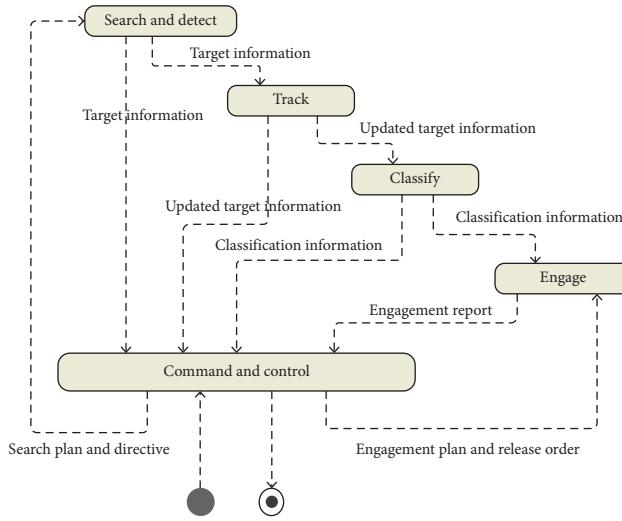


FIGURE 4: SUW operational activity model (OV-5b).

illustrated in Figure 4, target information is the primary information exchanged between different activities and command and control activity processes most of the sensor and communication information. The operational activities depend on a number of physical systems to complete the mission. We identify the potential system alternatives including LCS, helicopter, UAV, and USV and respective system functionalities in SV-4. Each type of system has four variations with different levels of capability. The systems and system functions in SV-4 can both be mapped to operational activities in OV-5b. The mapping between potential systems and operational activities is described in SV-5b, as shown in Figure 5. LCSs can conduct all the operational activities; helicopters and UAVs are able to detect, track, classify, and engage targets; USVs are only responsible for detecting and tracking targets.

SV-7 specifies the qualitative and quantitative measures of resources. In this example, SV-7 in Table 2 lists the selected metrics for the major operational activities and specific measures for each alternative system. The ranges of most input data in the table are extracted from Jacobson's thesis [49]. Flexibility indicates the ability of a system to incorporate new technology; it affects the technology development cost and performance improvement. For the ease

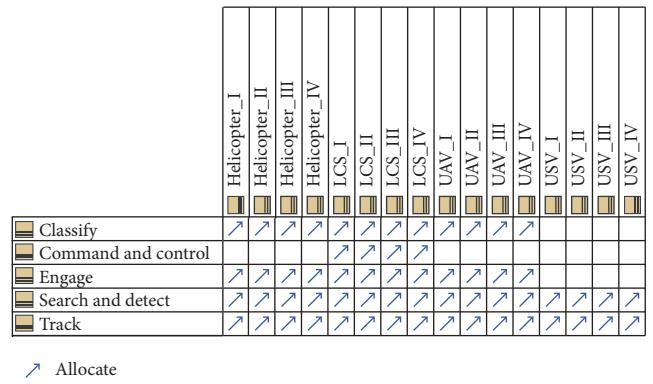


FIGURE 5: SUW operational activity to systems traceability matrix (SV-5b).

of representation, it is assumed that systems of type III and IV are better than systems of type I and II in terms of performance while systems of type II and IV are better than systems of type I and III regarding flexibility.

SV-8 in Figure 6 provides a graphical representation of the evolution process of a helicopter system. The change of a system depends both on internal decisions such as technology development and external environment such as budget change and technology maturation. OV-6a and SV-10a are able to describe the rules constraining the activities and systems, such as budget constraint, quantity constraint, and production capacity constraint.

**3.2. Step II: Mapping DoDAF Models to ADP Elements.** ADP is a flexible modeling and algorithmic framework for solving multistage decision-making problems under uncertainty. Figure 7 illustrates a representative decision-making process of ADP. The circles represent the system states while the rectangles represent the decision nodes. The decision nodes in Figure 7 can be mapped to the decision points "develop/evolve SoS architecture" in the "wave model." Prior to determining the architectural decisions, we need to map the DoDAF models to ADP elements during the SoS analysis phase. The key ADP formulation elements include state variables, decision variables, objective function, transition function, exogenous information, and constraints

TABLE 2: SUW systems measures matrix (SV-7).

Potential systems	Measure types							
	Search and detect		Classify		Engage			
	Range of detection (km)	Prob. of detection	Range of classification (km)	Prob. of classification	Range of attack (km)	Prob. of kill	Weapon capacity	Flexibility
LCS_I	100	0.75	20	0.85	40	0.65	10	Medium
LCS_II	100	0.75	20	0.85	40	0.65	10	High
LCS_III	130	0.9	40	0.9	80	0.8	20	Medium
LCS_IV	130	0.9	40	0.9	80	0.8	20	High
Helicopter_I	50	0.8	10	0.75	10	0.7	8	Medium
Helicopter_II	50	0.8	10	0.75	10	0.7	8	High
Helicopter_III	80	0.9	15	0.85	25	0.8	20	Medium
Helicopter_IV	80	0.9	15	0.85	25	0.8	20	High
UAV_I	40	0.8	8	0.8	10	0.5	8	Medium
UAV_II	40	0.8	8	0.8	10	0.5	8	High
UAV_III	65	0.9	12	0.9	20	0.75	20	Medium
UAV_IV	65	0.9	12	0.9	20	0.75	20	High
USV_I	15	0.8	—	—	—	—	—	Medium
USV_II	15	0.8	—	—	—	—	—	High
USV_III	30	0.9	—	—	—	—	—	Medium
USV_IV	30	0.9	—	—	—	—	—	High

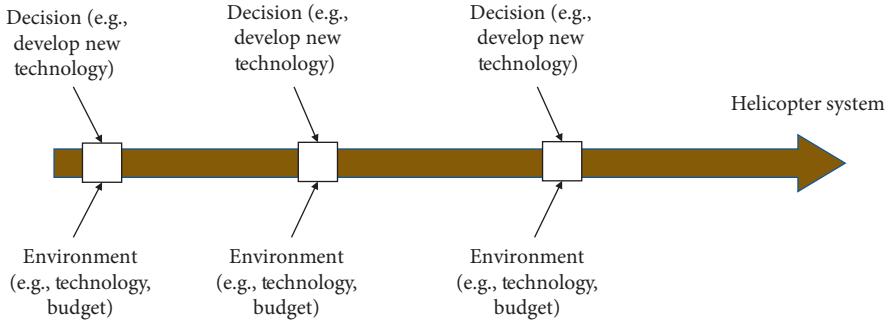


FIGURE 6: SUW systems evolution description (SV-8).

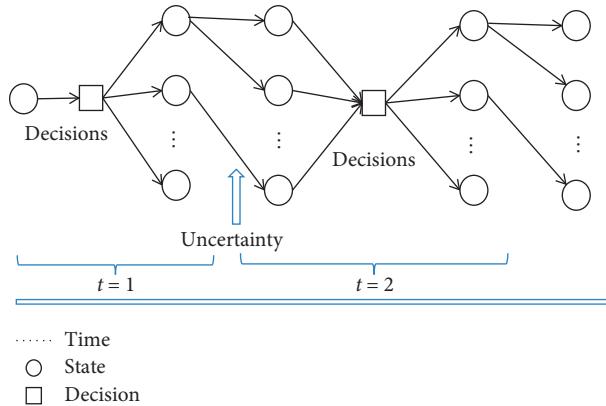


FIGURE 7: Representative decision-making process of ADP.

if necessary [8]. Table 3 illustrates the mapping relationship between the adopted DoDAF models and ADP modeling elements through the metamodel. The metamodels are the fundamental building blocks of DoDAF models, including the data elements of activity, resource, performer, capability, measure, measure type, and many others. For example, the

conceptual metamodel of OV-5 involves activity, resource flow, performer, etc.

*Decision Variables.* The optimization of SUW SoS capability of detecting and destroying enemy boats needs the support of physical systems in SVs to conduct the operational activities in OV-5. The decision variables are

TABLE 3: DoDAF models to ADP elements.

DoDAF views	Metamodel	ADP elements
OV-5a: operational activity decomposition tree	Activities	Objective function
OV-5b: operational activity model	Activities; resource flows	Objective function
OV-6a: operational rules model	Rules	Constraints; objective function
SV-4: systems functionality description	Systems; functions	Decision variable
SV-5b: operational activity to systems traceability matrix	Systems; activities	Decision variable
SV-7: systems measures matrix	Measure types; measures	Objective function; state variable
SV-8: systems evolution description	Systems; resources	Transition function; exogenous information
SV-10a: systems rules model	Rules	Constraints

associated with the physical systems listed in SV-5b in Figure 5. Two types of decisions are under consideration: an integer decision variable  $x_{i,t}^{\text{acq}}$ , describing the number of system  $i$  that is acquired at time  $t$ , and a binary decision variable  $x_{i,t}^{\text{dev}}$ , indicating whether a new technology is developed for system  $i$  at time  $t$ . The technology development decision  $x_{i,t}^{\text{dev}}$  affects the performance of potential systems in a cumulative way.

*State Variables.* The state variables consist of the current state of budget and information about system parameters in SV-7 in Table 2. Other state variables such as the operational tasks and activities are not taken into account because they are assumed to be stable in this paper. The system parameters influencing the contribution of a system  $i$  to an operational activity  $j$  and in turn impacting the SoS capability can be aggregated as a single index  $\text{fcap}_{j,i,t}$ .  $\text{fcap}_{j,i,t}$  can then be used as a representative state variable. This paper employs additive value model to obtain the values of  $\text{fcap}_{j,i,t}$  based on the information in SV-7. According to references [50, 51], additive value model requires identification of key metrics associated with a system function, measures of merit, and scoring functions that normalize different measures of merit to 0 to 100, as shown in the following expression:

$$\text{cap}(l) = \sum_{k=1}^q \omega_k v_k(l_k), \quad (1)$$

where  $k$  is the measure of merit;  $q$  is the total number of measures of merit;  $\omega_k$  is the weight of the  $k$ th measure of merit;  $l_k$  is the level of the  $k$ th measure of merit; and  $v_k(l_k)$  represents the value of the scoring function at level  $l_k$ . The scoring function is a single dimensional value function that assigns value to the system's ability to reach the objective. The following formulation based on references [50, 51] is one of the options for constructing scoring functions:

$$v(\text{MOM}) = 100 \cdot \frac{1 - \exp(\text{MOM}/\rho)}{1 - \exp((\text{MOM}_{\max} - \text{MOM}_{\min})/\rho)}, \quad (2)$$

where  $\text{MOM}$  represents a measure of merit (e.g., range of detection) with maximum capability of  $\text{MOM}_{\max}$  and minimum capability of  $\text{MOM}_{\min}$  and  $\rho$  is the shape parameter that changes the curve and reflects the subjective preferences of the decision makers.

*Objective Function.* According to DoDAF CV-6 (Capability to Operational Activities Mapping), although not explicitly specified in this paper, the capabilities of an SoS

can be fulfilled by the operational activities. That is, the capability contribution of each operational activity in OV-5b in Figure 4 composes the aggregate SoS capability. The basic structure is given as follows:

$$\max_{\{x_{i,t}^{\text{acq}}, x_{i,t}^{\text{dev}}\}} E \left[ \sum_{t=1}^T (g(F_{\text{cap}}_{1,t}, \dots, F_{\text{cap}}_{j,t}, \dots, F_{\text{cap}}_{n,t})) \right], \quad (3)$$

$$F_{\text{cap}}_{j,t} = h(f_{\text{cap}}_{j,1,t}, \dots, f_{\text{cap}}_{j,i,t}, \dots, f_{\text{cap}}_{j,m,t}), \quad (4)$$

where  $F_{\text{cap}}_{j,t}$ , a function of  $f_{\text{cap}}_{j,i,t}$ , represents the capability contributed by operational activity  $j$  to the overall SUW capability at time  $t$  and functions  $g()$  and  $h()$  are parametric models resulting from expert experience, engineering judgement, historical data, or simulation analysis. This paper presumes a linear form for both  $g()$  and  $h()$  for simplicity.

$$\max_{\{x_{i,t}^{\text{acq}}, x_{i,t}^{\text{dev}}\}} E \left[ \sum_{t=1}^T \left( \sum_{j \in \mathcal{J}} (\omega_{\text{fcn}}_j \cdot F_{\text{cap}}_{j,t}) \right) \right], \quad (5)$$

$$F_{\text{cap}}_{j,t} = \sum_{i \in \mathcal{I}} (\omega_{\text{sys}}_{j,i} \cdot f_{\text{cap}}_{j,i,t} \cdot x_{i,t}^{\text{acq}}), \quad (6)$$

where  $\omega_{\text{fcn}}_j$  and  $\omega_{\text{sys}}_{j,i}$  are weighting factors denoting the importance of operational activity  $j$  and system  $i$  in contributing to the overall SUW SoS and activity  $j$ , respectively. Weight selection of a multicriteria optimization problem has been long studied and various approaches are available (for example, analytic hierarchy process, analytic network process, and statistical analysis). This paper assumes that the values of  $\omega_{\text{fcn}}_j$  and  $\omega_{\text{sys}}_{j,i}$  can be elicited through expert opinions. Plugging equation (6) into (5), we obtain the complete objective function.

$$\max_{\{x_{i,t}^{\text{acq}}, x_{i,t}^{\text{dev}}\}} E \left[ \sum_{t=1}^T \left( \sum_{j \in \mathcal{J}} \left( \omega_{\text{fcn}}_j \cdot \sum_{i \in \mathcal{I}} (\omega_{\text{sys}}_{j,i} \cdot f_{\text{cap}}_{j,i,t} \cdot x_{i,t}^{\text{acq}}) \right) \right) \right]. \quad (7)$$

*Exogenous Information.* Exogenous information includes all sort of information that can change the current state, primarily in the form of random variables. According to SV-8 in Figure 6, exogenous information in this study contains the stochastic capability improvement of system  $i$ , denoted as  $\text{dtfcap}_{j,i,t}$ , due to the technology development decisions at each time step, as well as the budget change rate

$\hat{\delta}$ . For the ease of representation, system  $i$ 's capability improvement occurs stochastically only to one of the three operational activities (i.e., search and detect, classify, and engage). Specifically, systems of LCS type and USV type are improved stochastically on the capability of detecting threats; systems of helicopter type are enhanced stochastically on the capability of classifying targets; systems of UAV type are improved stochastically on the capability of engaging threats. The capability increases of all systems are assumed to follow truncated normal distributions.

*Transition Functions.* Transition function indicates the change of SoS states due to the decisions or exogenous information. Based on the evolution demonstration of SV-8, system capability improves when associated technologies are developed. Meanwhile, budget at each time step  $Bgt_t$  changes at a rate of  $\hat{\delta}$ . The equations are described as follows:

$$fcap_{j,i,t+1} = fcap_{j,i,t} + \widehat{dtfcap}_{j,i,t} \cdot x_{i,t}^{\text{dev}}, \quad \forall j \in \mathcal{J}, \forall i \in \mathcal{I}, \quad (8)$$

$$Bgt_{t+1} = Bgt_t \cdot (1 + \hat{\delta}). \quad (9)$$

*Constraints.* The information for formulating the constraints can come from the rules in OV-6a and SV-10a. The SUW objective function is assumed to be subject to budget constraint, production capability constraint, and some other requirement constraints.

$$\sum_{i \in \mathcal{I}^{\text{LCS}}} x_{i,t}^{\text{acq}} \geq \text{SysReq}_t^{\text{LCS}}, \quad \forall t \in \mathcal{T}, \quad (10)$$

$$\sum_{i \in \mathcal{I}^{\text{Helicopter}}} x_{i,t}^{\text{acq}} \geq \text{SysReq}_t^{\text{Helicopter}}, \quad \forall t \in \mathcal{T}, \quad (11)$$

$$\sum_{i \in \mathcal{I}^{\text{UAV}}} x_{i,t}^{\text{acq}} \geq \text{SysReq}_t^{\text{UAV}}, \quad \forall t \in \mathcal{T}, \quad (12)$$

$$\sum_{i \in \mathcal{I}^{\text{USV}}} x_{i,t}^{\text{acq}} \geq \text{SysReq}_t^{\text{USV}}, \quad \forall t \in \mathcal{T}, \quad (13)$$

$$\sum_{i \in \mathcal{I}} (\cos t_{i,t}^{\text{acq}} \cdot x_{i,t}^{\text{acq}}) + \sum_{i \in \mathcal{I}} (\cos t_{i,t}^{\text{dev}} \cdot x_{i,t}^{\text{dev}}) \leq Bgt_t, \quad \forall t \in \mathcal{T}, \quad (14)$$

$$0 \leq x_{i,t}^{\text{acq}} \leq \text{ProdCapacity}_{i,t}, \quad \forall i \in \mathcal{I}, \quad (15)$$

$$0 \leq x_{i,t}^{\text{dev}} \leq 1, \quad \forall i \in \mathcal{I}, \quad (16)$$

where  $\text{SysReq}_t^{\text{LCS}}$ ,  $\text{SysReq}_t^{\text{Helicopter}}$ ,  $\text{SysReq}_t^{\text{UAV}}$ , and  $\text{SysReq}_t^{\text{USV}}$  represent the minimum quantity requirements of systems of LCS type, helicopter type, UAV type, and USV type, respectively, and  $\cos t_{i,t}^{\text{acq}}$  is the acquisition cost of system  $i$  at time  $t$  while  $\cos t_{i,t}^{\text{dev}}$  is the technology development cost for system  $i$  at time  $t$ . Equation (14) constrains both the system acquisition decisions and technology development decisions by the limited budget.  $\text{ProdCapacity}_{i,t}$  indicates production capacity of system  $i$  at time  $t$ .

**3.3. Step III: Building ADP Formulation for SoS Evolution.** The foundation of dynamic programming is the Bellman equation, in which the value of an optimization problem at a specific decision stage is expressed in terms of the immediate payoff and the optimal value of the future objective function. The value at each decision stage is calculated recursively using the Bellman equation. The major issue of dynamic programming is the intractability of calculating expected value function when the sizes of state space, decision space, and sample space grow exponentially. ADP replaces the exact value function by approximate value to reduce the computational complexity while still being able to generate near-optimal solutions [8]. Meanwhile, ADP allows for a straightforward representation of decision-dependent parameters that are difficult to incorporate using dynamic programming. The Bellman equation of ADP can be expressed as follows:

$$V_t(\mathbf{S}_t) = \max_{\{x_{i,t}^{\text{acq}}, x_{i,t}^{\text{dev}}\}} E \left[ \sum_{j \in \mathcal{J}} \left( w \text{fcn}_j \cdot \sum_{i \in \mathcal{I}} (\omega \text{sys}_{j,i} \cdot fcap_{j,i,t} \cdot x_{i,t}^{\text{acq}}) \right) \right. \\ \left. + \gamma \bar{V}_{t+1}(\mathbf{S}_{t+1} | \mathbf{S}_t) \right], \quad (17)$$

where  $\mathbf{S}_t$  denotes the primary state variables that capture the future value. The first portion inside the expectation notation  $E$  describes the capability contributed by decision  $x_{i,t}^{\text{acq}}$  at time  $t$  while the second portion indicates the approximate value function of being in state  $\mathbf{S}_t$ .  $\gamma$  is a discount factor with the value between 0 and 1 that helps decrease the emphasis of future value in the objective function to reduce the approximation errors. Decision makers need to find appropriate approximation strategies for a given problem. For an SoS evolution problem at the architectural level, collecting adequate physical details to build a comprehensive and complex approximation model is challenging and unnecessary. A linear structure for approximating value function as in equation (18) would be an easier and faster solution for high-level decision makers, although some accuracy might lose as a price.

$$\bar{V}_{t+1}(\mathbf{S}_{t+1}) = \boldsymbol{\theta}_{t+1}^T \cdot \boldsymbol{\varphi}(\mathbf{S}_{t+1}) = \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} (\theta_{j,i,t+1} \cdot fcap_{j,i,t+1}), \quad (18)$$

where  $\boldsymbol{\varphi}(\mathbf{S}_{t+1})$  refers to basis functions that capture the important features contributing to the overall capability. The major state variables  $fcap_{j,i,t}$  are selected for the basis functions in the SUW example.  $\boldsymbol{\theta}_{t+1}^T$  represents the vector of adjusting parameters  $\theta_{j,i,t+1}$  that enable different approximate value functions. After replacing the exact value function with approximate one, equation (17) is still challenging to solve due to the computation of expectation. We adopt a concept of postdecision state variable proposed by Powell [8] to separate the effect of decisions and exogenous information. The expectation can then be removed by using sample realizations of uncertain parameters. The new objective function can be expressed as follows:

$$V_t(\text{fcap}_{j,i,t}) = \max_{\{x_{i,t}^{\text{acq}}, x_{i,t}^{\text{dev}}\}} \sum_{j \in \mathcal{J}} \left( w \text{fcn}_j \cdot \sum_{i \in \mathcal{I}} (\text{wsys}_{j,i} \cdot \text{fcap}_{j,i,t} \cdot x_{i,t}^{\text{acq}}) \right) + \gamma \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}} (\theta_{j,i,t+1} \cdot \text{fcap}_{j,i,t+1}). \quad (19)$$

This study employs a least square temporal difference (LSTD) learning method to update the coefficient parameters in the approximation. LSTD learns by sampling the exogenous information and approximates its current estimate based on previously learned estimates. The updating equations can be expressed as follows according to reference [8].  $\hat{v}_t^n$  is a sample estimate obtained by solving equation (19) with coefficient  $\boldsymbol{\theta}^{n-1}$  at iteration  $n$ .

$$\boldsymbol{\theta}^n = \boldsymbol{\theta}^{n-1} - \frac{1}{1 + (\boldsymbol{\varphi}^n)^T \mathbf{B}^{n-1} \boldsymbol{\varphi}^n} \mathbf{B}^{n-1} \boldsymbol{\varphi}^n (\bar{V}_t(\mathbf{s}_t^n | \boldsymbol{\theta}^{n-1}) - \hat{v}_t^n), \quad (20)$$

$$\mathbf{B}^n = \mathbf{B}^{n-1} - \frac{1}{1 + (\boldsymbol{\varphi}^n)^T \mathbf{B}^{n-1} \boldsymbol{\varphi}^n} (\mathbf{B}^{n-1} \boldsymbol{\varphi}^n (\boldsymbol{\varphi}^n)^T \mathbf{B}^{n-1}). \quad (21)$$

**3.4. Step IV: Generating SoS Evolution Solutions.** To generate SoS evolution solutions, we need to solve the mathematical formulation discussed in the previous sections. The algorithmic procedure for solving the ADP problem is summarized as follows:

- Step 1: Set  $n = 1$ . Initialize  $\bar{V}_t^{n=1}, t = \{1, \dots, T\}$ .
- Step 2: Sample the uncertain exogenous information (for example,  $\widehat{\text{dtfcap}}_{j,i,t}$  in this study).
- Step 3: For  $t = 1, \dots, T$ , do:
  - Step 3a: Determine decision variables  $x_{i,t}^{\text{acq},n}, x_{i,t}^{\text{dev},n}$  by solving equation (19).
  - Step 3b: Update  $\bar{V}_{t-1}^n$  by updating the coefficient vector  $\boldsymbol{\theta}$  according to equations (20) and (21).
  - Step 3c: Update system states using equation (8).
- Step 4: Let  $n = n + 1$ . If  $\|\hat{v}_{t=1}^n - \hat{v}_{t=1}^{n-1}\| \leq \epsilon$  or  $n < N$ , go to Step 2.

## 4. Results and Discussion

According to the DoDAF models enabled mathematical formulation and the adopted ADP algorithm provided in the previous section, this paper uses “intlinprog” function in Matlab to solve the integer programming problem at each time step for every iteration. The function “intlinprog” might be inefficient for large size problems, in which case we can switch to more efficient commercial computation tools such as CPLEX and Gurobi optimizers. This result section primarily aims to demonstrate the effectiveness of the ADP method in supporting SoS evolution while the usefulness of integrating MBSE and ADP will also be discussed in the last subsection.

**4.1. Convergence Analysis.** Before generating any decisions that could be useful, the first step is always checking the convergence status of the ADP algorithm. Since the deterministic experiment is a special case of the stochastic experiment, we only plot the convergence analysis figure of the stochastic experiment where the capability improvement  $\widehat{\text{dtfcap}}_{j,i,t}$  of each type of system follows a normal distribution. The initial budget is set as 1000 million dollars and increases at a constant rate of 5% during every decision stage. Figure 8 illustrates the convergence status using the linear value function approximation in equation (18). The objective value (i.e., the value function at the first stage) converges within 100 iterations. Fluctuations exist but remain in a small range, around 5% of the mean objective value.

**4.2. Solution Comparison.** Since the decision-dependent parameters are difficult to capture in many methods, the comparison starts with experiments that do not include the technology development decisions. The value function converges slowly in this case, so we set a small discount factor that equals 0.5 to increase the convergence rate. The discount factor does not impact the comparison once the ADP algorithm and the regular single-stage optimization choose the same discount factor. Figure 9 demonstrates the approximate total value (i.e., value function at the first stage) at each iteration and the exact total value obtained by adding together the capability value at each decision stage solved by single-stage integer programming. After 1000 iterations, the difference between the approximate objective value and exact one is within 1%, which demonstrates the effectiveness of value function approximation strategy with  $\text{fcap}_{j,i,t}$  as basis functions.

With technology development decisions incorporated, we begin with a simplified version as a three-stage problem with fixed capability increase because the increasing number of decision stages quickly increases the intractability of the problem. Figure 10 compares ADP solution to the solution of a nonlinear programming (particularly genetic algorithm). Based on the plot, both the objective value at each stage and the total objective value over three stages obtained by ADP (i.e., area under the solid blue curve) are larger than those values from the genetic algorithm. That is, the result from the ADP method is closer to the optimal solution than that from the genetic algorithm that was adopted by quite a few papers to support SoS evolution.

In the stochastic experiment with technology development decisions, Figure 11 compares the capability value obtained from the ADP algorithm at each decision stage with four other test cases under a special scenario where the mean values of the stochastic parameters are used as representatives. In those test cases, decision makers are not sure about the ratio between budget on system acquisition decisions and budget on technology investment decisions. Hence, we use four test cases, where the percentage of technology investment is 0%, 10%, 20%, and 50%, respectively, to represent the possible decisions. Based on the plot, only Case 1 provides a solution close to the results from ADP while some of the cases such as Case 4 provide pretty bad

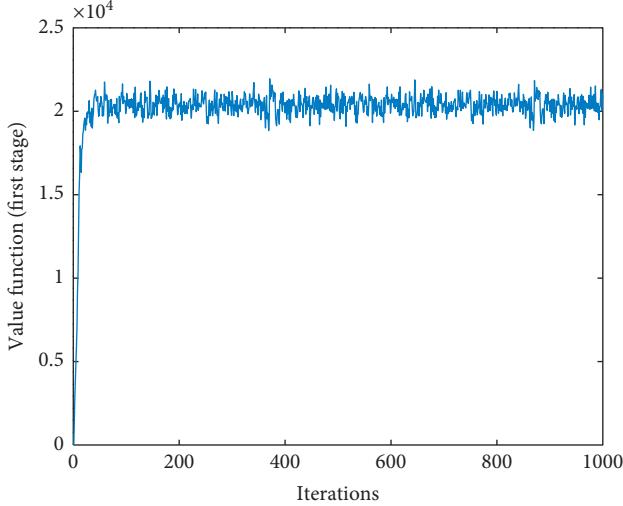


FIGURE 8: Value function convergence status.

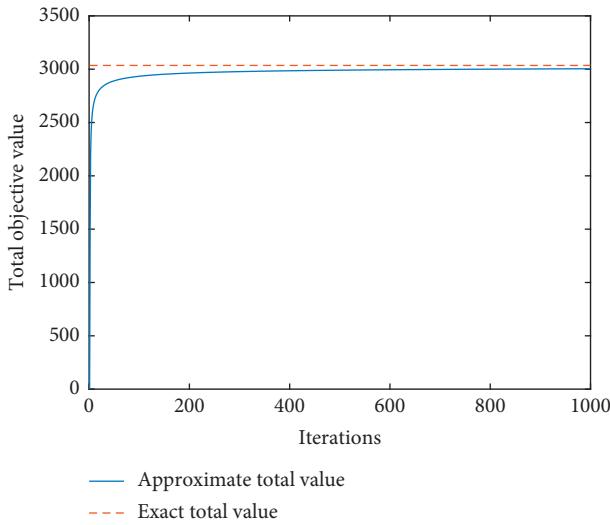


FIGURE 9: Comparison between approximate and exact values in the experiment without technology development decisions.

solutions. The total capability values over ten stages (i.e., area under each curve) of these five cases are 20302, 18052, 20023, 17864, and 11010, respectively. Although the ADP solution does not reach the exact optimal value, it outperforms all the other example solutions (and many others not listed) in this experiment. Although decision makers might still be able to pick the right decision, the loss for the bad decisions could be significant yet avoidable.

**4.3. Demonstration of SoS Evolution Decisions.** The SoS evolution decisions in this example include the number of systems acquired to join the SUW SoS and whether new technologies are developed for a specific system. Figure 12 demonstrates the system addition decisions in the form of mean values. Note that the actual decision that will be implemented is solely the current decision although the evolutionary decisions are generated over ten decision

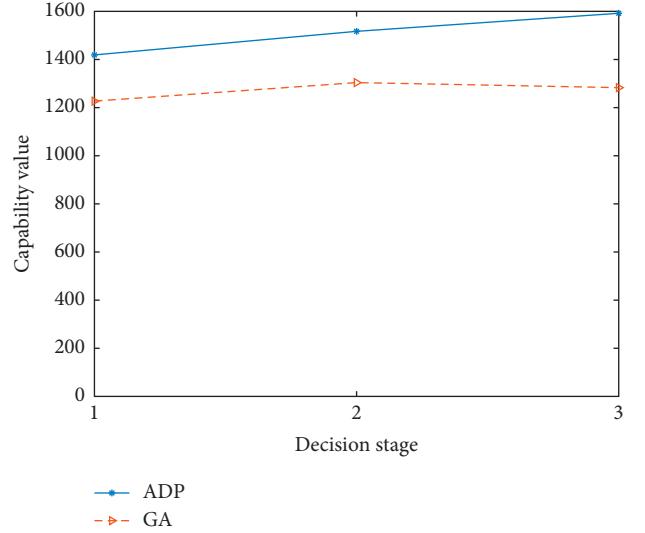


FIGURE 10: Comparison between objective values from ADP and nonlinear programming for a deterministic three-stage problem.

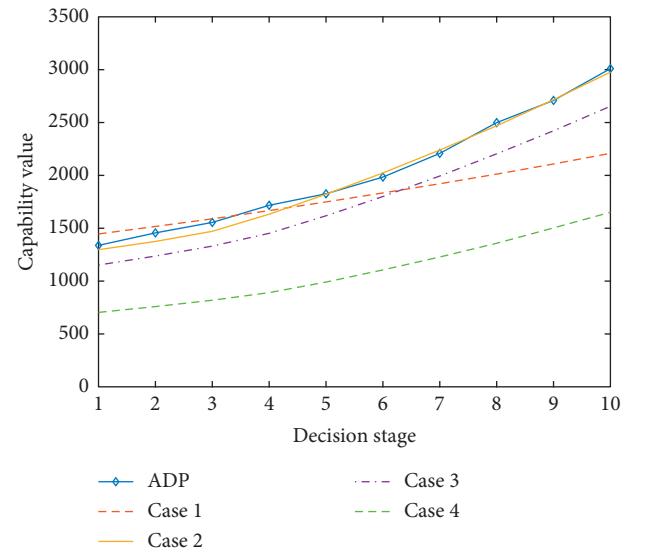


FIGURE 11: Solution comparison between ADP and other test cases (Case 1: no technology investment; Case 2: 10% technology investment; Case 3: 20% technology investment; Case 4: 50% technology investment).

stages. Based on the graph, LCS\_III, LCS\_IV, Helicopter\_III, UAV\_IV, and USV\_IV take up the majority part of the system acquisition plan. Recall that type III and IV systems have better performances than type I and II systems while type II and IV systems are more flexible to adapt to changes than type I and III systems. Although type I systems are cheaper, almost none are selected. On the contrary, type IV systems are the most attractive options. An interesting observation is that more type III systems are chosen than type II systems. Higher performance seems more favorable to decision makers than higher flexibility in this example. If the benefit provided by the flexibility increases, the result might change accordingly. We can also observe that the

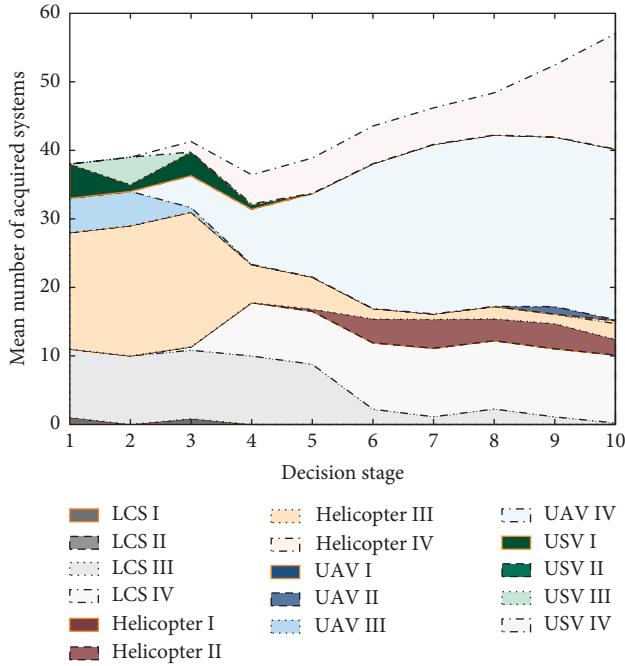


FIGURE 12: Mean number of acquired systems.

number of type IV systems increases quickly in the later stages; this is because the cumulative benefit earned by technology development and flexible design surpasses the cost at a certain point.

**4.4. Discussion.** The results in this section demonstrate the effectiveness of ADP in supporting SoS evolution by analyzing the convergence status and comparing solutions from ADP and other alternative methods. The mathematical equations in the previous section are of course not the only way to formulate the SoS evolution problem, but they did provide an idea for decision makers on how to extract useful information from DoDAF models, put the information together as mathematical expressions, and generate quantitative solutions for SoS evolution. Although we have made quite a few assumptions to simplify the process, such as linear parametric models used in the objective function, linear value function approximation, and simple exogenous information, these perspectives can definitely be improved by more sophisticated methods. We also used some synthetic input data, such as the capability improvement  $dtfcap_{j,it}$  and system quantity requirements; fortunately, they do not affect the essence of the proposed method and can be replaced by real data.

The function of the DoDAF models is to provide a common platform to enable more effective communication between different types of stakeholders involved in an SoS. The bridge made between DoDAF models and the ADP method allows the decision makers to directly generate evolutionary decisions based on the information extracted from the DoDAF models. On the one hand, the link increases the traceability between conceptual models and

analytical methods. On the other hand, when users employ DoDAF models to conduct other types of analysis, the source information and data will be kept as the same for different types of analysis. Decision makers can thus investigate and analyze an SoS problem from different facets using different methods with less information mismatch. This paper has not realized the automated conversion process from DoDAF models to ADP modeling elements yet, and the manual translation through metamodel can be considered as a preliminary step towards the automated process.

## 5. Conclusions

Using MBSE artifacts as a common input source for different SoS architectural analysis has been a pressing need for many industry and government agencies. This paper employed MBSE artifacts, more specifically DoDAF models using SysML, and the ADP method to provide decision support for SoS evolution in the form of “wave model.” Using a motivating SUW SoS example, this paper developed an entire framework from DoDAF model development, mapping between DoDAF models to ADP elements, to ADP formulation construction and solution generation. The key contribution of this paper is building the bridge between DoDAF models and ADP elements through metamodels including resources, activities, systems, rules, and so on. Although not fully automated yet, the integration can help decrease the possibility of information loss during the quantitative analysis and allow for a holistic understanding of an SoS problem. Moreover, this paper formulated the SoS evolution problem as a multistage portfolio planning problem involving decision-dependent parameters, uncertainties, etc., and made full use of ADP in addressing such a type of problems. The results demonstrated the effectiveness of ADP in achieving a near-optimal solution, even with simple linear value function approximation. Finally, the proposed method is not only suitable for ADP method but also provides an idea for decision makers on how to build connections between DoDAF conceptual models and other mathematical optimization methods.

Research in this paper can be considered as a preliminary step towards fully automated MBSE process. The manual connection between DoDAF models and ADP formulation still imposes a substantial burden on the SoS engineers. Therefore, part of the future work will focus on deriving mathematical formulation from DoDAF models in an automatic or partially automatic manner. This requires a clear understanding of the logic between metamodels in the DoDAF models and the logic between the ADP elements. In addition, many assumptions in the formulation can be relaxed in the future. For example, current ADP formulation has not captured the interaction effect among operational activities described in OV-5b yet. Furthermore, the proposed work should be applied to a large-scale and fully detailed problem in the future.

## Nomenclature

$j:$	Index of operational activity
$\mathcal{J}:$	Set of all operational activities
$n:$	Total number of operational activities
$i:$	Index of system
$\mathcal{I}:$	Set of all systems
$m:$	Total number of systems
$t:$	Index of time step or decision stage
$\mathcal{T}:$	Set of all time steps or decision stages
$Fcap_{j,t}:$	Capability contributed by activity $j$ to SoS at time $t$
$fcap_{j,i,t}:$	Capability contributed by system $i$ to activity $j$ at time $t$
$wfcn_j:$	Weight of activity $j$ in contributing to SoS
$\omega_{sys,j,i}:$	Weight of system $i$ in completing activity $j$
$x_{i,t}^{acq}:$	Number of system $i$ that is acquired at time $t$
$x_{i,t}^{dev}:$	Whether a new technology is developed for system $i$ at time $t$
$\widehat{dtfcap}_{j,i,t}:$	Capability improvement of system $i$ to activity $j$ at time $t$
$Bgt_t:$	Budget at time $t$
$\delta:$	Budget increase rate
$cost_{i,t}^{acq}:$	Acquisition cost of system $i$ at time $t$
$cost_{i,t}^{dev}:$	Technology development cost of system $i$ at time $t$
$SysReq_t^{LCS}:$	Quantity requirement of LCS system
$SysReq_t^{Helicopter}:$	Quantity requirement of helicopter system
$SysReq_t^{UAV}:$	Quantity requirement of UAV system
$SysReq_t^{USV}:$	Quantity requirement of USV system
$ProdCapacity_{i,t}:$	Production capacity of system $i$ at time $t$
$\overline{V}_{t+1}:$	Value function approximation from $t + 1$ to the end
$\gamma:$	Discount factor
$\varphi:$	Basis functions for value function approximation
$\Theta_t^T:$	Coefficient vector of basis functions for value function approximation
$\hat{v}_t^n:$	Sample estimate of value function at iteration $n$ at time $t$ .

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grant no. 61273207, Independent Innovation Funding of Huazhong University of Science and Technology under grant no. 2019kfyXJJS17, and Natural Science Foundation of Hubei Province under grant no. 2014CFB581.

## References

- [1] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.
- [2] W. C. Baldwin, T. Ben-Zvi, and B. J. Sauser, "Formation of collaborative system of systems through belonging choice mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 4, pp. 793–801, 2012.
- [3] R. Raman and M. D'Souza, "Decision learning framework for architecture design decisions of complex systems and system-of-systems," *Systems Engineering*, vol. 22, no. 6, pp. 538–560, 2019.
- [4] OUSD AT&L, *Systems Engineering Guide for Systems of Systems*, 2008.
- [5] Department of Defense, *Defense Acquisition Guidebook*, Defense Acquisition University, Fort Belvoir, VA, USA, 2008.
- [6] J. Dahmann, G. Rebovich, J. Lane, and K. Baldwin, "An implementer's view of systems engineering for systems of systems," in *Proceedings of the 2011 IEEE International Systems Conference*, Nanjing, China, September 2011.
- [7] C. Piaszczyk, "Model based systems engineering with department of defense architectural framework," *Systems Engineering*, vol. 14, no. 3, pp. 305–326, 2011.
- [8] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, Wiley, Hoboken, NJ, USA, 2nd edition, 2010.
- [9] M. LaSorda, J. M. Borky, and R. M. Sega, "Model-based architecture and programmatic optimization for satellite system-of-systems architectures," *Systems Engineering*, vol. 21, no. 4, pp. 372–387, 2018.
- [10] C. Guariniello, Z. Fang, N. Davendralingam, K. Marais, and D. DeLaurentis, "Tool suite to support model based systems engineering enabled system-of-systems analysis," in *Proceedings of the 2018 IEEE Aerospace Conference*, Big Sky, MT, USA, March 2018.
- [11] J. Huff, H. Medal, and K. Griendl, "A model-based systems engineering approach to critical infrastructure vulnerability assessment and decision analysis," *Systems Engineering*, vol. 22, no. 2, pp. 114–133, 2019.
- [12] M. Lehman and J. C. Fernandez-Ramil, "Software evolution," in *Software Evolution and Feedback: Theory and Practice*, John Wiley & Sons, Ltd, Hoboken, NJ, USA, 2006.
- [13] S. P. Welby, *Modular Open Systems Architecture in DoD Acquisition*, <https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2014/system/16943WedTrack2Welby.pdf>, 2014.
- [14] R. Lock, "Developing a methodology to support the evolution of system of systems using risk analysis," *Systems Engineering*, vol. 15, no. 1, pp. 62–73, 2012.
- [15] D. Carney, D. Fisher, and P. Place, *Topics in Interoperability: System-of-Systems Evolution*, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA, USA, 2005.
- [16] M. M. Lehman, "Programs, life cycles, and laws of software evolution," *Proceedings of the IEEE*, vol. 68, no. 9, pp. 1060–1076, 1980.
- [17] M. M. Lehman, J. F. Ramil, P. D. Wernick, D. E. Perry, and W. M. Turski, "Metrics and laws of software evolution-the nineties view," in *Proceedings of the Fourth International Software Metrics Symposium*, Albuquerque, NM, USA, November 1997.
- [18] R. d. Neufville, "Dynamic strategic planning for technology policy," *International Journal of Technology Management*, vol. 19, no. 3/4/5, pp. 225–245, 2000.

- [19] O. L. De Weck, R. De Neufville, and M. Chaize, "Staged deployment of communications satellite constellations in low earth orbit," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, no. 3, pp. 119–136, 2004.
- [20] J. Buurman, S. Zhang, and V. Babovic, "Reducing risk through real options in systems design: the case of architecting a maritime domain protection system," *Risk Analysis*, vol. 29, no. 3, pp. 366–379, 2009.
- [21] M. E. Fitzgerald and A. M. Ross, "Mitigating contextual uncertainties with valuable changeability analysis in the multi-epoch domain," in *Proceedings of the IEEE Syscon 2012*, Vancouver, BC, Canada, March 2012.
- [22] M. E. Fitzgerald and A. M. Ross, "Sustaining lifecycle value: valuable changeability analysis with era simulation," in *Proceedings of the IEEE Syscon 2012*, Vancouver, BC, Canada, March 2012.
- [23] M. R. Silver and O. L. De Weck, "Time-expanded decision networks: a framework for designing evolvable complex systems," *Systems Engineering*, vol. 10, no. 2, pp. 167–188, 2007.
- [24] P. Davison, B. Cameron, and E. F. Crawley, "Technology portfolio planning by weighted graph analysis of system Architectures," *Systems Engineering*, vol. 18, no. 1, pp. 45–58, 2015.
- [25] W. Tan, B. J. Sauser, J. E. Ramirez-Marquez, and R. B. Magnaye, "Multiobjective optimization in multifunction multicapability system development planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 785–800, 2013.
- [26] P. Acheson, L. Pape, C. Dagli, N. Kilicay-Ergin, J. Columbi, and K. Haris, "Understanding system of systems development using an agent-based wave model," in *Proceedings of the Complex Adaptive Systems*, Washington, DC, USA, November 2012.
- [27] M. Maier, "Research challenges for systems-of-systems," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Waikoloa, HI, USA, October 2005.
- [28] W. Powell, B. Ayari, J. Berger et al., "The effect of robust decisions on the cost of uncertainty in military airlift operations," *ACM Transactions on Modeling and Computer Simulation*, vol. 9, no. 4, pp. 39–57, 2010.
- [29] W. Powell, A. George, H. Simao et al., "SMART: a stochastic multiscale model for the analysis of energy resources, technology, and policy," *Informs Journal on Computing*, pp. 1–18, 2011.
- [30] M. Webster, N. Santen, and P. Parpas, "An approximate dynamic programming framework for modeling global climate policy under decision-dependent uncertainty," *Computational Management Science*, vol. 9, no. 3, pp. 339–362, 2012.
- [31] Nusawardhana, *Dynamic Programming Methods for Concurrent Design and Dynamic Allocation of Vehicles Embedded in a System-of-Systems*, Purdue University, West Lafayette, IN, USA, 2007.
- [32] Z. Fang and D. DeLaurentis, "Dynamic planning of system of systems architecture evolution," *Procedia Computer Science*, vol. 28, pp. 449–456, 2014.
- [33] Z. Fang and D. DeLaurentis, "Multi-stakeholder dynamic planning of system of systems development and evolution," *Procedia Computer Science*, vol. 44, pp. 95–104, 2015.
- [34] Z. Fang, N. Davendralingam, and D. DeLaurentis, "Multi-stakeholder dynamic optimization for acknowledged system-of-systems architecture selection," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3565–3576, 2018.
- [35] International Council on Systems Engineering, *Systems Engineering Vision 2020*, 2007.
- [36] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*, Morgan Kaufmann OMG Press, Burlington, MA, USA, 2012.
- [37] M. Mori, A. Ceccarelli, P. Lollini, B. Fromel, F. Brancati, and A. Bondavalli, "Systems-of-Systems modeling using a comprehensive viewpoint-based SysML profile," *Journal of Software: Evolution and Process*, vol. 30, pp. 1–20, 2017.
- [38] J. A. Lane and T. Bohn, "Using SysML modeling to understand and evolve systems of systems," *Systems Engineering*, vol. 16, no. 1, pp. 87–98, 2013.
- [39] M. Hause, "A solutions based approach to MBSE architectures with UPDM," in *Proceedings of the INCOSE International Symposium*, Philadelphia, PA, USA, 2013.
- [40] J. L. Hayden and A. Jeffries, "On using SysML, DoDAF 2.0 and UPDM to model the architecture for the NOAA's joint polar satellite system (JPSS) ground system (GS)," in *Proceedings of the 12th International Conference on Space Operations*, Stockholm, Sweden, 2012.
- [41] S. W. Mitchell, *Complex Product Family Modeling for Common Submarine Combat System MBSE*, Lockheed Martin, Bethesda, MD, USA, 2010, [http://www.omg.sysml.org/Complex\\_Product\\_Family\\_Modeling-SWFTS\\_IC-MBSE\\_2010\\_briefing.pdf](http://www.omg.sysml.org/Complex_Product_Family_Modeling-SWFTS_IC-MBSE_2010_briefing.pdf).
- [42] L. W. Wagenhals and A. H. Levis, "Service oriented architectures, the DoD architecture framework 1.5, and executable Architectures," *Systems Engineering*, vol. 12, no. 4, pp. 312–343, 2009.
- [43] R. Wang and C. H. Dagli, "Executable system Architecting using systems Modeling Language in conjunction with colored Petri nets in a model-driven systems development process," *Systems Engineering*, vol. 14, no. 4, pp. 383–409, 2011.
- [44] K. Griendlung and D. N. Mavris, "Development of a DoDAF-based executable Architecting approach to analyze system-of-systems alternatives," in *Proceedings of the 2011 Aerospace Conference*, San Anton, Malta, May 2011.
- [45] B. G. Kang, K.-M. Seo, and T. G. Kim, "Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network," *Complexity*, vol. 2018, Article ID 6201356, 16 pages, 2018.
- [46] B. G. Kang, S. H. Choi, S. J. Kwon, J. H. Lee, and T. G. Kim, "Simulation-based optimization on the system-of-systems model via model transformation and genetic algorithm: a case study of network-centric warfare," *Complexity*, vol. 2018, Article ID 4521672, 15 pages, 2018.
- [47] M. Hernandez, C. H. Jackson, O. Meza et al., *MH-60 Seahawk /MQ-8 Fire Scout Interoperability*, Naval Postgraduate School, Monterey, CA, USA, 2010.
- [48] J. Broadmeadow, N. Nguyen, F. Dziekan et al., *Improved Anti-Submarine Warfare (ASW) Effectiveness*, Naval Postgraduate School, Monterey, CA, USA, 2008.
- [49] K. R. Jacobson, *The Littoral Combat Ship (LCS) Surface Warfare (SUW) Module: Determining the Best Mix of Surface-To-Surface and Air-To-Surface Missiles*, Naval Postgraduate School, Monterey, CA, USA, 2010.
- [50] G. S. Parnell, R. E. Metzger, J. Merrick, and R. Eilers, "Multiobjective decision analysis of theater missile defense architectures," *Systems Engineering*, vol. 4, no. 1, pp. 24–34, 2001.
- [51] G. S. Parnell, H. W. Conley, J. A. Jackson, L. J. Lehmkuhl, and J. M. Andrew, "Foundations 2025: a value model for evaluating future air and space forces," *Management Science*, vol. 44, no. 10, pp. 1336–1350, 1998.