NPS-FM-20-008

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

An Approximate Dynamic Programming Approach for Weapon System Financial Execution Management

November 12, 2019

Dr. Erich Morman, Assistant Professor
Graduate School of Defense Management

Dr. Jefferson Huang, Assistant Professor
Graduate School of Operational and Information Sciences

Naval Postgraduate School

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

Abstract

Each year, the Department of Defense (DoD) fiscal calendar starts on October 1 and ends on September 30. Once a fiscal year (FY) begins, weapon system program offices, agencies, and other divisions throughout the DoD serve as the stewards for their budgets. In this role, these offices are tasked with the responsibility of ensuring that congressionally appropriated funding is allocated efficiently over the entirety of the 12-month FY cycle. Furthermore, the DoD financial execution process operates under use-or-lose budgetary regulations. As the calendar moves closer to the end of the FY, September 30, DoD offices undergo a FY closeout review. Dollars that are not adequately spent are at risk of being pulled-back or “swept-up.” In other words, funding can be taken away from an office that is underspending and essentially removed from their FY appropriated budget. During the FY closeout process each year, considerable time and energy is invested in assessing cash utilization levels (disbursements) across the DoD and then implementing where necessary the required “sweep-up” actions. In this research, we investigate the construct of using a learning algorithmic approach known as approximate dynamic programming (ADP) for modeling use-or-lose budgetary systems. ADP is a prescriptive analytics approach used to model sequential decision-making problems under uncertainty. In the context of use-or-lose budgets, we look to leverage ADP in order to generate an efficient month-to-month cash allocation policy in order to minimize the amount of both underspending and overspending that occurs during the FY closeout period. The research presents a framework for modeling and simulating use-or-lose budgets using ADP and discusses the computational complexity and the implications for leveraging the ADP approach in practice.

THIS PAGE LEFT INTENTIONALLY BLANK

About the Authors

Dr. Erich Morman is an assistant professor at the Naval Postgraduate School in the Defense Resources Management Institute (DRMI). He received a BA in Mathematics and Economics (1996) from the University of Dayton and an MS in Systems Engineering & Operations Research (2003) and a PhD in Systems Engineering & Operations Research (2013) from George Mason University. Prior to joining DRMI, Dr. Morman spent 17 years with the Missile Defense Agency (MDA) supporting various weapon system program offices in the area of Planning, Programming, Budgeting, and Execution. While working for the MDA, he completed his dissertation, which focused on applications of reinforcement learning algorithms to improve fiscal year cash allocation decisions. Additionally, Dr. Morman spent nearly two years as a support contractor for the DoD's Suicide Prevention Office (DSPO) conducting measures of effectiveness and data analysis.

Dr. Jefferson Huang is an assistant professor in the Operations Research Department at the Naval Postgraduate School. Prior to joining NPS in 2018, Dr. Huang was a postdoctoral associate with the School of Operations Research and Information Engineering at Cornell University. He received a PhD in Applied Mathematics (2016) from Stony Brook University, with a dissertation on the computational complexity of algorithms for Markov decision processes. Dr. Huang also holds an MS in transportation from the Massachusetts Institute of Technology and a BS in civil and environmental engineering from the University of California at Berkeley. His research interests include applied probability, optimization under uncertainty, and operations management.

THIS PAGE LEFT INTENTIONALLY BLANK

NPS-FM-20-008

ACQUISITION RESEARCH PROGRAM SPONSORED REPORT SERIES

An Approximate Dynamic Programming Approach for Weapon System Financial Execution Management

November 12, 2019

Dr. Erich Morman, Assistant Professor
Graduate School of Defense Management

Dr. Jefferson Huang, Assistant Professor
Graduate School of Operational and Information Sciences

Naval Postgraduate School

Disclaimer: The views represented in this report are those of the authors and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

THIS PAGE LEFT INTENTIONALLY BLANK

List of Acronyms

ADP	Approximate Dynamic Programming
CAKP	Commitment Allocation Knapsack Problem
CRA	Continuing Resolution Authority
DAR-Q	Dormant Account Review—Quarterly
DAU	Defense Acquisition University
DFAS	Defense Finance and Accounting Service
DoD	Department of Defense
DoD OIG	DoD Office of Inspector General
FY	Fiscal Year
IP	Integer Programming
mdi	*maximum demand increment*
MSE	Mean Square Error
MILCON	Military Construction
MILPERS	Military Personnel
NDAA	National Defense Authorization Act
O&M	Operation and Maintenance
OSD	Office of the Under Secretary
OUSD(C)	Office of the Under Secretary of Defense (Comptroller)
PM	Program Manager

THIS PAGE LEFT INTENTIONALLY BLANK

List of Figures

THIS PAGE LEFT INTENTIONALLY BLANK

1.0 Introduction

Over the course of a fiscal year (FY) cycle, Department of Defense (DoD) weapon system program offices are required to make quality cash allocation determinations. As the FY moves forward in time, a program office must decide which projects to fund and how much funding a project should receive. If a weapon system program office allocates more funding to a project than what that project can utilize, it runs the risk of overfunding the effort and having these dollars taken during a FY closeout process. If a weapon system program office allocates insufficient funding for the project, it risks underfunding the effort, which can result in a work stoppage or other delays.

As with most public sector organizations, DoD money that is managed by weapon system program offices contains an expiration point. Dollars not spent or utilized within a defined time frame are taken away and are no longer available as a resource for paying for support projects or activities. Organizations that manage money with this type of constraint are operating with what is informally referred to as a use-or-lose budget. Functioning under this framework, weapon system program managers and their financial officers must consider how to strategically allocate funding over an annual time horizon that balances between the immediate day-to-day cash allocation decisions and the aggregate long-term impact these decisions will have on the program office's FY financial closeout position.

There is ample evidence to indicate that a sizable portion of the DoD budget expires each year. The National Defense Authorization Act of 2014 (FY2014 NDAA) required that the DoD's financial statements undergo a full audit in FY2018. As a result of that audit, the DoD's Office of Inspector General (DoD OIG) reported that there was a total of $27.7 billion of expired department funding in FY2018. These dollars represent funds that went unutilized over a five-year time span that started in FY2013.

Furthermore, while analyzing FY2012 DoD budget data, Conley, Dominy, Kneece, Mandelbaum, and Whitehead (2014) pointed out that the rate of spending as measured by expenditure rates across the DoD was declining for several years prior.

The report highlights how spending benchmarks issued by the Office of the Secretary of Defense (OSD) are based on 30 years of financial execution history. Theoretically, this means that DoD spending benchmarks are correlated to the work schedules and associated spending patterns that are emblematic of the acquisition efforts within a typical DoD weapon system program office. However, the actual acquisition experience for each weapon system program is unique and always evolving, compounding the difficulties faced by weapon system program managers, business financial managers, and their staff.

Serving as additional evidence that there are cash flow problems within the DoD, a 2013 Defense Acquisition University (DAU) study by Tremaine & Seligman (2013) provides a summary of survey results from 229 DoD personnel who responded to questions regarding the top challenges they see as factors impeding cash flow and hindering the ability of a program office to meet the OSD's spending benchmarks. The report highlighted a myriad of growing challenges and endogenous issues that DoD personnel working in a weapon system program office contend with on a routine basis. The following is a short list of standard problems that are impediments and bottlenecks to efficiently allocating and spending money in a timely manner:

- The more routine use of continuing resolution authorities (CRAs) by Congress to issue yearly budgets through multiple installments
- Congressional marks or program cuts
- Delays in contract negotiations and awards
- A high volume of contract modifications related to warfighter requirement changes
- Constant rotation or shortages of key program office personnel
- Complications with getting funding documents issued and approved in a timely manner
- An inability to obtain timely data on contractor outlays or expenditure positions

Also, it is reasonable to assume that given the pressure to adequately meet end-of-year spending benchmarks and as the FY closeout period draws nearer, program offices will look to quickly allocate funding to unnecessary and wasteful endeavors that are able to quickly spend the dollars and artificially elevate financial performance statistics. The annually recurring news headlines during the FY closeout month of September suggests that this is a common problem and that the DoD continues to have problems with efficient cash allocations (Mehta, 2018; Moritz-Rabson, 2018).

We look to the use of ADP as a solution approach to the financial execution problem for weapon system program offices. Fundamentally, the financial execution problem is a dynamic sequential resource allocation problem, where the resource variable in question is the amount of cash that is committed to projects on a daily basis. Although use-or-lose budget resource problems are not explicitly addressed, there are a number of publications that highlight ADP's applicability to solving other various types of resource allocation problems. ADP contains a number of features that make it an attractive tool for the financial execution challenges associated with use-or-lose budgets. First, ADP is a well-established prescriptive analytical tool. It is also designed to create a sequential decision-making policy. In the case of the financial execution problem, a program office must consider a cash allocation policy over a fiscal year that provides an appropriate level of funding installments to projects that minimizes the amount of vulnerable end-of-year money. Second, ADP "learns" a financial execution policy by iteratively interacting with the decision environment. Lastly, the ADP methodology can be adjusted to incorporate the uncertainty and stochastic information of separate program offices. In this manner, ADP can be specialized for individual program offices to more readily account for their unique financial challenges and circumstances.

THIS PAGE LEFT INTENTIONALLY BLANK

2.0 Literature Review of ADP

Dynamic programming has a history as a mathematical tool for modeling and solving sequential decision-making problems that traces back to the 1950s and early 1960s. A number of the seminal works at this time that set the foundations for dynamic programming include publications by Bellman (1954), Bellman (1957), Howard (1960), and Bellman and Dreyfus (1962). Since then, the dynamic programming field has grown to include newer techniques such as ADP that address the inherent difficulties with using traditional dynamic programming solution approaches and the complexities of real-world problem structures. Unfortunately, as pointed out by Powell (2009), the various sub-communities working to advance dynamic programming concepts use different vernacular and notional symbols to express essentially the same fundamental ideas. For further discussion on relationships between ADP and artificial intelligence, see for example Powell (2010), Tsitsiklis (2010), and Gosavi (2009).

As a point of comparison, there are a number of publications in the field of ADP that address resource allocation problems. Dell'Olmo & Lulli (2004) leveraged dynamic programming as part of their model that examines an application for managing the resources of a transshipment container terminal. The Aerospace Corporation provided a project selection model called SWORD that leverages a combination of optimization approaches including dynamic programming (Crawford et al., 2003). Another DoD ADP application model was presented by Davis, Robbins, and Lunday (2017). In this research effort, ADP was used as the primary analytical tool for managing missile defense interceptor fire controls. Powell, Shapiro, and Simão (2002) used ADP as an alternative to a relaxed linear programming approach for handling the large-scale problem of assigning drivers to trucks for a freight transportation application. Lastly, Powell et al. (2012) offered an ADP resource allocation model for energy resource management. Although each of these examples is outside the scope of examining use-or-lose budgets, they each offer a perspective on how ADP was used to examine a resource allocation problem. Furthermore, to the best of our knowledge there are currently no publications within the ADP literature that have looked at an application for analyzing and tracking use-or-lose budgetary systems.

THIS PAGE LEFT INTENTIONALLY BLANK

3.0 Overview of DoD Financial Execution

A program office acquisition environment is interwoven with a number of important schedules and critical timelines. The more prominent time-oriented processes that a program manager (PM) must adhere to include: (1) a schedule for budget preparation, review, submission, and approval; (2) the timeline for prime contract awards or modifications, which can include periods for request for proposals (RFPs), time for proposal preparations and responses to proposal questions, review and assessment of submitted proposals, and time for resolving a possible bid protest after a contract award is announced; (3) the fiscal year calendar that involves mid-year financial reviews, end-of-year closeout reviews, and even possible monthly spending benchmark reviews; (4) programmatic schedules with well-defined milestone review thresholds. Unfortunately, these separate process schedules do not always complement one another or align cohesively in a streamlined method that facilitates the delivery of a weapon system platform.

"It's tough to manage an event-driven program in a schedule-driven budget."

- William T. Cooley (Cooley & Ruhm, 2014)

The FY calendar includes important start dates (October 1) and stop dates (September 30) that are necessary for comptrollers and budgetary personnel to track and manage funding that supports the acquisition of a weapon system. However, the fact that the fiscal year calendar starts on October 1 and ends on September 30 has little to do with timing for parts, materials, test events, and other programmatic activities necessary for fielding a weapon system. Nonetheless, the reality is that these dates have considerable influence on when funding is available and the timing of financial commitment actions or cash allocation decisions a program office is likely to take. In the remainder of this section, we take a closer look at different aspects of the DoD financial execution environment: stages of a transaction, appropriation categories, and spending timelines and benchmarks.

3.1 Stages of a Transaction

Once a cash determination is made to allocate money for a particular project, the transaction moves through formal DoD financial execution stages. The flowchart in Figure 1 from the Army's financial management operations field manual provides the order of execution stages. This financial execution process is the standard used throughout the DoD. The first step is the authorization of a funding transaction. After the appropriate authorization documentation is completed and signed, the funding is said to be committed. Committing dollars is an important first step in the execution process that occurs prior to the actual movement of money to a recipient. This initial stage serves as a cross-check that helps to avoid antideficiency violations that result when funding is issued to a contractor or service provider in excess of what is available. Committed dollars are then used to prepare formal and legal contractual obligations between the weapon system program office and a hired vendor. The obligation creates a legal reservation of funds and represents the allocated funds that are available for paying for a project. As work is performed on the project, expenses are accrued. A vendor then provides invoices to the program office for which payment is issued. Once payment is received by the vendor or contractor, the funding is considered disbursed. The terms *outlays* and *expenditures* can also be used to refer to disbursed funding. Throughout the course of a fiscal year, the financial execution status of a weapon system program office is routinely tracked and assessed. The basis of measurement used to evaluate fiscal year execution is the amount of overall budget that currently resides in each of these respective stages. However, significant attention is paid particularly to the obligation and expenditure positions of a weapon system program. To highlight the magnitude of the amount of funding that moves through this process each year, the Defense Finance and Accounting Service (DFAS) reported for FY2017 that it paid out $554 billion in disbursements and for FY2018 that it paid out $558 billion in disbursements.

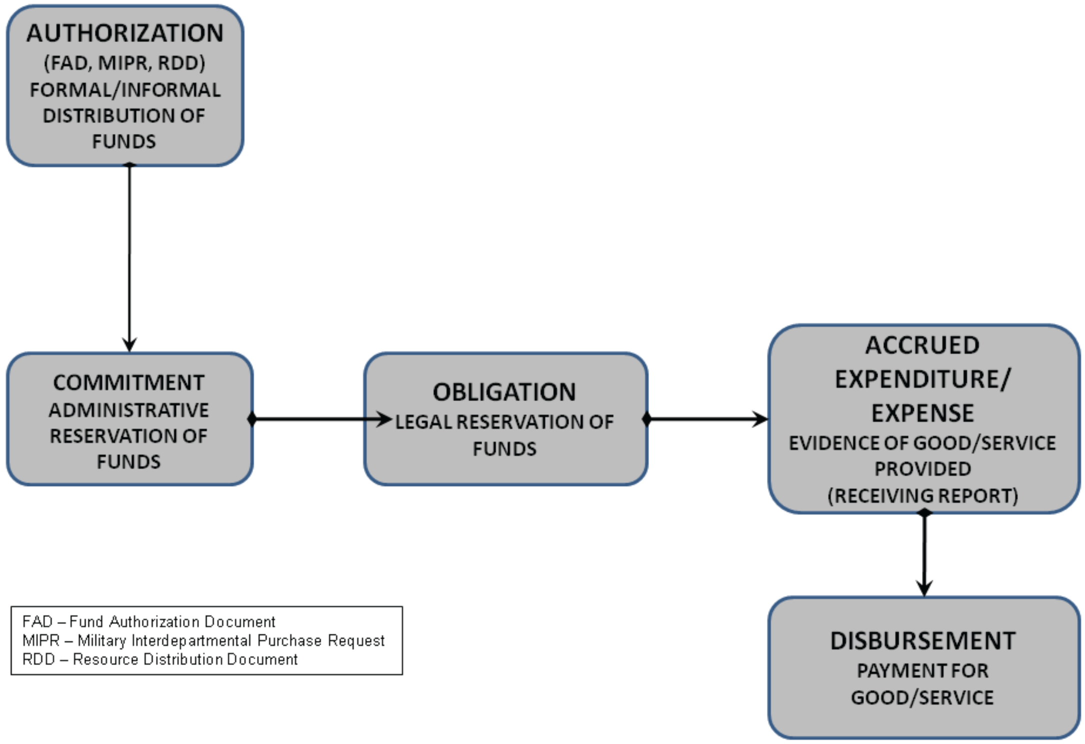

Figure 1: Stages of Transaction

3.2 Appropriation Categories

An additional factor that contributes to the complexity of financial execution at the DoD is the agency's use of different appropriation categories. When creating a budget for a weapon system program office, similar types of projects or work are categorized together in the same appropriation category. Furthermore, the activities of the separate appropriation categories are funded with unique types of money, or with what is more commonly referred to as different "colors"-of-money. These categorizations of activities and funding allow regulators, comptrollers, and other oversight officials to have better insight on how money is spent and on what activities constitute most of the defense budget. However, weapon system program managers and their financial staff are now encumbered with the additional responsibility of managing their programs to correct appropriation categories and must account for these delineations when making decisions related to budget preparations, funding requests, and cash allocations. The following is a short summary of the more common appropriations:

- Military Personnel (MILPERS): Funds salary and benefits of military personnel, including active duty, reserve, and DoD government civilian employees.
- Research, Development, Test, and Evaluation (RDT&E): Funds projects and initiatives that support program research, technology development, engineering development, manufacturing development, and programmatic test events.
- Procurement: Funds the purchase of military equipment and weapon systems, including the production and fielding costs associated with the assets.
- Operation and Maintenance (O&M): Funds activities directly related to the operations, servicing, and upkeep of fielding military systems and platforms.
- Military Construction (MILCON): Funds construction projects related to buildings, facilities, and property improvement efforts that directly support the operations and maintenance of a fielded weapon system.

3.3 Spending Timelines and Benchmarks

Each of the DoD's appropriation categories are subject to guidance regarding the amount of time allowable for moving money through the different stages of a transaction described in Section 2.1. Particular attention is paid to the rate at which funding is obligated and disbursed. Within DoD financial execution, regardless of the appropriation category, money exists in two possible periods: (1) the current period and (2) the expired period. Weapon system program offices must ensure all new obligation actions occur during the current period. The length of the current period is different for each "color"-of-money or appropriation category. O&M and MILPERS have the shortest current period at one year, RDT&E funding has a two-year current period timeframe, the current period for procurement funding can range between three to five years, and military construction has the longest current period at five years. Once the current period for an appropriation has lapsed, the funding moves into an expired period. Regardless of the appropriation, the expired period lasts for five years once the current period is over. During the expired period, no new obligations are allowed. However, funds that were already obligated during the current period can be expensed and recorded as an outlay. Once the expired period has lapsed, the funding is considered canceled and can no longer be used for obligations or expenditures.

The current period and expired period set strict cash flow stopping points; however, the cash flow performance of a weapon system program office is judged on a continual basis. If for any reason it appears that a program office is falling too far behind in its ability to effectively issue and spend money, it runs the risk of being perceived as having too large of a budget for its mission. Comptroller officials and leadership at a more senior level to the program office have the authority to reallocate funding from underperforming program offices to other program offices or activities. Thus, there is an imperative for program offices to maintain constant vigilance of their financial execution position and to make quality cash allocations to contracts and vendors that will expeditiously accrue and expense their funding allotments.

From the perspective of purely protecting funds in a use-or-lose environment, the sooner money moves through the complete stages of a transaction, the better it is for the program office. Unfortunately, programmatic activities and acquisition initiatives that require funding are not always conveniently timed or necessarily ready to receive funds in a manner that allows program offices to keep pace with the spending benchmarks in Figure 2. Furthermore, if a program office expends funding too quickly, it runs the risk of running over its budget before the fiscal year is over. Much like underutilizing funds, overrunning a budget is another financial execution position that a program office needs to avoid and must take into consideration when making cash allocation determinations.

	Month	RDT&E		Procurement		O&M		MILCON	
		Obl.	Exp.	Obl.	Exp.	Obl.	Exp.	Obl.	Exp.
First Year of Availablity	Oct	7.5%	4.6%	6.7%	N/A	8.3%	6.3%	5.4%	1.2%
	Nov	15.0%	9.2%	13.3%	N/A	16.7%	12.5%	10.8%	2.3%
	Dec	22.5%	13.8%	20.0%	N/A	25.0%	18.8%	16.3%	3.5%
	Jan	30.0%	18.3%	26.7%	N/A	33.3%	25.0%	21.7%	4.7%
	Feb	37.5%	22.9%	33.3%	N/A	41.7%	31.3%	27.1%	5.8%
	Mar	45.0%	27.5%	40.0%	N/A	50.0%	37.5%	32.5%	7.0%
	Apr	52.5%	32.1%	46.7%	N/A	58.3%	43.8%	37.9%	8.2%
	May	60.0%	36.7%	53.3%	N/A	66.7%	50.0%	43.3%	9.3%
	Jun	67.5%	41.3%	60.0%	N/A	75.0%	56.3%	48.8%	10.5%
	Jul	75.0%	45.8%	66.7%	N/A	83.3%	62.5%	54.2%	11.7%
	Aug	82.5%	50.4%	73.3%	N/A	91.7%	68.8%	59.6%	12.8%
	Sep	90.0%	55.0%	80.0%	N/A	100.0%	75.0%	65.0%	14.0%
Second Year of Availablity	Oct	90.8%	57.9%	80.8%	N/A	100.0%	77.1%	67.1%	18.1%
	Nov	91.7%	60.8%	81.7%	N/A	100.0%	79.2%	69.2%	22.2%
	Dec	92.5%	63.8%	82.5%	N/A	100.0%	81.3%	71.3%	26.3%
	Jan	93.3%	66.7%	83.3%	N/A	100.0%	83.3%	73.3%	30.3%
	Feb	94.2%	69.6%	84.2%	N/A	100.0%	85.4%	75.4%	34.4%
	Mar	95.0%	72.5%	85.0%	N/A	100.0%	87.5%	77.5%	38.5%
	Apr	95.8%	75.4%	85.8%	N/A	100.0%	89.6%	79.6%	42.6%
	May	96.7%	78.3%	86.7%	N/A	100.0%	91.7%	81.7%	46.7%
	Jun	97.5%	81.3%	87.5%	N/A	100.0%	93.8%	83.8%	50.8%
	Jul	98.3%	84.2%	88.3%	N/A	100.0%	95.8%	85.8%	54.8%
	Aug	99.2%	87.1%	89.2%	N/A	100.0%	97.9%	87.9%	58.9%
	Sep	100.0%	90.0%	90.0%	N/A	100.0%	100.0%	90.0%	63.0%
Third Year of Availablity	Oct	100.0%	90.8%	90.8%	N/A	100.0%	100.0%	90.4%	65.5%
	Nov	100.0%	91.7%	91.7%	N/A	100.0%	100.0%	90.8%	68.1%
	Dec	100.0%	92.5%	92.5%	N/A	100.0%	100.0%	91.3%	70.6%
	Jan	100.0%	93.3%	93.3%	N/A	100.0%	100.0%	91.7%	73.2%
	Feb	100.0%	94.2%	94.2%	N/A	100.0%	100.0%	92.1%	75.7%
	Mar	100.0%	95.0%	95.0%	N/A	100.0%	100.0%	92.5%	78.3%
	Apr	100.0%	95.8%	95.8%	N/A	100.0%	100.0%	92.9%	80.8%
	May	100.0%	96.7%	96.7%	N/A	100.0%	100.0%	93.3%	83.3%
	Jun	100.0%	97.5%	97.5%	N/A	100.0%	100.0%	93.8%	85.9%
	Jul	100.0%	98.3%	98.3%	N/A	100.0%	100.0%	94.2%	88.4%
	Aug	100.0%	99.2%	99.2%	N/A	100.0%	100.0%	94.6%	91.0%
	Sep	100.0%	100.0%	100.0%	N/A	100.0%	100.0%	95.0%	93.5%

Figure 2: DoD Spending Guidance by Appropriation

Figure 2 provides DoD spending guidance that serves to assist program offices with determining whether their cash flow performance is maintaining an adequate pace. A close examination of the information in Figure 2 reinforces the concept that there are different benchmark spending expectations for the different “colors”-of-money. Not shown on the chart is MILPERS. Since this appropriation is primarily for salaries, its expenditure cycle occurs at a relatively predictable and standard pace. Also, procurement funding does not show a monthly expenditure rate. Since procurement is used to buy and support the purchase of large weapon systems and platform end items, its expenditures often occur in single large sums, as opposed to small monthly incremental allotments. However, the remaining three appropriations—RDT&E, O&M,

and MILCON—represent initiatives that a program office could fund and receive outlays against in relatively smaller installment amounts to projects. Figure 2 reveals that after the first year of availability, the expectation is that RDT&E funds will be 55% expended, O&M funds will be 75% expended, and MILCON funding will be 14% expended. It is these appropriations that are of interest for use in an ADP approach for financial execution management. ADP is ideal for either appropriation categories or specific projects where a program office would consider issuing staggered multiple allotments of cash or commitment actions to pay for the activity. This cash allocation approach is one where the program office is attempting to determine whether the contractor or vendor will spend the current funds allotted to it before another installment of money is provided.

THIS PAGE LEFT INTENTIONALLY BLANK

4.0 A Financial Execution Management Model

The following section provides a mathematical formulation for the financial execution problem of weapon system program offices. We define critical variables of the financial execution system and adopt them to a dynamic programming formulation.

At the start of the fiscal year, a budget of bud_i is allocated to each of a finite number $I$of projects $i = 1, \dots, I$. During each of a finite number of time periods $t = 1, \dots, T$, each project i has a (random) disbursement need $\widehat{D}_{i,t}$, which must be satisfied from the current "inventory" of funds that have been committed and have become available to project i by period t.

The agency's objective is to allocate funds in a way that tracks the actual disbursements as closely as possible. This is reflected in the model as follows. For $t = 1, \dots, T$, let $b_{i,t}^{c}$ denote the total amount committed to project i by the end of period t. In particular,

$$b_{i,t}^{c} = \sum_{s=1}^{t} x_{i,s} \qquad (1)$$

where $b_{i,s}^{c} = 0$ for $s \leq 0$. Moreover, we assume that at the start of each period, the agency has a cumulative disbursement schedule $\bar{b}_{i,t}^{d} = [\bar{b}_{i,t}^{d}(1), \dots, \bar{b}_{i,t}^{d}(T)]$ for each project i, where $\bar{b}_{i,t}^{d}(n)$ denotes the current (i.e., at the end of period *t*) projected amount of money that project i will need during time n. Once the actual disbursement requirement $\widehat{D}_{i,t}$ for project i during period t is revealed, the disbursements for each project i are updated according to a given function F^{d}, so that

$$\left(\bar{b}_{1,t+1}^{d}, \dots, \bar{b}_{I,t+1}^{d}\right) = F^{d}\left[\left(\bar{b}_{1,t}^{d}, \dots, \bar{b}_{I,t}^{d}\right), \left(\widehat{D}_{1,t}, \dots, \widehat{D}_{I,t}\right)\right]. \qquad (2)$$

At the start of each period $t = 1, \dots T$, and for each project i, the agency must decide on a total amount x_t to commit. This amount is allocated to the I projects based on fixed allocation rules and is subject to constraints that depend on the cumulative commitments $b^c_{i,t}$ and current disbursement schedule $\bar{b}^d_{i,t}$ for each project i. Given $b^c_{i,t}, \ldots, b^c_{I,t}$ and $\bar{b}^d_{1,t}, \ldots, \bar{b}^d_{I,t}$, let

$$\chi(b^c_{i,t}, \ldots, b^c_{I,t}, \bar{b}^d_{1,t}, \ldots, \bar{b}^d_{I,t}) \quad (3)$$

denote the corresponding set of feasible total commitment amounts x_t. If the agency elects to commit x_t, the cumulative commitments for each project i are updated according to a given function F^c(describing a given allocation rule), so that

$$\left(b^c_{1,t+1}, \ldots, b^c_{I,t+1}\right) = F^c\left[\left(b^c_{1,t}, \ldots, b^c_{I,t}\right), x_t\right]. \quad (4)$$

If the agency commits x_t at time t, its associated "cost" for that time period is the absolute difference between the cumulative amount committed by the end of time t, and the cumulative projected disbursements by the end of time $t + \alpha_i$ (which is when x_t first becomes available for disbursement), that is,

$$\left|\sum_{i=1}^{I} b^c_{i,t-1} + x_t - \sum_{i=I}^{I} \bar{b}^d_{i,t}(t + \alpha_i)\right|. \quad (5)$$

The term α_i is a project specific sensitivity parameter. The choice α_i reflects the number of time periods beyond the current time period t that a program office wants to provide an incremental amount of funding that will sufficiently cover project i costs occurring between time periods t and $t + \alpha_i$.

4.1 Formulation as a Dynamic Program

To formulate the agency's sequential decision problem as a dynamic program, we need to specify the *state variables*, the *decision variables,* the *exogenous information processes, transition function*, and the *objective function.*

State Variables: For $t = 1, \ldots, T$, the *state* S_t at the start of period t is a pair that includes, for each project $i \in \{1, \ldots, I\}$, the values $b^c_{i,t-1}$ (i.e., the cumulative commitment to project i by the end of time t-1) and $\bar{b}^d_{i,t-1}$ (i.e., the projected disbursement schedule for project i as of the end of period $t - 1$), that is,

$$S_t = \left[\left(b^c_{1,t-1}, \ldots, b^c_{I,t-1}\right), \left(\bar{b}^d_{1,t-1}, \ldots, \bar{b}^d_{I,t-1}\right)\right]. \quad (6)$$

Decision Variables: For $t = 1, \dots, T$ and $i = 1, \dots, I$ the *decision variable* x_t denotes the amount that the agency commits at the start of time t. If the state at the start of period t is S_t, then x_t is constrained to satisfy

$$x_t \in A(S_t) \coloneqq \chi\left(b^c_{1,t}, \dots, b^c_{I,t}, \bar{b}^d_{1,t}, \dots, \bar{b}^d_{I,t-1}\right). \tag{7}$$

Exogenous Information Process: There is a single exogenous information process $\{\widehat{D}_{i,t}\}_{t=1}^{T}$

associated with each project i, where each $\widehat{D}_{i,t}$ are simulated actual disbursement requirements for each project i during period t.

Transition Function: Suppose that at the start of period t, the state is S_t. If the decision $x_t = (x_{1,t}, \dots, x_{I,t})$ is made, and the exogenous information for that period is $\widehat{D}_t = (\widehat{D}_{1,t}, \dots, \widehat{D}_{I,t})$, then the state at the start of period $t+1$ is

$$\begin{aligned} S_{t+1} &= S^M(S_t, x_t, \widehat{D}_t) \\ &= \left[\left(b^c_{1,t}, \dots, b^c_{I,t}\right)\left(\bar{b}^d_{1,t}, \dots, \bar{b}^d_{I,t-1}\right)\right] \\ &= \left[F^c\left(\left(b^c_{1,t-1}, \dots, b^c_{I,t-1}\right), x_t\right), F^d\left(\left(\bar{b}^d_{1,t-1}, \dots, \bar{b}^d_{I,t-1}\right), \left(\widehat{D}_{1,t}, \dots, \widehat{D}_{I,t}\right)\right)\right]. \end{aligned} \tag{8}$$

where F^c and F^d are defined by the total amount committed calculation and disbursement schedule updates provided in Equation 1 and Equation 2 respectively. Figure 3 depicts the relationship that exists between the *state variables* S_t, *decision variables* x_t, and *exogenous information process* $\widehat{D}_t$. At the beginning of a time period *t*, the financial execution status of a program office is captured by S_t which includes the cumulative commitment amounts and project disbursement schedules for each project i. At this point, exogenous information $\widehat{D}_t$ regarding the previous time period's disbursements is revealed. The decision process utilizes information from the state position S_t and exogenous information $\widehat{D}_t$ to select a commitment action x_t regarding the amount of additional incremental funding to allocate to each project i. This commitment action x_t along with our knowledge regarding the current actual project disbursement amounts $\widehat{D}_t$, allows our decision system to step forward one time period and into the next state position S_{t+1}, which contains updated information regarding our program office's cumulative commitment amounts and project disbursement schedules.

The process continues forward for a predefined limited number T of time periods or decision periods.

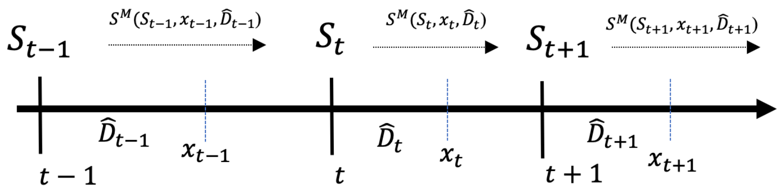

Figure 3: State-to-State Transitions

Objective Function: Suppose that at the start of period t, the state is S_t and the decision x_t is made. Then the corresponding *contribution* of period t is

$$\hat{C}(S_t, x_t) := -\left|\sum_{i=1}^{I} b_{i,t-1}^{c} + x_t - \sum_{i=1}^{I} \bar{b}_{i,t-1}^{d}(t+\alpha)\right|. \tag{9}$$

The objective is to find a policy that maximizes the expected total contribution over the T periods, that is, a policy that maximizes

$$\mathbb{E}\left\{\sum_{t=1}^{T} \hat{C}(S_t, x_t) | S_0\right\}. \tag{10}$$

4.2 Cash Allocation Example

We now consider the simple case of allocating funding for a single project with a total project budget $bud_1 = \$24$ million. We define the time period t as a month and consider the cash allocation process for this single project over a fiscal year horizon $T = 12$ months. The choice of t reflects the frequency of how often a program office wants to assess its financial execution status and make an allotment of funding decision x_t across all the projects within its budget. Additionally, we'll select $\alpha = 2$, to indicate that the program office wants to consider funding allotments in amounts that cover three-month timeframes. When the decision system reaches a month whereby there are fewer than three months remaining in the fiscal year, at $t = 11$ or $t = 12$, we can adjust the future allotment consideration for a two-month or single-month time frame respectively and update the α term to either $\alpha = 1$ or $\alpha = 0$. An initial cumulative disbursement schedule $b_{i,t}^{d}$ is created from either a direct vendor quote, similar work completed in the past, or from any other viable technique available to the program office that can be used to create an initial spend plan forecast. For our single project, we'll assume the following cumulative disbursement schedule in millions of dollars:

$$\bar{b}_{1,1}^{d} = [0,0,0,2,4,6,10,13,16,19,22,24].$$

This disbursement profile represents a project that starts work in the fourth month of the fiscal year, January, and is forecasting a demand of cash for $2 million per month for months $t = 4, 5, \text{and } 6,$ a demand of $4 million per month for month $t = 7,$ a demand of $3 million per month for months $t = 8, 9,$ 10, and 11, and last, a demand of $2 million for the final month $t = 12$.

Let's consider a case where the decision system arrives at time period $t = 4$, January, with $S_t = (2, [0,0,0,2,4,6,10,13,16,19,22,24])$. At this point, $2 million are committed to the project and $0 million are disbursed. The decision system makes a commitment action according to Equation 3. Given that $\alpha = 2$, the next allocation of funding will attempt to bring the current total committed funding level $b_{1,4}^{c}$ up to a level that matches as closely as possible the estimated cumulative disbursement amount for March (time period $t + 2$). In our example, we'll assume that the choice for the next allotment of funding is $4 million. The decision system moves into the next time period, $t = 5$, February. At this point, exogenous information is revealed regarding actual disbursements that occurred in time period $t = 4$. This information is then used to create an updated cumulative disbursement schedule. For example, if the actual disbursement amount in January was only $1 million as opposed to the anticipated $2 million that was expected, an updated disbursement schedule might look like the following

$$\bar{b}_{1,5}^{d} = [0,0,0,1,3,6,10,13,16,19,22,24].$$

The implication is that the contractor supporting the work fell behind schedule during the month of January; however, the updated cumulative disbursement schedule indicates a belief that the contractor will be able to make up the additional work over the next two time periods and still require the full $24 million total budget to pay for the project prior to the end of the 12-month period.

4.3 Curse of Dimensionality—Single Project Case

One drawback of using the dynamic programming formulation for solving the financial execution problem is that it suffers from the "curse of dimensionality," which is a common issue for many optimization modeling approaches. Using the single project scenario described in Section 4.2, we can consider the computational demands of our decision system based on the size of the action space x_t and state-space S_t. In order to determine these dimensions, we'll first need to make an assumption about the discretized amount with which our project receives and disburses dollars. For simplicity, we assume money is received and spent to the nearest $1 million increment. However, examples of other viable increments include $5 million, $2 million, $0.5 million, $0.25 million, $0.1 million, and so on. Additionally, we need to make another assumption about the range of variability that can occur with our simulated exogenous data $\widehat{D}_{1,t}$. In this case, we'll assume that disbursements can occur with variability of +$2 million to -$2 million, above and below the forecasted amount for a given time period t. Given these parameters, we can now calculate both the sizes of the action-space and state-space.

Given that the project receives money to the nearest $1 million increments, this means that for each time period t, there are 25 possible commitment or de-commitment actions to our $24 million project. De-commitment actions are allowed as long as sufficient funding remains committed to the project to cover all expenses (disbursements) that have occurred to date. The state-space is defined as the combination of our cumulative commitment amount $b_{1,t}^{c}$ and disbursement schedule $\bar{b}_{1,t}^{d}$. For the $24 million project, there are 25 possible values for the scalar $b_{1,t}^{c}$. Furthermore, since we are anticipating disbursements to occur in nine months out of our 12-month timeframe, there are 5^9 possible vectors combinations for $\bar{b}_{1,t}^{d}$, and when combined with the 25 possible values of $b_{1,t}^{c}$ means that there are nearly 50 million state-space possibilities. Even for this single project situation, to model all possible outcomes for all the possible state-action pairings is computationally intractable. This difficulty is further exacerbated when we consider budget scenarios that examine multiple projects simultaneously.

As an alternative, we consider using an approximate dynamic programming (ADP) modeling approach to the financial execution problem. ADP allows us to estimate a “good” decision-making solution without having to explicitly enumerate and calculate the values of all possible action-outcome pairings. Rather, it provides a means of approximating state-space values through the use of Bellman’s formula. The expectation form of Bellman’s equation is

$$V_t(S_t) = \max_{x_t}\left(\hat{C}(S_t, x_t) + \gamma\mathbb{E}\{V_{t+1}(S_{t+1})|S_t\}\right). \tag{11}$$

Bellman’s formulation contains two components. It retains the *contribution* from the previously stated objective function, $\hat{C}(S_t, x_t)$, and combines with it a discounted expected value of the state the decision system arrives at as a result of the action x_t taken at time period t. Through the use of simulation, the ADP approach allows us to approximate or “learn” the values of state-spaces in our decision system. As a result, the ADP algorithm can generate a cash allocation policy that directs a program office to allocate funding during each time period t to successively move the decision-maker from one high valued state-space (financial execution position) to another high valued state-space position. Therefore, the cash allocation policy generated by the ADP algorithm will balance between allocation decisions taken earlier in the FY with those generated later, creating a sequential cash allocation policy that limits that amount of over-committed funding without shortchanging the projects in the future.

4.4 Curse of Dimensionality—Multiple Project Case

Unfortunately, both state-space and action space challenges increase as additional projects are introduced into the system. When more projects are added, we now have to consider generating optimal cash allocation policies across two or more projects simultaneously. Consider the following initial cumulative disbursement schedule for two projects stated in millions of dollars ($ million):

$$\bar{b}^d_{1,1} = [0,0,0,0,0,3,7,11,14,14,14,14]$$
$$\bar{b}^d_{2,1} = [0,0,0,0,0,0,2,5,8,10,10,10]$$

In this case, we have two projects in our budgetary system. Again, the total available budget that needs to be executed over the 12-month period is $24 million. However, the total budgetary amount is divided between two projects, where the budget for project one is $bud_1 = \$14$ million and the budget for project two is $bud_2 = \$10$ million. Additionally, the month-to-month forecasted spending schedule is different for both projects. The planned start month for project one is March, $t = 6$, and the completion date is at the end of June, $t = 9$. For the second project, the planned start month is April, $t = 7,$ and the completion date is planned for July, $t = 10.$ For simplicity, we'll continue to assume that projects will receive and spend money in $1 million discretized increments and that the range of variability that can occur with our simulated exogenous data [$\widehat{D}_{1,t}$ $\widehat{D}_{2,t}$] is +$2 million to -$2 million, above and below the forecasted amount for a given time period t for both projects. Again, we have large state-space considerations for both projects. For project 1, we are anticipating a demand for funds across four time periods and that any time period can assume one of five possible values. Also, given that project 1 has 15 possible commitment values for the scalar term $b_{1,t}^{c}$, this means that there are 9,375 possible state-space values. Project 2 also has five possible values for funding needs across four time periods, and since there are only 11 possible values for the scalar term $b_{2,t}^{c}$, this means that there are a total of 6,875 possible state-space values for project 2. When combined together, this equates to over 64 million possible state-space combinations.

As multiple projects are added to the decision system, the action-spaces takes on an additional computational complexity problem. Although project 1 has 15 possible commitment actions and project 2 has 11 possible commitment actions, the decision system in aggregate when the two projects are combined has only 25 possible cash commitment actions. The challenge is determining for any given time period t the proper disbursement of funds between the two projects given a system-level commitment decision. For example, if a program office chooses to commit zero dollars ($0.0 million) during any given time period, then the allocation between the two projects is trivial and each project receives zero funding ($0.0 million). However, if the system level action is to commit $1.0 million dollars, then there are two possible ways to distribute this down to the two projects: (i) either project 1 receives $1 million and project 2 receives $0.0 million or (ii) project 1 receives $0.0 million and project 2 receives $1 million. In this situation, the question becomes determining which of the two projects should receive the $1 million. Furthermore, the problem becomes more complicated as the system-level commitment action under consideration increases. For example, if the system-level commitment action is increased to $2 million, then there are now three possible combinations for allocating this funding across the two projects: (i) project 1 receives $2 million and project 2 receives $0.0 million, (ii) project 1 receives $0.0 million and project 2 receives $2 million, and (iii) project 1 receives $1 million and project 2 receives $1 million. In order to determine the appropriate allocation for a given system-level commitment action, we define a fairness property that establishes the rules governing the distribution of system-level commitment funds to the individual projects.

4.5 The Resource Allocation Fairness Property

In this section, we describe a viable fairness property rule to determine how an agency level commitment action x_t is allocated to create the n-tuple of individual commitment actions $(x_{1,t}, \ldots, x_{I,t})$. A weapon system program office or agency may choose an alternative allocation process; however, the following implementation offers one approach to defining the function F^c referenced in Equation 4.

In an incremental funding environment where a weapon system program office has limited access to its total obligation authority budget level throughout the year, the fairness property dictates that priority for funding goes to projects with the largest and most immediate demand. However, smaller projects with an immediate demand will also receive primacy. The idea is to prevent a situation whereby for a given time period t, when there is a limited system-level commitment action amount x_t, that this funding is allocated to a project i that does not have a positive forecasted demand for funding until several time periods into the future, leaving no funding or very little funding available to provide to a project that does possess a positive forecasted cash demand within the immediate time periods. In this regard, the fairness property considers the magnitude of a time period's disbursement demand $\| \bar{b}^d_{i,t}(n) \|$ as well as the number of time periods n that the demand is beyond the current time period t.

We consider an example using the data from our two-project case. Let's assume that we arrive at time period $t = 6$ and using Equation 6 define the system state position containing the financial execution position of the two projects as follows:

$$S_t = [(6,1), (0,0)]$$

We interpret our system state S_t as having \$6 million of funding committed to project 1 and \$1 million committed to project 2 as of the end of February, $t - 1$. At this point in time neither project has started their effort, therefore both projects have zero dollars, \$0.0 million, currently disbursed. The decision system has just arrived in March, time period t, and must select a commitment action x_t. Since spending has not commenced on either project, no new information regarding actual disbursements has been introduced into the system and the initial forecasted disbursement schedule for both projects remains valid.

$$\bar{b}_{1,1}^{d} = [0,0,0,0,0,3,7,11,14,14,14,14]$$
$$\bar{b}_{2,1}^{d} = [0,0,0,0,0,0,2,5,8,10,10,10]$$

Given that \$6 million is currently committed to project 1, without any further funding project 1 is predicted to run short of cash starting in time period $t = 7$. The unfulfilled demand for funding in time period $t = 7$ is \$1 million. Furthermore, in the next time period, $t = 8$, the unfulfilled demand increases by \$4 million and in the next time period, $t = 9$, it increases by an additional \$3 million. At the end of February and for the start of March, project 2's current commitment level is \$1 million. Without any additional funding, project 2 will fall short of its demand for cash starting in time period $t = 7$ by \$1 million. For time period $t = 8$, the forecasted demand increases by \$3 million. In time period $t = 9$, it increases by an additional \$3 million and for $t = 10$, the demand increase is another \$2 million.

The fairness property dictates that the project with the largest and most immediate demand for funding will receive funding until its demand for cash is the same as another project in the same time period. At this point, if there is any unallocated funding remaining, the two projects, which now both have the highest demand for cash, will receive funding equally until their demand is equivalent to a third project's demand for cash. Then, if there is any unallocated funding remaining, the three projects will all receive funding evenly until their demands are equivalent to a fourth project with cash demands in the same time period. Now, all four projects will receive funding evenly. The process continues until all demands for all projects with cash demands in the current time period are met. If there is still any remaining unallocated funding, the fairness property moves one time period forward and with the remaining available cash begins to fulfill the needs of the project with the largest demand for cash in time period $t + 1$. This project's demands for cash in the updated time period are satisfied until the demand is the same as a project with the second highest demand for cash also in time period $t + 1$. Now, both projects' demand for cash is satisfied evenly until it is equivalent with a third project's demand in time period $t + 1$, at which point the three projects will receive funding evenly. The process continues as it did for time period t. If there is still unallocated funding remaining after all demands for all projects in time period $t + 1$ are met, the allocation scheme steps forward one more time period to $t + 2$ and repeats the same process. The fairness allocation process continues to step forward in time fulfilling the cash needs of projects with the highest demand until the full amount of the commitment action x_t is completely distributed across each of the projects i.

Leveraging the fairness property, we can examine how a commitment action x_t = $8 million made in March is allocated between projects 1 and 2. The largest and most immediate demand for cash is $1 million, which is the time period's $t = 7$ remaining cash demand amount for both projects. Since there are sufficient funds to cover both $1 million demands, the allocation scheme starts by giving $1 million to each project and then examining demands of the next time period. In time period $t = 8$, project 1 has the largest demand of $4 million as opposed to project 2 with only a $3 million demand. Therefore, a $1 million allocation is given to project 1. Now, both project 1 and project 2 have the same demand for cash of $3 million. However, there is only $5 million of unallocated funding remaining. Projects 1 and 2 will receive funding evenly at this point, which translates into both projects receiving at least an additional $2 million worth of funding. At this point, there is only $1 million remaining of unallocated funding, yet both projects have the same unmet demand for cash of $1 million remaining for time period $t = 8$. In this situation, there is a tie between projects regarding which one possess the largest demand in the most immediate time period. Furthermore, there is only $1 million remaining of unallocated funding, which is an insufficient amount to satisfy the remaining demands of both projects evenly. In such circumstances, unless there is a rationale for prioritizing one project over another, the final $1 million can be distributed randomly to any of the projects with the remaining highest demand. Clearly, the random choice is only implemented under this special situation at the end of the fairness distribution process when there is a tie among projects with the largest remaining demand, and the nature of how funding allocations are discretized (in this case $1 million increments) prevents these demands being satisfied evenly. Assuming in our scenario that project 1 randomly receives the remaining $1 million increment of funding, we now have the following defined commitment action for the month of March:

$$x_6 = \$8.0M$$

$$\left(x_{1,6}, \dots, x_{I,6}\right) = (\$5.0M, \$3.0M)$$

4.6 Algorithm for the Resource Allocation Fairness Property

The algorithm shown in Figure 4 summarizes the steps of the resource allocation fairness property.

Initialization

Initiate for time period t

$\hat{x}_i = 0 \quad \forall\, i \in I$

$\vartheta = \$1.0M$ # set curly theta equal to the discretization allowance

Select system level action x_t according to Bellman's Equation 12

$x_t = arg\, max_{x_t}\big(\hat{C}(S_t, x_t) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1})|S_0\}\big)$

$commitment$_action = x_t # stores value of the selected commitment action

Fairness property implementation

    PMD = 1 # PMD defined as project with the maximum demand

    for $\theta = t\ to\ T$

        for $i = 2\ to\ I$

            if $\bar{b}^d_{PMD,\theta}(\theta) - x_{PMD,t} + \hat{x}_{PMD} > \bar{b}^d_{i,\theta}(\theta) - x_{i,t} + \hat{x}_i$

            PMD = PMD

        else

            PMD = i

        end if

        end for

        $\hat{x}_{PMD} = \hat{x}_{PMD} + \vartheta$

        $x_t = x_t - \vartheta$

        if $x_t = 0$

            Stop

        end if

    end for

Return system and project commitment actions

    return $(\hat{x}_i) \quad \forall\, i \in I$

    $x_t = commitment\_action$

    $x_t = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_I)$

Figure 4: Fairness Property Algorithm

4.7 An Alternative Integer Programming (IP) Formulation for the Resource Allocation Fairness Property

We can leverage the structure of a binary knapsack optimization model as an alternative implementation of the fairness property described in Section 4.5. Consider the following knapsack problem formulation.

CAKP
$$Max\ z\ =\ -\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}v_{ijk} \tag{12}$$
$$such\ that\ \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} v_{ijk} = \beta$$
$$v_{ijk} \in \{0,1\}$$

We will refer to the optimization problem provided in Equation 12 as the commitment allocation knapsack problem (CAKP). For Equation 12, we have objective function coefficients c_{ijk}, a resource constraint variable β, and decision variables v_{ijk}. Solving Equation 12 after structuring the coefficient c_{ijk} and β terms appropriately will provide the inputs needed to generate the same n-tuple project level cash allocations $x_t = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_I)$ that was derived using the fairness property algorithm described in Figure 4. Once we have the solution values v_{ijk}^* for our CAKP, we can create the project level allocation amounts $\hat{x}_i$ using the following expression,

$$\hat{x}_i = \sum_{j\in J}\sum_{k\in K} v_{ijk} + 2 * b_{i,t}^{a^-}(0) \tag{13}$$

where $b_{i,t}^{a^-}(0)$ represents a de-commitment factor for projects that have the option to decommit some of their currently allocated committed amount $b_{i,t}^{c}$ and have the decommitted funds possibly transferred to another project or projects to pay for shortfalls.

Creating the CAKP objective function coefficients requires the use of the information from the cumulative disbursement schedule $\bar{b}_{i,t}^{d}$ in Section 4.0. By leveraging this information, we can create a fairness allocation schedule $\bar{b}_{i,t}^{a}$ for each project i.

$$\bar{b}_{i,t}^{a} = [b_{i,t}^{a}(0), b_{i,t}^{a}(1), b_{i,t}^{a}(2), \ldots, b_{i,t}^{a}(N-1)] \tag{14}$$

The number of elements in $\bar{b}_{i,t}^{a}$ is determined by the choice of α that was used in the contribution function defined in Equation 9 where $N = \alpha + 3$. The structure of $\bar{b}_{i,t}^{a}$ aligns with the fairness property rules that prioritize funding to the largest project demands for cash in the earliest time period. The variable $b_{i,t}^{a}(0)$ represents project i demands for cash from the previous time period $t-1$. Also, $b_{i,t}^{a}(1)$ represents the demands for cash of project i in the current time period t and $b_{i,t}^{a}(2)$ represents demands for cash in time period $t+1$, that is one month or time period into the future. The sequence continues through the remaining terms in $\bar{b}_{i,t}^{a}$ where the next term continues to represent the successive demands for cash for the next time period. The final term $b_{i,t}^{a}(N-1)$ captures all the remaining demands for cash of project i across all the remaining time periods not yet accounted for up to and including the final time period *T*.

We'll use the following formulation to create the fairness allocation schedule:

$$b_{i,t}^{a}(n) = \begin{cases} 0 & for\ t\ =\ 1 \\ \bar{b}_{i,t}^{d}(n+1) - \sum_{i} b_{i,t-1}^{c} & for\ t\ >\ 1 \end{cases} \tag{15}$$

The term $\sum_{i} b_{i,t-1}^{c}$ can be thought of as the cumulative commitment of funding allocated through time period $t-1$ intended to meet the cumulative disbursement demands $\bar{b}_{i,t}^{d}(t-1)$ that have already occurred.

For $n=0$, it is possible that the allocation schedule term $b_{i,t}^{a}(0)$ is negative. This represents a situation where at time $t-1$ there exists more cumulative funding committed to a project than the total amount disbursed. Therefore, there is the option to decommit funding from project i at time period t and allocate it elsewhere. The counter situation is when $b_{i,t}^{a}(0)$ is positive. In this case, we have a must fund situation where the cumulative commitment level in time period $t-1$ is less than the total disbursements. Therefore, at a minimum a sufficient allocation of funding $\hat{x}_i$ must be allocated to project i in time period t to cover the shortfall. Also available is an alternative strategy to address this shortfall, which is to leverage some of the potential decommit funding and redistribute these dollars to the projects currently running shortfalls. We can isolate those projects that are running a surplus from those that are running a shortfall by creating two variables that are derived from $b_{i,t}^{a}(0)$. That is, we can create

$$b_{i,t}^{a^{-}}(0) = b_{i,t}^{a}(0)\ for\ b_{i,t}^{a}(0) < 0\ , otherwise\ b_{i,t}^{a^{-}}(0) = 0 \tag{16}$$

$$b_{i,t}^{a^+}(0) = b_{i,t}^{a}(0)\ for\ b_{i,t}^{a}(0) \geq 0\ , otherwise\ b_{i,t}^{a^+}(0) = 0 \quad (17)$$

where $b_{i,t}^{a^-}(0)$ represents the amount of possible negative commitments or de-commitment amounts available by project i and $b_{i,t}^{a^+}(0)$ are the shortfall amounts. Now that we have isolated our shortfall projects from our currently overfunded projects, we'll update the fairness allocation schedule $\bar{b}_{i,t}^{a}$ to remove all negative values.

$$\bar{b}_{i,t}^{a} = \left[\left\|b_{i,t}^{a}(0)\right\|,\ b_{i,t}^{a}(1), b_{i,t}^{a}(2), \dots, b_{i,t}^{a}(N-1)\right] \quad (18)$$

Lastly, we'll note the maximum value contained in our updated fairness allocation vector $\bar{b}_{i,t}^{a}$ and refer to it as the *maximum demand increment,* or *mdi.*

$$mdi = \max_{n}\ b_{i,t}^{a}(n) \quad (19)$$

We can now define both the decision variables and the objective function coefficients of the CAKP problem. For our binary variables v_{ijk} and integer coefficients c_{ijk}, we define the subscripts $i, j,$ and k as follows:

$i = 1, 2, \dots, I$	→	project i
$j = 0, \dots, n, \dots, N-1$	→	allocation demand period n in $\bar{b}_{i,t}^{a}$
$k = 1, 2, \dots, K$	→	incremental demand units

The subscript i refers to project i. The subscript j refers to an allocation demand period. For example, j = 2 will refer to the variable $b_{i,t}^{a}(2)$ from the allocation schedule $\bar{b}_{i,t}^{a}$ which is referencing the disbursement demands in the period $t+1$. Lastly, the subscript k is a reference to the demand increment of a given project i within a given demand period j. Furthermore, the largest value in the sequence k is $K = mdi = \max_{n} b_{i,t}^{a}(n)$. To simplify the structure of our CAKP we can reorder the values of j from $j = 0$, ..., n, …, N-1 to $j = 1, 2, \dots, J$ where $J = N$.

We now have the following interpretation for our decision variables v_{ijk} and objective function coefficients c_{ijk}. If a decision variable $v_{ijk} = 1$ this means that a discretized unit (e.g., $1 million) of the resource parameter β will be allocated to satisfy the k^{th} increment of demand from the j^{th} time period of the allocation schedule for project i. The values for the objective function coefficient c_{ijk} are created from a function that leverages the fairness allocation schedule $\bar{b}_{i,t}^{a}$ and the value K as inputs.

$$c_{ijk} = f(\bar{b}_{i,t}^{a}, K) \tag{20}$$

The function shown in Equation 20 will dictate that c_{ijk} will be assigned the lowest values to the largest demand increments in the earliest time periods and systematically decrease these values as the demand increment lessens. The c_{ijk} values will continue to decrease further for demand increments that exist in later time periods of the fairness allocation schedule $\bar{b}_{i,t}^{a}$. The total number of objective function coefficients is equal to $I * J * K$. If a demand increment for a given project in a given time period does not exist, the coefficient will take on a very large positive value M. A coefficient value that is equal to M will push the corresponding decision variable v_{ijk} out of consideration in our CAKP. For example, if $b_{2,t}^{a}(2) = 4$ and $K = 6$ then both the coefficient terms $c_{2,3,5}$ and $c_{2,3,6}$ will not exist and therefore will take on the value M. The coefficient term $c_{2,3,5}$ represents the value associated with the 5^{th} demand increment of project 2 in the 3^{rd} period of the allocation schedule (i.e., one time period into the future). However, since the total demand for cash of project 2 one time period in the future is only $b_{2,t}^{a}(2) = 4$, this means that there can be no demand for a 5^{th} increment of cash by project 2 at this time. Similarly, there can also be no demand for a 6^{th} increment of cash at this time as well. Therefore, we will set both $c_{2,3,5}$ and $c_{2,3,6}$ equal to the large positive value, big M.

The final term to define is β, which represents the resource constraint for our CAKP. The β term can take on any integer value such that $\sum_i b_{i,t}^{a}(0) < \beta \leq \sum_i \sum_n b_{i,t}^{a}(n)$ and is dependent upon the system level commitment action x_t. Since x_t can potentially take on negative values we will normalize our CAKP resource constraint term β such that β is nonnegative. That is,

$$\beta = x_t + \sum_i b_{i,t}^{a}(0) - \left(\sum_i \bar{b}_{i,t}^{d}\,(t-1) \; - \sum_i b_{i,t-1}^{c} \right) \tag{21}$$

If we have a situation where the term $b_{i,t}^{a^-}(0)$ is zero for all values i, then the CAKP resource constraint variable $\beta = x_t$.

Given the definitions of the decision variables v_{ijk}, the coefficient variables c_{ijk}, and the resource constraint variable β, we now have the ability to leverage the CAKP formulation in Equation 12 along with the decision variable transformation in Equation 13 as an alternative mechanism for implementing the fairness resource allocation property. These types of alternative allocation formulation considerations such as the CAKP are important given the potentially large computational expenses involved when scaling the ADP approach to consider actual weapon system program office budgets that may contain hundreds of projects and are managing budgetary dollars that are in the millions.

5.0 "Learning" with ADP

Simulation is the method used for training the financial execution system to "learn" a good cash commitment policy decision. Slightly modifying the structure of Bellman's equation in Equation 11, we can summarize the process for training the financial execution process with the following state-space sampling and update equations.

$$\hat{v}_t^n(S_t^n) = Max_{x_t}[(C(S_t^n, x_t) + \gamma\, \bar{V}_t^{n-1}(S_t^{n,x})] \tag{22}$$

$$\bar{V}_{t-1}^n(S_{t-1}^{x,n}) = (1-\alpha_{n-1})\hat{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) + \alpha_{n-1}(\hat{v}_t^n(S_t^n)) \tag{23}$$

Throughout the ADP "learning" process, the simulation will continually iterate between selecting a sample state-space variable $\hat{v}_n^t(S_t^n)$ from Equation 22 and then using the stochastic smoothing or updated Equation 23 to slowly "train" the previous period's state-space value represented by the variable $\bar{V}_{t-1}^n(S_{t-1}^{x,n})$. For example, we can consider a decision system that has $T = 12$ total time periods where each time period t represents a single month. Assuming that the start of the simulation process is aligned to the beginning of the fiscal year, $t = 1$ represents the month of October. Therefore, when $t = 5$ and $n = 500$, the simulation process is in the middle of its 500th simulation iteration and is currently in the month of February. At the start of each time period t, the model generates a sample state-space value by solving the maximization operand in Equation 22. The objective function in Equation 22 consists of two parts: (a) the one-step contribution $C(S_t^n, x_t)$ defined in Equation 9 and (b) the current value of the discounted state-space position $\bar{V}_t^{n-1}(S_t^{n,x})$ which the simulation process arrives at by selecting the action x_t that solves the objective function in Equation 22. The one-step contribution represents the immediate cost that is incurred to the decision system by taking an action x_t from state S_t. The value $\bar{V}_t^{n-1}(S_t^{n,x})$ represents the currently "trained" and stored value of the state-space $S_t^{n,x}$ at the point when the simulation process has reached its $(n-1)^{st}$ iteration. Before the ADP simulation process begins, the value of each state-space is initialized at $\bar{V}^0(S^{0,x}) = 0$. Each time a state-space is visited during the simulation, the average estimate of its state-space value is continually refined and updated until we reach an acceptable approximation of the estimate of the state-space's true value.

5.1 Step-Size Properties

Equation 23 provides a stochastic smoothing process that is rooted in the algorithm structure of Robbins and Monro (1951). In this equation, the sample state-space value $\hat{v}_n^t(S_t^n)$ calculated in Equation 22 is used to update the stored state-space value $\bar{V}_{t-1}^n(S_{t-1}^{x,n})$ of the state that the decision system was in during the previous time period $t-1$. The successive value of a state has a direct impact on the value of the previous state. It is this recursive structure of the ADP "learning" process that allows the model to capture the long-term impact of taking an action x_t. Once the "learning" and update process is complete, the approximated state-space value $\bar{V}(S_t^x)$ captures not only the value of being in state S_t^x, but the value of all the future states that the decision system can arrive at as the simulation process moves forward in time.

The stochastic smoothing process in Equation 23 requires the use of an alpha-decay or step-size parameter α_{n-1}. Although there are numerous viable and effective choices for alpha-decay, each must satisfy the following three convergence criteria for a given iteration sequence $n = 1, 2, \dots, \infty$:

a) $$\alpha_{n-1} \geq 0$$

b) $$\sum_{n=1}^{\infty} \alpha_{n-1} = \infty$$

c) $$\sum_{n=1}^{\infty} (\alpha_{n-1})^2 < \infty$$

The ADP literature offers a number of different strategies for the choice of alpha-decay. For example, the alpha-decay parameter of $\alpha_{n-1} = \frac{1}{n}$ will satisfy the three criteria above and thus will provide a sequence that converges. However, this step-size is generally not used in practice since its convergence rate is very fast, giving little room to train the decision system's state-space values. As an alternative, Powell (2007) provides a *Polynomial* alpha-decay learning rate and a *Generalized Harmonic* alpha-decay learning rate. Also, Gosavi (2003) and Darken, Chang, and Moody (1992) recommend the use of an *Adapted Deterministic Harmonic* alpha-decay learning rate. Figure 5 provides a graphic for each of these learning rates.

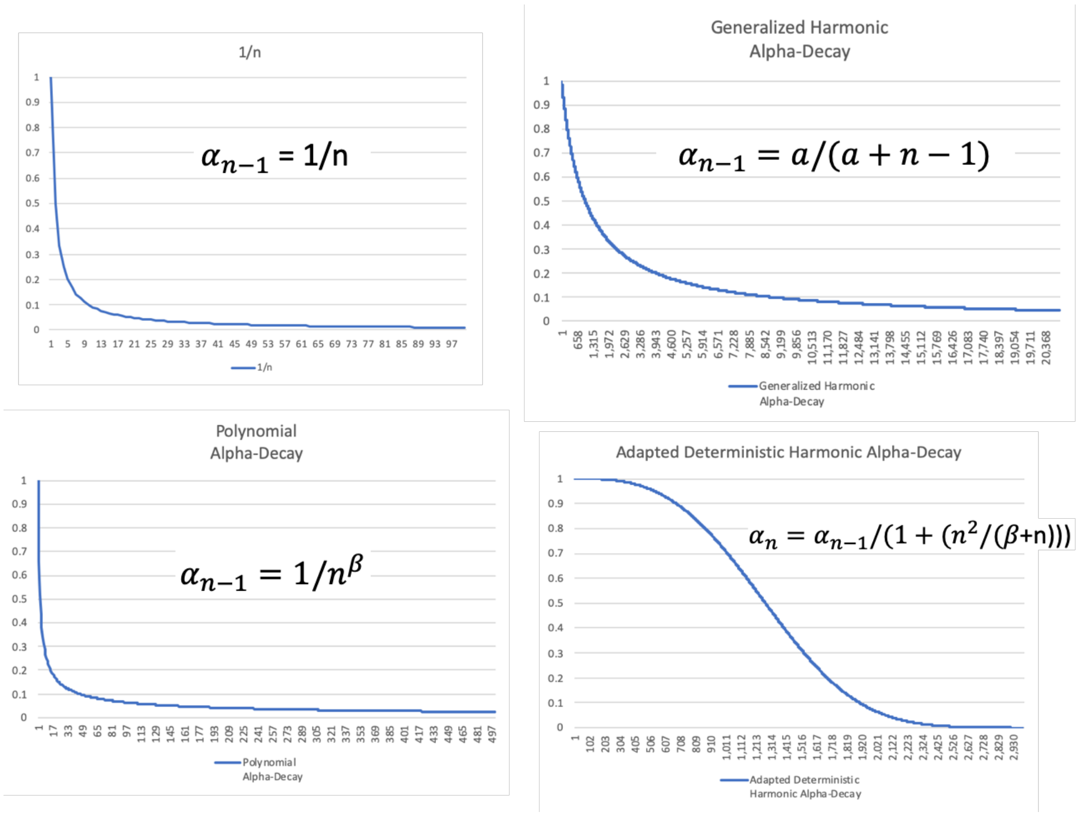

Figure 5: Types of Alpha-Decay (Step-Sizes)

The four figures shown previously show how the alpha-decay process moves from a value of 1 to a value of 0 at various learning rates. During the earlier part of the simulation the alpha-decay value is near 1, meaning that the stochastic smoothing process will emphasize the value of the initial samples of each state-space $\hat{v}_t^n(S_t^n)$. As the learning process progresses, the alpha-decay value decreases and shifts the focus of the stochastic smoothing process from the sampling values of $\hat{v}_t^n(S_t^n)$ to the trained state-space values of $\bar{V}(S_t^x)$.

In considering the four possible learning rates above, both the $1/n$ step-size and the *Polynomial* step-size approaches have relatively fast learning rates. That is, it only takes relatively few iterations for the alpha-decay process to drop from 1 to 0. However, the *Polynomial* alpha-decay property does possess a tuning parameter $\beta \in \left(\frac{1}{2}, 1\right]$ that can be modified to adjust the rate of alpha-decay. The risk of using a faster alpha-decay process is that it can provide the illusion of convergence, when in reality the simulation system hasn't collected enough sample data points $\hat{v}_t^n(S_t^n)$ to provide a good approximation of state-space values. The other two step-size functions, *Generalized Harmonic* and *Adapted Deterministic Harmonic,* possess a much slower rate of convergence. For these two step-sizes, the alpha-decay value remains relatively high before descending to 0. In each of these two cases, the learning rate will spend more time on collecting sampling data on state-space values. However, collecting this additional information involves a higher computational cost. For the examples provided in Figure 5, both the *Generalized Harmonic* and *Adapted Deterministic Harmonic* required over a thousand iterations before the alpha-decay rate expired. Granted, similar to the *Polynomial* alpha-decay rate, the *Adapted Deterministic Harmonic* and *Generalized Harmonic* rates each contain tuning parameters a and β respectively that can be leveraged to speed up or slow down the learning rate. Nonetheless, regardless of the choice of a or β, the *Generalized Harmonic* and the *Adapted Deterministic Harmonic* will still be slower than the $1/n$ and *Polynomial* learning rates.

5.2 Exploration and Exploitation

In addition to determining an appropriate alpha-decay property, another consideration to assess when using ADP is the appropriate mix of simulation iterations dedicated to either exploration or exploitation. One of the problems with using the form of the ADP process provided by Equation 22 and Equation 23 is that due to the minimum operand, the simulation process will always favor the selection of the lowest value state-spaces. As a result, there is a danger that on each iteration of the simulation, the ADP methodology will always select the same or similar actions x_t and repeatedly visit the same states-spaces S_t. This can lead the model to continually train the values of the same states and miss "learning" about alternatively "good" states. In order to prevent the simulation process from cycling on the same state-space values, an exploration technique is implemented that causes the simulation process to randomly select actions as it moves through the decision system. The exploration phase can be expressed by changing the expression in Equation 22 as follows:

$$\hat{v}_t^n(S_t^n) = Random_{x_t}[(C(S_t^n, x_t) + \gamma\, \bar{V}_t^{n-1}(S_t^{n,x})] \quad (24)$$

Equation 24 forces the ADP simulation to select non-optimal actions x_t and to sample the value of states that it would otherwise not visit. The use of an exploration phase requires that a select number of the initial simulation iterations n are dedicated to taking a random sample of state-space values. Once the model has executed its allotment of exploration iterations, the model transitions to an exploitation learning phase as defined by the earlier sampling process in Equation 22 for the remaining $N - n$ iterations of the ADP model. The stochastic update process described in Equation 23 is the same for both the exploitation and exploration phases, only the nature of how the sample-state $\hat{v}_n^t(S_t^n)$ is determined is different.

Figure 6 provides an example of how the iterations can be distributed across exploration and exploitation. In this graphic the simulation is using an *Adapted Deterministic Harmonic* alpha-decay process whereby roughly the first 500 iterations are dedicated to exploration where the states are selected randomly. The remaining 2,000 iterations are then dedicated to exploitation, and the state-space values are now being sampled through the use of the maximization operand.

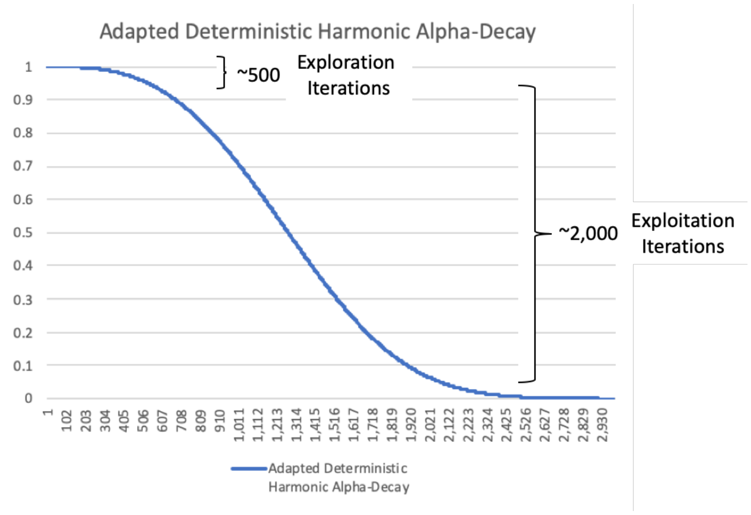

Figure 6: Exploration and Exploitation

5.2 ADP Stopping Criteria

An additional question to consider when implementing the ADP approach is determining how many total ADP simulation iterations n are needed to train the decision system model. One statistic that is commonly used to assist with determining the simulation stopping criteria is the Mean Square Error (MSE). Taken from Powell (2007), Equation 25 is the MSE calculation for the ADP approach.

$$MSE^n = (1 - \alpha_{n-1})MSE^{n-1} + \alpha_{n-1}\left(\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n}) - \hat{v}_t^n(S_t^n)\right)^2 \qquad (25)$$

The structure of the MSE equation is similar to the stochastic smoothing equation of Equation 23. The formula can leverage the same alpha-decay (i.e., step size) term α defined in Section 5.1. The formula consists of smoothing the stored MSE term from the previous iteration MSE^{n-1} with the squared difference between the value of the currently sampled state-space $\hat{v}_t^n(S_t^n)$ with the stored value of the previous time period's state-space position $\bar{V}_{t-1}^{n-1}(S_{t-1}^{x,n})$; the same two variables that were leveraged in Equation 23. Figure 7 provides an example of how the MSE statistic might work for a basic allocation model simulated for nearly 4,500 iterations.

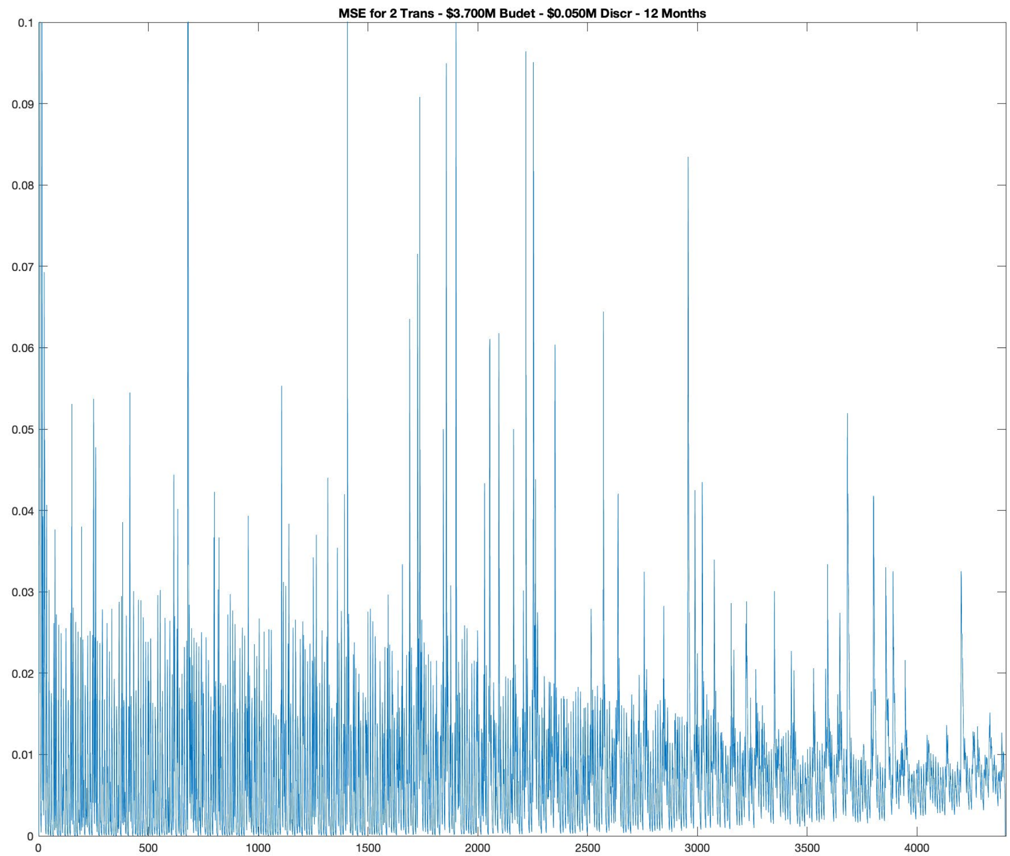

Figure 7: Mean Square Error Calculations

THIS PAGE LEFT INTENTIONALLY BLANK

Conclusion

This paper presents a framework for integrating ADP as a solution approach to DoD financial execution management. At the end of each FY, millions of unspent dollars are returned by weapon system program offices to DoD comptrollers as a result of use-or-lose budget environments. Currently, traditional FY cash allocation strategies implemented by program offices are myopic and risk projects receiving more funding than what can be spent within the FY calendar. ADP offers an alternative analytical tool that creates a sequential cash allocation plan balancing between the current allotment of funding to a project and the final end-of-year financial position of a project.

The next steps of this research involve testing the ADP algorithm in a theoretical DoD financial execution construct. ADP is a solution approach that contains flexibility allowing its structure to be modified to accommodate different parameters and facets that are unique to separate program offices. Further work will focus on experimenting with these different structures and facets of the ADP formulation and determining how they can be customized to capture realistic scenarios.

First, we will consider how different definitions of the epoch period t will impact the effectiveness of our model. In the example provided, t represented making a cash allocation decision, x_t, every month. Other options for t can include weekly or daily epochs. One rationale for changing the definition of t is to be able to better align it to the actual decision periods used by program offices. Another reason would be to evaluate to what extent making more cash allocation or fewer cash allocation decisions over a FY has on the objective of reducing the total amount of vulnerable end-of-year overcommitted funding.

Another feature to closely examine is the sensitivity variable α. The value α is a parameter that establishes how many time periods, t, into the future the current allotment of cash will be able to pay for project disbursements. Realistically, this value is dynamic and not static; its value would be dependent on the point in time in the FY in which a cash allocation decision is being made. If it is early in the FY, the program office may be comfortable with setting α at a larger value given that the contractor has a longer time period before the end of the FY to utilize the money. Then, the program office may implement a strategy that slowly reduces the parameter α as the FY calendar starts to approach the end of the year. Another strategy to use is if the program office is operating under a CRA is to set α to the length of time of the CRA. Under this scenario, program offices are aligning a project's cash allocation with the CRA timeframe.

Lastly, we look to consider different ways of defining the exogenous data $\widehat{D}_t$. At the start of each time period t, the ADP model simulates a sample of exogenous data $\widehat{D}_t$ and uses the information to define the current period's state-space S_t. The variable $\widehat{D}_t$ represents both the expenses (i.e., disbursement information) that occurred for a project in the previous time period along with the strategy for how this information is used to update the cumulative disbursement schedule $\bar{b}^d_{i,t}$. To provide more fidelity to the ADP model, $\widehat{D}_t$ can be uniquely defined for each project. For example, $\widehat{D}_t$ would take into consideration any available historical spending data on the project as well as subject matter expert input specifically related to the execution management of the project.

As part of this research effort, an initial exploratory and coordination review of the ADP model formulation was conducted with the Enterprise Analytics Leadership Team within the Office of the Under Secretary of Defense (Comptroller) (OUSD[C]) in Washington, DC. The team is currently in development of a big data enterprise platform called ADVANA. Among other analytic efforts, the ADVANA platform will be leveraged as part of the DoD's Dormant Account Review—Quarterly (DAR-Q) process, which investigates the status of unliquidated obligation and other funds at risk of expiring. In its current form, the analysis provided by the ADVANA platform is primarily descriptive. The ADP approach presented in this research provides a potential roadmap for enhancing the ADVANA effort by adding a predictive and prescriptive capability to its already existing descriptive analytic features. Furthermore, the ADVANA platform will include a cloud computing capability that can facilitate access to computers with greater computational capacity and therefore potentially serve as a mitigation strategy for dealing with some of the curse of dimensionality and computational complexity challenges of the ADP model. It is anticipated that continuing research efforts on this

ADP approach for tracking use-or-lose budgets will be conducted in conjunction with OUSD(C)'s Enterprise Analytics Leadership Team.

THIS PAGE LEFT INTENTIONALLY BLANK

References

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society, 60*(6), 503–515.

Bellman, R. (1957). *Dynamic programming.* Princeton, NJ: Princeton University Press.

Bellman, E. & Dreyfus, S. (1962). *Applied dynamic programming.* Princeton, NJ: Princeton University Press.

Conley, K., Dominy, J. R., Kneece, R. R., Mandelbaum, J., & Whitehead, S. K. (2014). *Implications of DoD funds execution policy for acquisition program management*. Alexandria, VA: Institute for Defense Analysis.

Cooley, W. T., & Ruhm, B. C. (2014). *A guide for DoD program managers*. Fort Belvoir, VA: Defense Acquisition University.

Crawford, J., Do, J., Leduc, C., Malik, A., Gormley, K., Luebke, E., Scherer, W. (2003). SWORD: The latest weapon in the project selection arsenal. *Proceedings of the 2003 IEEE Systems and Information Engineering Design Symposium.* Charlottesville, VA: IEEE Press.

Darken, C., Chang, J., & Moody, J. (1992). Learning rate schedules for faster stochastic gradient search. In *Neural Networks for Signal Processing 2, Proceedings of the 1992 IEEE Workshop.* Piscataway, NJ: IEEE Press.

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.

Defense Finance Accounting Service agency overview. (2019). Defense Finance and Accounting Service: Providing payment services of the U.S. Department of Defense. Retrieved from https://www.dfas.mil/pressroom/aboutdfas.html

Dell'Olmo, P., & Lulli, G. (2004). Planning activities in a network of logistic platforms with shared resources. *Annals of Operations Research. 129(1-4): 155-169.*

Easton, M. (2019, August 20). *Dormant account review quarterly* (FPM18-02). Retrieved from https://comptroller.defense.gov/Portals/45/documents/fmr/current/03/VOL_03_CH_08_20190826_DAR.Q.pdf

Gosavi, A. (2003). Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, *21*(2), 178–192.

Gosavi, A. (2003). *Simulation-based optimization: Parametric optimization techniques and reinforcement learning.* Boston, MA: Kluwer Academic Publishers.

Headquarters, Department of the Army. (2014). *Financial management operations* (FM 1-06). Retrieved from https://armypubs.army.mil/epubs/DR_pubs/DR_a/pdf/web/fm1_06.pdf

Howard, R. (1960). *Dynamic programming and Markov processes.* Cambridge, MA: MIT Press.

Mehta, A. (2018, October 1). DoD hands out $7.54 billion in year-end contract bonanza. *DefenseNews.* Retrieved from https://www.defensenews.com/pentagon/2018/10/01/dod-hands-out-754-billion-in-year-end-contract-bonanza/

Mortitz-Rabson, D. (2018, September 27). Pentagon rushes to spend billions before end of month. *Newsweek*. Retrieved from https://www.newsweek.com/pentagon-rushes-spend-billions-end-month-1137552

Powell, W. B. (2007). *Approximate dynamic programming: Solving the curse of dimensionality*. Hoboken, NJ: John Wiley & Sons.

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

Powell, W. B. (2010). Merging AI and OR to solve high-dimensional stochastic optimization problems using approximate dynamic programming. *INFORMS Journal on Computing*, *22*(1), 2–17.

Powell, W. B., George, A., Simão, H., Scott, W., Lamont, A., & Stewart, J. (2012). SMART: A stochastic multiscale model for the analysis of energy resources, technology, and policy. *INFORMS Journal on Computing*, *24*(4), 665–682.

Powell, W. B., Shapiro, J. A., & Simão, H. P. (2002). An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem. *Transportation Science, 36*(2): 231–249.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, *22*(3), 400–407.

Tremaine, R. L., & Seligman, D. J. (2013). OSD's obligation and expenditure rate goals: An examination of the factors contributing to the interference. *Defense AT&L (July–August)*, 14–21.

Tsitsiklis, J. N. (2010). Perspectives on stochastic optimization over time. *INFORMS Journal on Computing*, *22*(1), 18–19.

Under Secretary of Defense (Comptroller). (2011a). *Volume 2A: Budget formulation and presentation (Chapters 1–3)* (DoD 7000.14-R). Retrieved from https://comptroller.defense.gov/Portals/45/documents/fmr/Volume_02a.pdf

Under Secretary of Defense (Comptroller). (2011b). *Volume 2B: Budget formulation and presentation (Chapters 4–19)* (DoD 7000.14-R). Retrieved from https://comptroller.defense.gov/Portals/45/documents/fmr/Volume_02b.pdf

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net