# SYSTEM OF SYSTEMS STAKEHOLDER PLANNING IN A MULTI-STAKEHOLDER, MULTI-OBJECTIVE, AND UNCERTAIN ENVIRONMENT

A Dissertation
Presented to
The Academic Faculty

By

Nelson Gregory Andriano

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2021

# SYSTEM OF SYSTEMS STAKEHOLDER PLANNING IN A MULTI-STAKEHOLDER, MULTI-OBJECTIVE, AND UNCERTAIN ENVIRONMENT

Approved by:

Prof. Dimitri Mavris, Advisor
School of Aerospace Engineering
*Georgia Institute of Technology*

Prof. Daniel Schrage
School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Kelly Griendling
School of Aerospace Engineering
*Georgia Institute of Technology*

Gen. Philip Breedlove
School of International Affairs
*Georgia Institute of Technology*

Prof. Mariel Borowitz
School of International Affairs
*Georgia Institute of Technology*

Date Approved: July 26, 2021

We demand rigidly defined areas of doubt and uncertainty!

*Vroomfondel*

Dedicated to Zeke.

# ACKNOWLEDGMENTS

First, I would like to recognize the members of my committee. My advisor **Prof. Dimitri Mavris** showed me how to navigate the ambiguous world of self-directed research. **Prof. Daniel Schrage** helped provide context for my work and enabled me to focus on its applicability when my ideas seemed esoteric. I would like to thank **Dr. Kelly Griendling** for her advice and for holding me accountable over the years. She gave generously of her time – from that first step of taking qualifying exams – to the final step of defending my dissertation. **Prof. Mariel Borowitz** helped me to remain focused on the point of view of the analyst and future user of my methodology. **Gen. Philip Breedlove** provided a real-world perspective to keep my work firmly grounded in reality.

The journey toward earning a doctoral degree is a long one and requires the help of teachers, friends, and colleagues. My Montessori teacher **Mrs. MacMahn** instilled in me the mantra of always making my good my better and my better my best. Lifelong friends **Dr. Lizhou Huang** and **Joseph Wilder** took my mind off of work when our busy lives allowed. **Dr. Santiago Balestrini-Robinson** introduced me to Operations Analysis and Military Operations Research, a turning point in my interests and career. I thank those with whom I worked at the Aerospace Systems Design Lab to survive classes, qualifying exams, and research projects. **Dr. Joseph Iacobucci** provided motivational speeches and words of advice as I balanced school with life. **Lisa Martin** took on the challenge of fighting through unfamiliar jargon to help me edit my dissertation. Thank you to the **Fernadez family** for providing food, fun, and friendship.

My family is also key to my success. My grandmother **Dr. Peggy Cowen** inspired me from a young age and through her achievements helped me understand the value of higher education. Her example and legacy live on through me and many others who would not have succeeded without her. My grandfather **Richard Cowen** taught me the value of patience and planning. My parents **Fred and Stacia Andriano** have always supported me

and believed in me. I want to thank them for the opportunities they provided that turned my dreams into reality. My sisters also provided support: **Biz Andriano** helped me think critically and introspectively while **Janelle Holliday** provided much needed encouragement. A special thank you to my cats **Pyewacket** and **Dante** for supplying me with late-night company and random strings of letters.

Most importantly, I would like to thank my wife **Theresa Lyn Martin Andriano** for her continuous support during my PhD journey. She inherited this challenge and I could not have made it through without her. Lastly, I would like to thank **Nelson Ezekiel Andriano** for giving me the motivation to push across the finish line. A whole new adventure awaits.

# TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

The Department of Defense (DOD) planning process currently works to translate national strategic goals into a force structure. The Joint Capabilities Integration and Development System (JCIDS) requirements generation process for acquisition is a primary force structure driver and is built around reducing redundancy between organizations, enabling capability based acquisition, and evaluating both needs and solutions at a joint level. The JCIDS process is an example of raising the organizational level at which needs and resulting resource requests are decided. The current acquisition environment has imposed new fiscal and political constraints (e.g. budget reductions, continuing resolutions) while the mission requirements have increased with an increase in operational needs. Uncertainty in resources and requirements are driven by budgetary volatility and ever changing operational need. The trade-off between technology refresh, asset recapitalization, and asset reallocation has emerged as a primary driver during acquisition decisions to balance the constraints, needs, and uncertainty. New methods are needed to ensure that individual DoD stakeholders can maximize their mission success while remaining within the current constraints and dealing with the uncertainty without another level of consolidated coordination. A new methodology to devise a "playbook" of technology investment, system development, and system allocation strategies with regards to other stakeholder decision making and future uncertainty will help individual stakeholders better allocate their resources.

A military force structure can be defined as an acknowledged System of Systems (SoS). A body of work exists that addresses SoS Engineering processes, the evaluation of SoS performance, and SoS system evaluation. However, few approaches holistically address the SoS planning and evolution problem at the level needed to assist defense stakeholders in strategic planning. Current approaches do not address the impact of multiple-stakeholder decisions, multiple goals for each stakeholder, the uncertainty of decision outcomes, and the temporal component to strategic decision making.

The author developed a three step methodology to address the above short-comings to inform the production of a playbook for an individual stakeholder based on a review of the current state-of-the-art and the synthesis of existing methods from other fields.

A game framework, considered a Truth Model, is assumed for this work and represents the stakeholder's decisions and resulting outcomes played out over time. The first step creates a computationally reasonable meta-model from the complex game framework. The Truth Model is sampled using Monte Carlo techniques to generate $s, a, r, s$ sample tuples. The tuples are used to train a meta-model MDP. The meta-model results in a lower dimension state space composed of meta-model states, specific action based transition probabilities, and stochastic stakeholder rewards.

The second step addresses leveraging the now computationally manageable decision space to extract useful information for the stakeholder of interest. The MDP meta-model is used to evaluate risk-based policies, state significance, and action significance. A novel algorithm was developed based on mean-variance portfolio theory applied to stakeholder utility and combined with Reinforcement Learning (RL) policy iteration methods to construct risk-based policies using the MDP meta-model. Entropy measurements of stakeholder metrics are taken before and after each state to measure state significance. The opportunity cost between individual stakeholder metrics for a given action for each meta-model state-action pair is measured using a comparison of mean outcome and outcome variance.

The final step generates the information to help inform a stakeholder specific playbook. The risk-based policies are used to develop Risk-Tolerance Sensitivity Profiles (RTSP) at each state. A state RTSP can identify the Pareto efficient and inefficient actions with regards to risk and reward. The state RTSP can also identify the worst, low risk, and high risk actions. Additionally, decision spaces can be analyzed to identify consistent trends among similar RTSPs as well as bifurcations in RTSPs as a function of state values. The significant states and actions are identified using entropy and opportunity cost metrics.

The output of the method is the derived action and state based risk-based information and is provided to stakeholders to support the development of a risk-based playbook.

The methodology was created in part to test the applicability of existing and novel constructs. Three hypotheses were developed as part of reviewing current methods, synthesizing novel methods, and developing the overarching methodology. Hypothesis 1 asserts that Pareto efficient actions can be identified using the novel risk-based policy algorithm. Hypothesis 2 asserts that state-space reduction techniques can be applied to create a reduced MDP meta-model reducing computation time while maintaining usable risk-based policy outputs. Hypothesis 3 asserts that the risk-based policy metrics can be used to derive information above and beyond the current state-of-the-art, represented by optimal policy methods.

Experiment 1 tests Hypothesis 1 using an increasingly complex set of MDPs and demonstrates the ability of the risk-based policy algorithm to identify Pareto efficient and inefficient actions. Experiment 2 tests Hypothesis 2 by varying the state compression ratio and demonstrating both a reduction in computation time and the similarity of resulting risk-based policies. Experiment 3 tests Hypothesis 3 using both less complex scenarios and a single full complexity scenario. The full capability of the methodology is demonstrated and benchmarked against optimal policy methods. A significantly more nuanced set of information is shown when compared to the result from optimal policy methods.

The successful evaluation of each hypothesis demonstrates that the methodology can provide a military defense planner (a single SoS stakeholder) with information to develop a risk-based playbook to assist in decision making over time in an uncertain environment with multiple cooperative and non-cooperative stakeholders, budgetary constraints, and expanding operational needs. This will allow for robust planning at the stakeholder level without the need of an additional level of consolidation and review.

## Dissertation Structure

**Chapter 1** presents the motivation for approaching long-term strategic defense planning from a different perspective.

**Chapter 2** presents a general characterization of the defense planning problem and identifies challenges with regards to long term strategic defense planning.

**Chapter 3** reviews relevant background material, identifies gaps in current methods, and presents the research objective.

**Chapter 4** presents the research questions developed around the identified gaps, the subsequent literature review, the synthesis of concepts, and the hypotheses developed to address the research questions.

**Chapter 5** presents the methodology developed in response to the research objective and the developed hypotheses.

**Chapter 6** describes the experiments designed to evaluate the hypotheses.

**Chapter 7** presents the results and analysis from the experiments.

**Chapter 8** discusses the application of the methodology, examines the resolution of hypotheses, reflects on the research objective, summarizes contributions of this work, and provides areas of future work.

# CHAPTER 1
# MOTIVATION

The motivation for this work stems from the challenges and limitations of the current United States (U.S.) Department of Defense (DOD) Defense Planning system; specifically, the ability of the system to adjust to the imposed external environment and internal constraints. Resource constraints on the defense community have increased without a reduction in operational need. Reduced resources and increased required performance makes the value of cross-organizational force-level trades significant. The need for higher level trades increases each time the defense system is squeezed to a new normal. Doing more with limited resources requires an additional level of allocation management above the current level that the defense acquisition community provides today. Another level of management is not necessarily feasible but the need for individual stakeholders within the DoD to continue to maximize the use of their given resources still exists. The inherent distributed management of resources leads to the need for individual stakeholders to make strategic decisions with consideration of the decisions of both cooperative stakeholders in addition to changes in the global environment and non-cooperative stakeholders (adversaries). The strategic decisions stakeholders face when determining resource allocation over time include refreshing the technology used in deployed systems, reallocating existing resources to cover capability gaps, and expand current capabilities or systems.

## 1.1 Defense Planning

Defense Planning can be characterized as the “employment of analytical, planning, and programming efforts to determine what sort of armed forces a state needs” [1]. The current United States DoD approaches to Defense Planning, or force structure planning, is outlined in Figure 1.1. Strategic national-level guidance is given through the National Se-

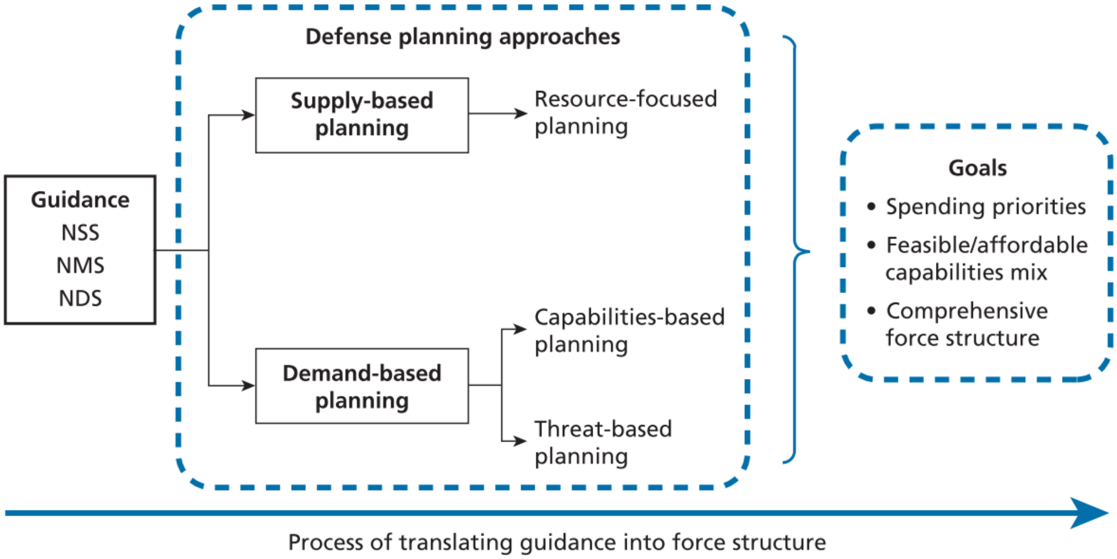

Defense planning approaches
Guidance
NSS
NMS
NDS
Supply-based planning
Resource-focused planning
Demand-based planning
Capabilities-based planning
Threat-based planning
Goals
• Spending priorities
• Feasible/affordable capabilities mix
• Comprehensive force structure
Process of translating guidance into force structure

Figure 1.1: Defense Planning Approaches [1]

curity Strategy (NSS), the National Defense Strategy (NDS), and the National Military Strategy (NMS) as inputs to Defense Planning. The outputs yielded by Defense Planning are spending priorities, feasible/affordable capabilities, and a comprehensive force structure.

Today, the effort varies to bridge the gap between the national-level strategy and the resource allocation, capability needs, and force structure. Demand-based planning constitutes the majority of planning and consists of developing new "strategies, capabilities, and capacities" [1]. Supply-based planning takes the opposite approach and looks at current structure, capabilities, and budgets in a more bottoms-up method. The Defense Planning process evaluates options across the different armed services as part of the current DoD acquisition system. The defense planning process is intimately tied to strategic defense planning, force structure planning, and, ultimately, defense acquisition. [2]

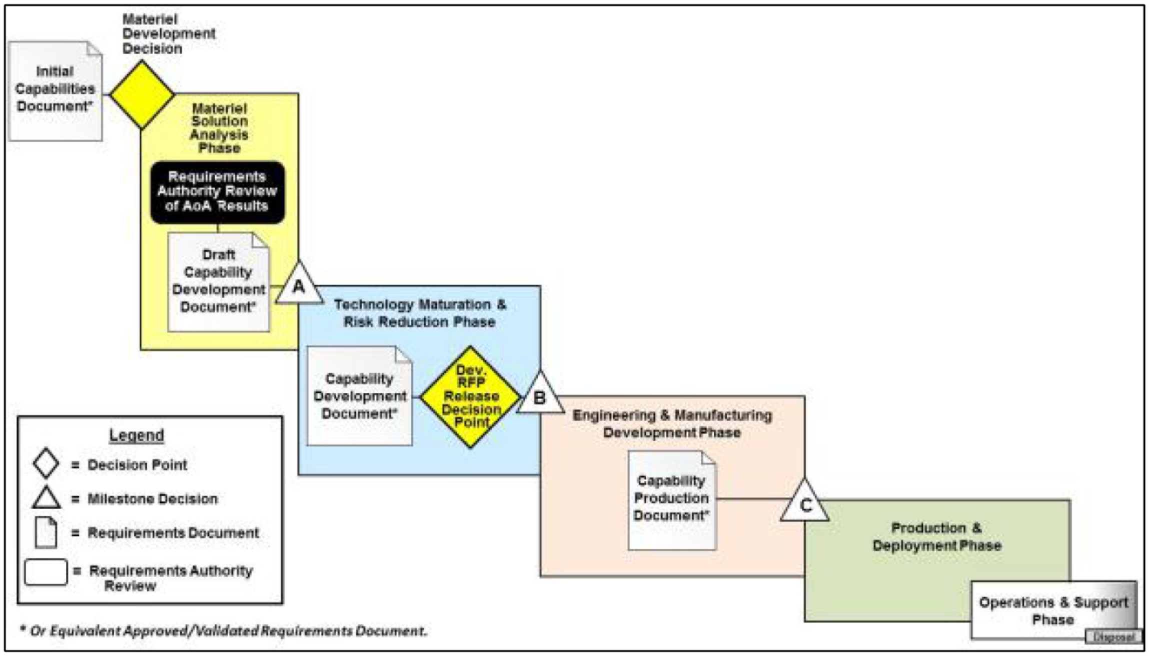

Materiel Development Decision
Initial Capabilities Document*
Materiel Solution Analysis Phase
Requirements Authority Review of AoA Results
Draft Capability Development Document*
A
Technology Maturation & Risk Reduction Phase
Capability Development Document*
Dev. RFP Release Decision Point
B
Engineering & Manufacturing Development Phase
Capability Production Document*
C
Production & Deployment Phase
Operations & Support Phase
Disposal
Legend
= Decision Point
= Milestone Decision
= Requirements Document
= Requirements Authority Review
* Or Equivalent Approved/Validated Requirements Document.

Figure 1.2: JCIDS Process Outline [4]

## 1.2 DoD Acquisition System

In 2003, The Department of Defense (DoD) reformed it's acquisition process and began using the Joint Capability Integration and Development System (JCIDS) [3]. The JCIDS was created as a top-down acquisition process that would allow better coordination between joint needs across DoD services. The goal is to facilitate synergies between branches and organizations of the DoD while moving towards a capability based planning process. The process does not begin with an idea for a new system or a technology concept but with the identification of needed capabilities. The JCIDS process begins by identifying current capability gaps through a Capability Based Assessment (CBA) as seen in Figure 1.2. The CBA identifies the needed mission and capability gaps that currently exist. The goal is to then define capability requirements which are not functional or physical requirements but are capabilities that are 'required to meet an organization's roles, functions, and missions in current or future operations'. [3]

The CBA first identifies the current gaps that exist in current capabilities and assesses for their risk against the completion of an organization's mission. The process moves forward to further action if there is an identified capability gap and the capability doesn't exist elsewhere in the joint force and the organization is not willing to accept the risk of no capability. Once the gap is identified and assessed as in need of addressing, it is either categorized as urgent need or a nominal need. For non-urgent needs, as most strategic acquisitions will be, first non-material (DOTMLPF-P) and then material (acquisition) are evaluated.

Studies and analyses related to the non-material and material solutions are done by the individual organizations and then shared via responsible Functional Capability Board (FCB) and with the Joint Capability Board (JPB) [3]. The goal of this step is to make sure a material solution (new acquisition or recapitalization) is necessary instead of reallocating existing resources. A non-material solution can be seen as a supply-based approach (using existing force structure, capabilities, and budget) and a non-material as a demand-based planning approach (increased force structure, capabilities, and budget).

If a non-material solution exists then a Doctrine, Organization, Training, materiel, Leadership and Education, Personnel, Facilities, and Policy (DOTMLPF-P) Change Recommendation (DCR) is the output of the CBA. Otherwise, if a material solution is determined to be needed, a Initial Capabilities Document is generated which describes the need for a material solution and outlines a material approach as the output of the CBA. The JROC can decide to accept the operation risk of no change, move forward with a non-material solution, or accept the need for a materiel solution. If a material solution is accepted, an Analysis of Alternatives (AoA) identifies alternative material solutions that could fulfill the capabilities identified in the ICD and evaluates the alternatives against their life-cycle cost and mission effectiveness. Ultimately the output of the AoA is a draft Capabilities Development Document (CDD) which outlines the requirements for the material solution. The draft CCD is the ultimate output of the Pre-Milestone A portion of the DoD JCIDS acquisition process and ultimately determines the next developed material solution. The material solution selected for Post-Milestone A development should ideally represent the needs of and solutions from a joint point of view. [3]

The goal of the JCIDS process is to develop and acquire based on capability needs, not legacy system and historical system specific missions. The initial purpose of the JCIDS process was to help reduce overlap in capability and system development by individual branches of the military by bringing joint requirements under a single roof. The JROC (Joint Requirements Oversight Council) uses the JCIDS process to balance joint needs equitably while making informed decisions. [3]

## 1.3 Pressures on DoD Long Term Planning

Over the last decade, a number of instigators have increased pressures on the defense planning system. (1) Budgetary pressures driven by sequestration and changing US priorities have began to increase the constraints on future acquisitions [5, 6, 7]. (2) The political volatility coupled with the short funding cycles have increased the amount of funding uncertainty. This uncertainty ripples through the defense community, government and contractor, and significantly impacts its long-term planning efforts [8]. (3) A pivot from asymmetrical warfare in Iraq and Afghanistan to the Pacific has significant impacts on the capabilities the DoD needs [9]. Technology changes, specifically an increase in connectivity, has enabled new possibilities in system collaboration. The current pivot outlines the need for an agile long-term strategic planning need. Long term strategic planning with in the DoD will have to account for more fiscal constraints and uncertainty while working to meet more mission requirements driven by more operational needs [10].

### 1.3.1 Budgetary Constraints and Uncertainty

The current domestic and international environment has increased financial resource constraints and uncertainty. These increases are driven by political and economic forces. Examining defense spending and its impacts over the last decade shows increased constraints with sequestration and uncertainty in funding due to Continuing Resolutions (CR). Sequestration was the result of a automatic spending cuts enacted in 2013 due to lack of intra-government agreement on a new budget act. During the first year, resource allocation decisions accounted for only near-term impacts to current programs with little account for interdependencies of cuts [11]. Short term decisions projected long term impacts [7]. In 2014, cuts to RDT&E were projected based on CRs demonstrating preference toward short term needs versus long term investment [12]. Continued sequestration put all programs at risk of delay or cancellation and upends the certainty of the long term enterprise force structure [5]. The level of available funding would not allow for all major weapons system acquisition to move forward as anticipated [13]. The cost constraints and uncertainty are not just a factor of resources, but also consumption in terms of schedule and cost performance of the systems [14, 11, 15]. Continuing resolutions impact cost and schedule uncertainty of existing programs and increase the difficulty of beginning new programs. Efficiencies in cost are reduced due to bulk buys and multi-year commitments to on going efforts. [8]

### 1.3.2 Changing Mission Requirements

In addition to shrinking budgets, worldwide security concerns, and a rapidly changing world —including technology development pace and threat advancement —drive a changing operational environment [13]. A enterprise military force structure must adjust to new mission needs and shifting mission priority over time.

After the cold war, the United States moved from a peer focus to a regional, traditional, non-peer threat focus, executing engagements like Desert Storm and Allied Force. Development over the next decade toward the long term goals of peer and near-peer threats left the US open to a new threat: terrorism. An asymmetrical threat developed and dominated US military missions for the next decade and a half from 2001 to near present. The United States was not prepared for the shift and required a significant revectoring to accommodate the new mission.

Yet again, a “Pivot to the Pacific” was declared and heads have been turned to near-peer and peer threats for short term planning purposes [9, 16]. The Third Offset Strategy was borne out of a need to address this new and changing environment with the goal of utilizing technology superiority to enable a favorable cost differential for the United States [17, 18, 19].

The defense strategic planning process could be improved if the short term changes to current force structure remained flexible, despite long-term development. A method that allows the impact of a changing environment and mission priorities over time (e.g. near-peer to asymmetric adversary) would enable strategic and tactical force-structure planning to account for the ability to re-vector and re-orient the focus of the force structure.

## 1.4 Gaps in Current Military Strategic Planning

Section 1.1 and 1.2 outline the current Defense Planning system that drives the defense structure planning for the US DoD. There are a number of gaps that exist with the current strategic planning process within the existing acquisition system for individual stakeholders. The specific issues derive from the ability of the current process to fully allocate resources and make trades across divisional boundaries and across time frames. Allocating resources across divisional, or stakeholder, boundaries would enable resource sharing to better jointly prepare for individual stakeholder needs and to jointly allocate existing assets to better accomplish multiple missions. Additionally, there is a lack of understanding between the value of resources allocated to a short term versus a long term need that is necessary to enable temporal trades.

### 1.4.1 Stove Piped Acquisitions

In March of 2007, the Government Accountability Office (GAO) examined methods that could be used to better support weapon system program stability (i.e. better control over cost and schedule overruns). A number of commercial companies were surveyed and it was found that enacting portfolio management techniques to evaluate cost/benefits in terms of viability would increase stability. Specifically, appropriate portfolio management techniques combined with the organizational capability to cut losses and make go/no-go decisions based on anticipated viability were determined key to success. The GAO found that the government make long term commitments early without respect to long term viability and overall portfolio performance. Overall portfolio management is considered at the joint level. It was found that individual services, though part of joint forces during mission execution, individually allocate resources. Integrated portfolio management was identified as a serious need despite the implementation and continued refinement of the JCIDS process. [20]

In August of 2015, a follow up report from the GAO was released which looked at the extent to which the DoD had implemented the recommended integrated portfolio management approach. The GAO found that the implementation of the integrated portfolio management system was inadequate and identified affordability challenges and program duplication as evidence. It was found that integrating requirements, acquisition, and budget information at an enterprise level would help but is typically hampered by fragmented governance, lack of sustained leadership and policy, and perceived lack of decision making authority. The GAO recommended high-level oversight, frequent reviews integrated with key decisions points, and investment in analytical tools to support efforts. [21] Much of the recommendations seem to echo the purpose behind the existing JCIDS process.

Part of the responsibilities of the JROC is to look across the forces and determine overlapping needs and capabilities before approving a solution. In 2011, the GAO Reviewed a number of requirements approvals and found the the "JROC does not currently prioritize requirements, consider redundancies across proposed programs, or prioritize and analyze capability gaps in a consistent manner" [22]. The result of the review is an example of a continued stakeholder and mission stove piped acquisition process living within the new

JCIDS process.

The current implementation of the JCIDS process does not meet its initial goals [23, 22, 24, 25]. Influenced by legacy acquisition practices, there is still a significant focus on one for one replacement present and issues crossing traditional stove pipes. Another level of consolidated joint planning would not bring about an optimum resource allocation. There is no appropriate consolidation level above the current JCIDS process. Stakeholders within the DoD can only manage their own missions within the scope they control through technology refresh, asset recapitalization, and asset reallocation. Each stakeholder does not control the ultimate mission utility generated due to the many inter-dependencies between the missions stakeholders need to accomplish and the assets they each control. There is a need to enable stakeholders to optimize their resource allocation amidst this multi-mission, mutli-stakeholder environment when no true overarching centralized authority exists.

There has been significant duplication in overlap between various DoD organizations with respect to acquiring capabilities. A single example of this is the acquisition of ISR platforms over the last two decades in support and in the aftermath of the Iraq war. A case study that can act as a single use case to demonstrate current issues is the DoD's approach to it's enterprise Intelligence, Surveillance, and Reconnaissance (ISR) over this time frame. [26, 27, 28] This exemplar case exhibits a multi-stakeholder and multi-mission environment with stove piped acquisitions dealing with constricting budgets and shifting mission needs.

### 1.4.2 Balancing Long and Short Term Needs

The military strategic planning process can looked at through the lens of the 'Iron Triangle of Painful Trade-offs (ITPT)' [29]. The ITPT is characterized by the need to balance "preparing to be ready today (readiness), preparing to be ready tomorrow (investment), and sizing the force (structure)". This can be recast as a short term (readiness) vs. medium term (structure) vs. long term need (investment). Cancian reorganizes the triangle as "readiness (the ability of forces to do what they were designed to do), capacity (the size of the force), and capability (the ability of forces or equipment to achieve a desired effect)" [30]. Given a need for readiness (training and immediate preparedness for military conflict) there are trades within the structure/capacity and the investment/capability categories. Structure and capacity involved the resources allocated toward increasing the number of existing systems or reallocating systems to new missions. Investing in future capability includes trades between new more-expensive system acquisitions (recapitalization of assets) and investing in modernization (technology refresh). An increasingly common key trade has developed between recapitalizing assets, refreshing technology of currently deployed assets, and reallocating existing resources.

The trade-off between short and long term needs is intertwined with ever changing mission requirements 1.3.2. Wong examined the impact of short term needs when planning is made with regards to longer term needs. A lack in flexibility within the system to identify, analyze, and make acquisition decisions to provide flexibility to the force structure needed to deal with near term changes in the environment. [31] The current planning and acquisition system is not equipped to handle the trade-off between short term and long term needs given the long term uncertainty in changing mission requirements coupled with constricting resources.

### 1.5 Impacts on Strategic Force Planning

The pressures described in Section 1.3 on the current defense planning system have had and will have significant impacts on individual stakeholders within the defense planning and acquisition communities. Planning and operating under severe budgetary constraints and budget uncertainty while addressing an ever-evolving set of requirements will significantly impact the acquisition of new systems. Continuing to strategically plan new technologies and new systems using the current process will have a number of failure points that will result in an ill prepared force that lacks the required capabilities to adequately execute needed operations.

The current acquisition process is (1) not structured to address defense planning holistically and to take advantage of capability synergies in a single acquisition. The current process also (2) lacks the capability to address high level force trades between individually stove piped planning groups. The ability to measure and evaluate high level force structures across missions and groups is needed to address the issue. It is also imperative to (3) join the long term strategic investing (technology development and early research) with the tactical investing (asset acquisition) across organizations. Strategic investing directly impacts the available capabilities for tactical investments. All of the above needs fall into the category of increasing efficiencies to adjust to the pressures that the acquisition system is experiencing. The efficiencies come together to facilitate a key and driving need. Ultimately, the increased pressures create a need for acquisition planning to analyze the trade-offs between technology refresh, system re-capitalization, and current asset reallocation at a much higher force structure level which the current acquisition system is not prepared to address. Alternatively, it is the imperative of individual branches to try to best allocate their resources (in an optimal or robust manner) within the current paradigm of a loosely consolidated planning environment.

## 1.6 Motivation Summary

Several key insights can be taken from the current world environment described in the first chapter of this proposal. Numerous stressing constraints, needs, and uncertainties have emerged simultaneously highlighting the shortcomings of the current strategic planning approach to acquiring, deploying, and managing military systems. Specifically, there are tightening constraints that, at the same time as making trade-offs more significant, are introducing larger amounts of uncertainty to force level planning:

1. Increased Resource Constraints
2. Increased Resource Uncertainty

3. Acquisition Cost and Schedule Uncertainty
4. Shifting Operational Need over Time
5. Multi-Mission Objectives
6. Evolving and Dynamic Threats

The increase in constraints and needs, along with the accompanying uncertainty to each, has shown that the current process to acquire, deploy, and manage the United States force structure as a whole has the following shortcomings:

1. Stove Piped Planning and Acquisition Process
2. Lack of Planning Under Uncertainty
3. Inability to Efficiently Trade Short and Long Term Needs
4. Lack of Balancing Increased Constraints with Increased Operational Need

# CHAPTER 2
# PROBLEM CHARACTERIZATION

The motivation for this research described in Chapter 1 outlines a growing issue with defense planning as systems designed, acquired, and deployed by the United States government become increasingly inter-connected and complex while at the same time there are increased stresses in both resources and in capability need. The number of trade-offs that must be made during acquisition has increased due to lower level budget constraints and the greater demand for capabilities to address continuously evolving threats.

Developing and applying design methodologies requires first understanding and then characterizing the problem at hand. In this chapter, a System of System definition and taxonomy is synthesized. The strategic force level trade problem that motivates this research is then categorized and classified against the existing SoS body of work. Additionally, aspects of the defense planning problem that do not currently fit within classification methods or need further definition are identified and included in the final synthesized taxonomy. A clear characterization of the problem allows the further exploration of the applicability of specific methods and techniques.

## 2.1 A System of Systems Definition

The term "System of Systems" has become commonplace and overused. In this section, the author addresses the issue by defining terms that will be used for the remainder of this work.

### 2.1.1 System Characteristics

Using the term System-of-Systems implies that there is an understood definition of a System. The term "System", in the context of this proposal, is defined in order to help define

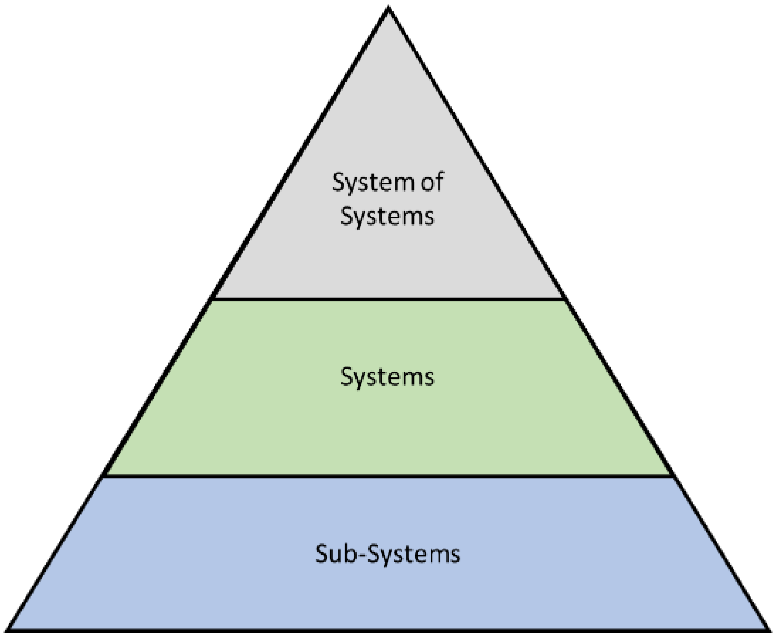


Figure 2.1: System of System Hierarchy

a System of Systems and the taxonomy used to address them. A system can be seen as the building block of a System of Systems just as traditional Systems Engineering defines functional and physical Sub-Systems. A Sub-System is a delineation that applies to components of a system that would be unable to function or contribute without the additional Sub-Systems defined within the System [32]. A system can then be defined as:

- “any set of related parts for which there is sufficient coherence between the parts to make viewing them as a whole useful” [32].
- “combination of interacting elements organized to achieve one or more stated purposes” [33].

The commonality between definitions is bringing interacting elements together to accommodate a specific function. The general definition can be applied very broadly and does not draw strict problem boundaries. This leads to defining the complexity of a System. An example of such a taxonomy to describe and classify can be seen in Figure 2.1. [34]

Table 2.1: System Taxonomy [35]

| **System Type** | **System Type** | **System Type** |
|---|---|---|
| Simple System | Small number of components<br>Act according to well understood laws | Pendulum |
| Complicated System | Large number of components<br>Well defined components<br>Components behavior is well understood | Boeing 747-400 |
| Complex System | Large number of components<br>Components change behavior over time<br>Behavior of components not well understood | Flock of Geese<br>Stock Market |

The system taxonomy does not give measurable boundaries but does give a context from which to begin. The simple system can be described as something that is fully understood and quantifiable. The behavior can be explicitly described and predicted. The development of Simple Systems is commonplace and needs little overview. The understanding of Simple Systems allows the development of Complicated Systems. Complicated Systems are not an uncommon occurrence and are regularly developed, built, and deployed. Contemporary Systems Engineering and Project Management practices have developed to manage the System problem whereby the stakeholders, needed capabilities, requirements, resources, and timeline are well defined and understood. [32]

The third and final category in the system taxonomy breaks down the assumption of the solution being well defined and understood. A Complex System can also have a large number of components but it lacks the system definition that is common in Simple and Complicated Systems. Complicated Systems may not be developed but evolve from the interaction of complex systems. Over time, the behavior of the Complex System changes in response to the changing of the components that create it. [35]

For the purpose of this proposal, it is important to differentiate between commonly developed Complicated Systems and what they are brought together to create. A Complicated

System, referred to onwards as a System, can be defined as [36]:

- has a well understood and controlled life-cycle.
- is discrete and can by itself provide utility.
- is created from components that do not independently provide utility.
- is created to fulfill a well-defined specification to produce well defined capabilities.
- has strict control over its components.
- as predictable behavior and can be tested before deployment.

Defining a System allows a characterization of Complex Systems to be developed. This characterization is commonly known as a System of Systems.

### 2.1.2 System of System Characteristics

The term System of Systems has enjoyed an increase in use and attention over the last twenty years. The general conceptual idea it represents has been defined many times: System of Systems, Federation of Systems, Complex Systems, Collaborative Systems, Network-Centric Systems, etc. There have been many attempts to consolidate and define the concept of a System of Systems [36, 37, 38, 39, 40] along with distinct individual interpretations [39].

For the purpose of this work, the characterization of a System of Systems is based on the previously defined System characteristics. This relative characterization is important to distinguish a System of Systems from its components. It also allows the juxtaposition of the currently well understood Systems Engineering process and the developing System of Systems Engineering process. The most accepted primary characteristics of a SoS are [41, 42, 43, 44]:

- evolves over time and has a loosely defined life-cycle as constituent systems leave and new ones join (**Evolution**) .

- is created from components that can operation independently of the system to fulfill a desired purpose (**Operational Independence**).
- is comprised of constituents that join together to fulfill a greater purpose beyond their individual capabilities.
- is not directed or controlled by a central authority (**Managerial Independence**).
- does not have fully understood behavior and modes which leads to the emergence of behavior as the SoS operations and evolves (**Emergent Behavior**).
- is often dispersed geographically (**Geographic Distribution**).

These primary characteristics of a SoS lead to additional secondary characteristics that are commonly addressed. The Managerial and Operational independence leads to a collection of **Stakeholders** that represent both the constituent systems and the SoS. There may be competition for resources and direction of the SoS or simply a prioritization of each constituent system over the greater SoS. Many times, there is no single decision maker but many decision makers acting in either a coordinated or independent fashion. [43]

The addition of many stakeholders brings into effect more than just technological solutions and expands the problem to be **Trans-Domain**. As one examines both the evolution and operations of a SoS it becomes necessary to address more than just stand alone system development and supporting engineering and programmatics. It is essential to include additional fields such as economics and public policy [45].

The evolutionary aspects and the dispersed control of a SoS make it difficult to define the boundary of the SoS. A typical System is defined in scope by the stakeholder and requirements where there are neither a single stakeholder nor a set of requirements provided to a SoS. The **Fuzzy Boundary** condition is itself not an issue but becomes one when there is a desire to analyze and guide the SoS as it evolves.

There are two remaining consequences of the primary characteristics are **Diversity**

Table 2.2: System Engineering vs. System of Systems Engineering Characteristics [46]

| | **SE** | **SoSE** |
|---|---|---|
| **Focus** | Single Complex System | Multiple Integrated Complex Systems |
| **Objective** | Optimization | Satisficing, Sustainment |
| **Boundaries** | Static | Dynamic |
| **Problem** | Defined | Emergent |
| **Structure** | Hierarchical | Network |
| **Goals** | Unitary | Pluralistic |
| **Approach** | Process | Methodology |
| **Timeframe** | System Life Cycle | Continuous |
| **Centricity** | Platform | Network |
| **Tools** | Many | Few |
| **Management Framework** | Established | Not Established |

and **Connectivity** of constituent systems. Often, the resulting SoS is composed of a non-homogeneous set of similarly defined and behaving systems. The result of having a large number of geographically dispersed and heterogeneous constituents drives **Interoperability** and connectivity to be a key aspect of the performance of the SoS. The degree of interoperability (ability to effectively communicate and collaborate) and connectivity (degree to which the constituent systems are connected and share information) explicitly define the complexity of a SoS.

Similarly, Boardman and Sauser developed elements, or axis, by which to measure the difference between a System and a System-of-Systems [36]. The specific characteristics are outlined in Table 2.3.

### 2.1.3 Synthesized System of Systems Definition

Moving forward in this work, the overarching and significant determinants that describe a SoS are as follows:

**Operational Independence:** composed of distinguishable parts which alone can perform their designed purpose.

Table 2.3: System Engineering vs. System of Systems Engineering Characteristics [46]

| **Element** | **SE** | **SoSE** |
|---|---|---|
| **Autonomy** | Autonomy is ceded by parts in order to grant autonomy to the system. | Autonomy is exercised by constituent systems in order to fulfill the purpose of the SoS. |
| **Belonging** | Parts are akin to family members; they did not choose themselves but came from parents. Belonging of parts is their nature. | Constituent systems choose to belong on a cost/benefit basis; also in order to cause greater fulfillment of their own purposes, and because of believe in the SoS supra purpose |
| **Connectivity** | Prescient design, along with parts, with high connectivity hidden in elements, and minimum connectivity among major subsystems | Dynamically supplied by constituent systems with every possibility of myriad connections between constituent systems, possibly via a net-centric architecture, to enhance SoS capability. |
| **Diversity** | Managed i.e. reduced or minimized by modular hierarchy; parts' diversity encapsulated to create a known discrete module whose nature is to project simplicity into the next level of the hierarchy | Increased diversity in SoS capability achieved vy released autonomy, committed belonging, and open connectivity |
| **Emergence** | Foreseen, both good and bad behavior, and designed in or tested out as appropriate. | Enhanced by deliberately not being foreseen, though its crucial importance is, and by creating an emergence capability climate, that will support early detection and elimination of bad behaviors |

**Evolution:** evolves as components are added and removed over time without a beginning or an end.

**Managerial Independence:** there is no centralized directing authority such as a single stakeholder or contributor.

Characteristics of a SoS (emergent behavior, connectivity, interdependence, etc.) are separate from a specific definition. Characteristics of a SoS are further expanded as a Taxonomy is developed to identify and categorize occurrences. Under this simplified definition a military force structure can be clearly categorized as a SoS.

Within the United States military there exists assets separately controlled and managed which are attempting to work toward common goals and missions. These groupings of systems with disparate control and guidance are also subject to the continuous acquisition, deployment, and retirement of assets. Viewing the force level trade problem through the lens of a SoS with multiple stakeholders and multiple objectives is a first step towards identifying solutions to the strategic planning process.

## 2.2 System of Systems Taxonomy

A definition defines a term based on an external viewpoint. The next step, once a definition is solidified, is to turn inward and begin to classify within the original definition. A Taxonomy allows further analysis and breakdown of a problem set to enable tailored responses. A Taxonomy is necessary to refine and classify a SoS based on the characteristics defined in the previous section. There have been just as many efforts to define SoS Taxonomies as there have been to develop a SoS definition. Each taxonomy has similarities and nuances that help define a standard viewpoint of SoS as well as set them apart. In this section, prominent taxonomies are reviewed and reflected upon. Ultimately, a common taxonomy is synthesized and used to classify the force level trade problem presented earlier. This classification allows for the identification of similar problem sets and definition of the solution space.

### 2.2.1 Traditional Taxonomy

Maier's first largely accepted definition of a System-of-Systems also was branded with the initial taxonomy by which to classify them. This can be thought of as the Traditional Taxonomy first used to classify and categorize types of Systems-of-Systems. This taxonomy first identifies the discriminating factors that characterize the System of Interest (SOI) with respect to Maier's SoS definition described previously (focusing on Managerial and Operational Independence) and then uses the taxonomy to classify the SoS. The Traditional Classification focuses on the amount of centrality of the SoS with respect to the Operational and Managerial Independence it possesses. The taxonomy can be considered in many ways to be a single axis classification despite the two properties the classification relies upon.

The degree of centrality that leads to the classification revolves around two main considerations: The level of managerial independence and the level of operational independence the constituent systems have. From Maier [47] the two properties by which a SoS should be classified are as follows:

**Operational Independence of the Components:** If the system-of-systems is disassembled into its component systems the component systems must be able to usefully operate independently. That is, the components fulfill customer-operator purposes on their own.

**Managerial Independence of the Components:** The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the system-of-systems.

The taxonomy is broken into four classifications [47, 48, 44]:

**Virtual:** "The SoS lacks central management and a centrally agreed-upon purpose."

**Collaborative:** “Component systems within the SoS interact more or less voluntarily to fulfill agreed upon-central purposes.”

**Acknowledged:** “The SoS has recognized objectives, a designated manager, and resources, while the constituent systems retain their independent ownership, objectives, funding, development, and sustainment approaches.”

**Directed:** “The SoS is built and managed to fulfill specific purposes. Constituent systems operate independently, but their normal operational mode is subordinate to central management purposes.”

### 2.2.2 Qualitative Taxonomy

It is clear that there can be more than a single axis on which to define and classify a System-of-Systems. The Traditional Taxonomy defines a range from centralized managerial and operational control to a fully federated SoS with independent managerial and operational control. As the field of Systems-of-Systems Engineering (SoSE) has developed, there has been a greater and greater need of a more descriptive way to describe and classify a SoS. Typically, these SoS Taxonomies look to measure the different axis by which a SoS diverges from a traditional system. Ultimately, these taxonomies pull heavily from the definition of a SoS versus a System. [39]

The axis of this taxonomy system are Autonomy, Belonging, Connectivity, Diversity, and Emergence as outlined in the relative System-of-Systems definition in Figure 2.2. This taxonomy looks qualitatively at the point at which a System of System deviates from a System against these metrics.

### 2.2.3 Three Dimensional Taxonomy

Independently from the Gorod taxonomy described above, Delaurentis defined a separate taxonomy from which to view a SoS. This is a three axis taxonomy with elements for

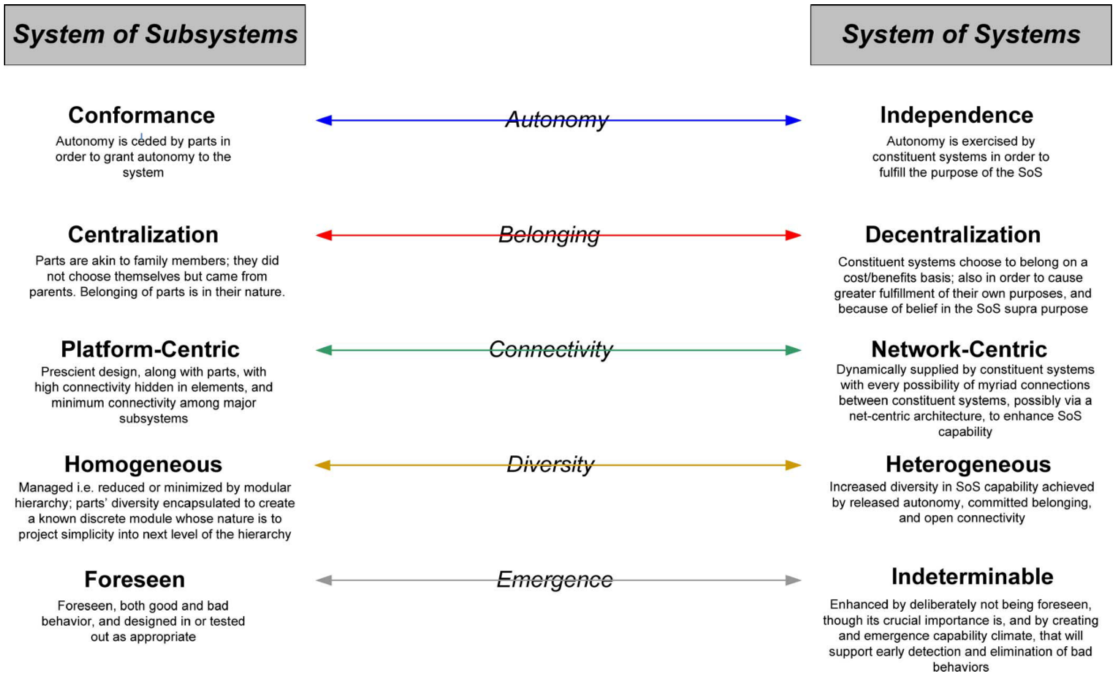

System of Subsystems
System of Systems
Conformance
Autonomy is ceded by parts in order to grant autonomy to the system
Autonomy
Independence
Autonomy is exercised by constituent systems in order to fulfill the purpose of the SoS
Centralization
Parts are akin to family members; they did not choose themselves but came from parents. Belonging of parts is in their nature.
Belonging
Decentralization
Constituent systems choose to belong on a cost/benefits basis; also in order to cause greater fulfillment of their own purposes, and because of belief in the SoS supra purpose
Platform-Centric
Prescient design, along with parts, with high connectivity hidden in elements, and minimum connectivity among major subsystems
Connectivity
Network-Centric
Dynamically supplied by constituent systems with every possibility of myriad connections between constituent systems, possibly via a net-centric architecture, to enhance SoS capability
Homogeneous
Managed i.e. reduced or minimized by modular hierarchy; parts' diversity encapsulated to create a known discrete module whose nature is to project simplicity into next level of the hierarchy
Diversity
Heterogeneous
Increased diversity in SoS capability achieved by released autonomy, committed belonging, and open connectivity
Foreseen
Foreseen, both good and bad behavior, and designed in or tested out as appropriate
Emergence
Indeterminable
Enhanced by deliberately not being foreseen, though its crucial importance is, and by creating and emergence capability climate, that will support early detection and elimination of bad behaviors

Figure 2.2: SE vs. SoS Defined Taxonomy [39]

Connectivity, Majority Type of System, and Control/Autonomy [45]:

- **System Type:** Spectrum of wholly human system to wholly technological
- **Control of Systems:** Spectrum of fully centralized control to full autonomy granted to individual constituent systems (control/autonomy)
- **Connectivity of Systems:** Degree to which constituent systems are interdependent and share information

The Three Dimensional Taxonomy does not rely on the definition of a System of Systems. It relies on the evaluation characteristics of the SoS while still being based on the structure and behavior of the System of Systems. Connectivity plays a role in the analysis of a network or network topology and its impact on the evaluation of a SoS. The idea of Control of Systems is derived from Maier's initial taxonomy that looks at Managerial and Operational Control. The Connectivity of Systems is derived from the need to

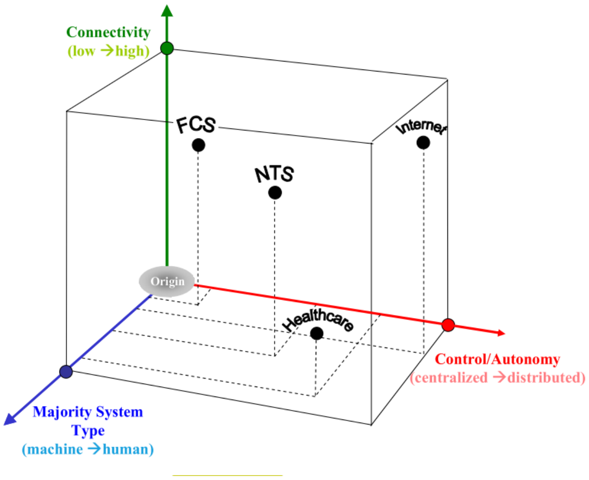


Figure 2.3: Three Dimensional SoS Taxonomy

evaluate emergent behavior based on the amount of intra-SoS information exchange and intra-dependencies. An example of classifying systems is captured in 2.3. The Army's Future Combat System (FCS), the National Transport System (NTS), the internet, and US healthcare are represented. The FCS is highly centralized (as many defense SoS are) and designed with high connectivity for a human dominated system. The NTS has high connectivity with moderate federation and a moderate amount of autonomy. The internet as expected is high on machine versus human domination, high connectivity, and significant decentralization.

## 2.3 Problem Synthesis and Description

The cycle of defense strategic and force level planning was outlined by Liotta as seen in Figure 2.4. The depicted cycle works to take the defense planning goals [1] and tie them down to the acquisition process [3]. This cycle can be combined with the depicted SoS Trapeze model [49, 50] to yield a problem specific and SoS stakeholder specific description (Figure 2.5).

The problem of stakeholder planning can be viewed in terms of the feedback loop depicted in Figure 2.5 where the SoS referenced would be any number of those existing under

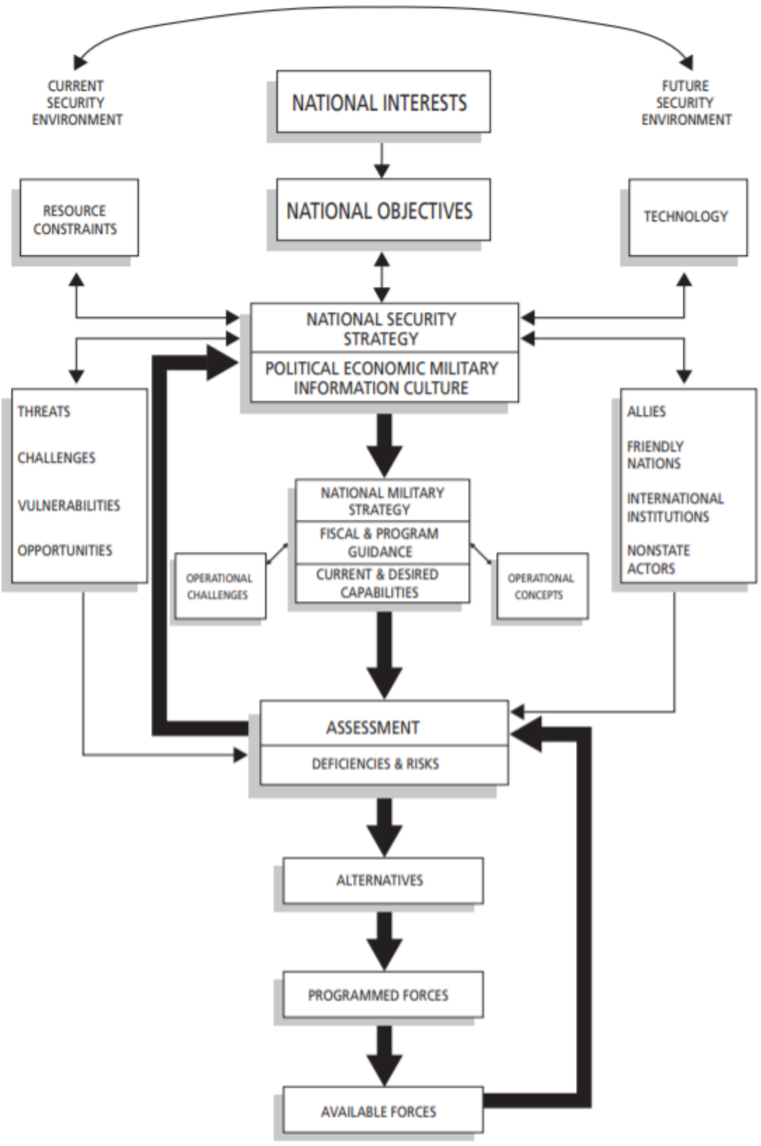


Figure 2.4: Defense Strategy and Force Planning Framework [2]

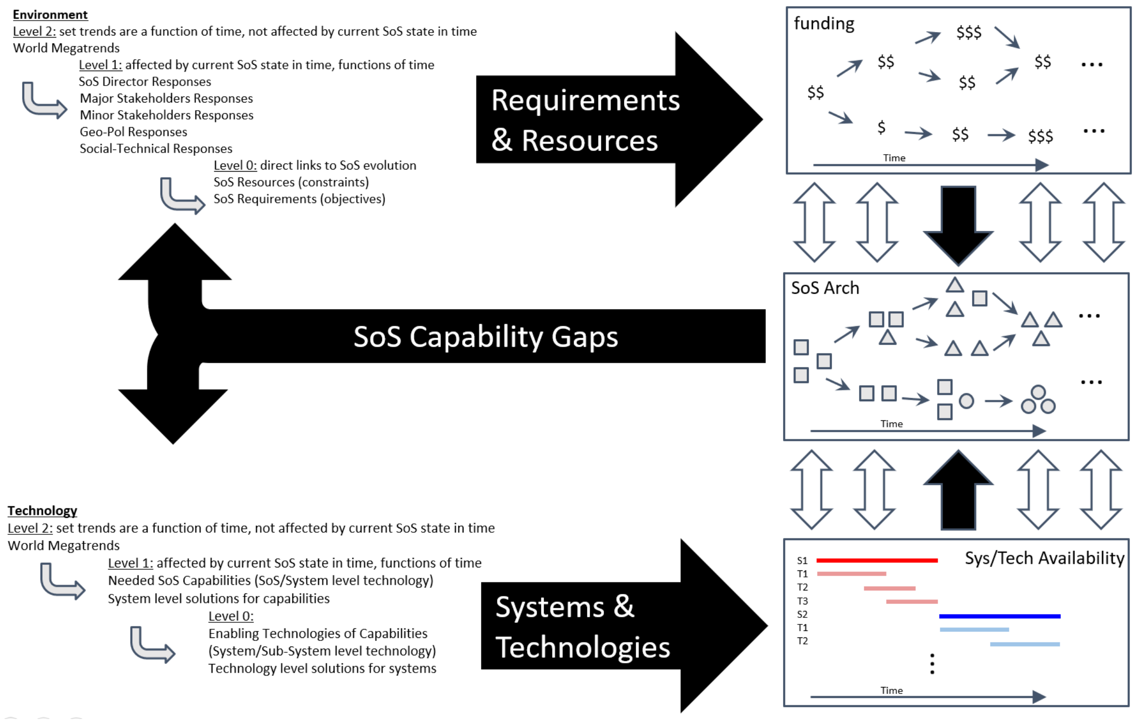

Environment
Level 2: set trends are a function of time, not affected by current SoS state in time
World Megatrends
Level 1: affected by current SoS state in time, functions of time
SoS Director Responses
Major Stakeholders Responses
Minor Stakeholders Responses
Geo-Pol Responses
Social-Technical Responses
Level 0: direct links to SoS evolution
SoS Resources (constraints)
SoS Requirements (objectives)
Requirements & Resources
funding
$$
$$
$$$
$$
$$
$
$$
$$$
Time
SoS Capability Gaps
SoS Arch
Time
Technology
Level 2: set trends are a function of time, not affected by current SoS state in time
World Megatrends
Level 1: affected by current SoS state in time, functions of time
Needed SoS Capabilities (SoS/System level technology)
System level solutions for capabilities
Level 0:
Enabling Technologies of Capabilities
(System/Sub-System level technology)
Technology level solutions for systems
Systems & Technologies
Sys/Tech Availability
S1
T1
T2
T3
S2
T1
T2
Time

Figure 2.5: Framing the Evolution of Defense SoS Problem

the umbrella of general force level planning. The current state capabilities is observed by stakeholders and decisionmakers who act as the catalyst for decisions. Strategic technology investments and tactical system acquisition decisions are made and feed the availability of systems and technology. The environment which stakeholders respond to is influenced externally by the operating environment and internally by budgets and priorities. These lead to requirements and budgets that drive and constrain the available decisions (technology and system investments).

A conceptual diagram of a methodology that addresses the evolution of SoS is depicted in Figure 2.6. The environment specifically influences the current capabilities of the SoS and the priorities of stakeholders. Technology and system investments are made based on the perceived lack of capability. Together, the available resources (technology, system) and requirements (objectives, desires of stakeholders) provide the SoS architecture tradespace and performance metrics. Architectures can be selected and evaluated using methods

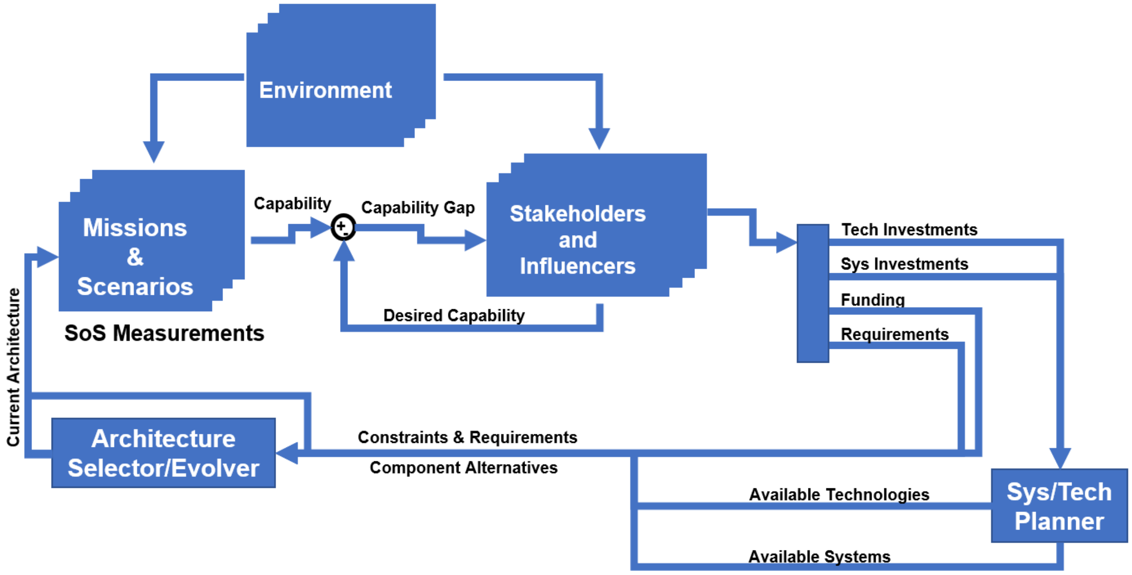


Figure 2.6: Conceptual System of System Evolver

(further explored in Chapter 3) and provide the capability at a given time.

### 2.3.1 Intuitive Example Problem: Acquire or Develop

An intuitive example problem is used to bring more depth and understanding to the previously depicted theoretical problem synthesis. A understandable and concrete example is used to explain initial concepts. The example problem encompasses key components but leaves some complexity to be added to allow ease of understanding. This example problem is used in Section 5 as an anchor point to walk through the Methodology.

*Example Problem Overview*

The intuitive example problem is a simple constructed multi-step game between two non-cooperative stakeholders. At each time step, specific actions are available to each stakeholder and represent potential decisions in the future. Each stakeholder can either develop a new system or acquire a system previously developed. Discrete time steps are used as decision points with some actions have impacts multiple time steps later.

Table 2.4: Intuitive Example: Stakeholder System Ownership

| | **Stakeholder 1** | **Stakeholder 2** |
|---|---|---|
| System 1 | 1 | 0 |
| System 2 | 1 | 0 |
| System 3 | 1 | 0 |
| System 4 | 1 | 0 |
| System 5 | 0 | 1 |
| System 6 | 0 | 1 |
| System 7 | 0 | 1 |
| System 8 | 0 | 1 |

A state is defined as the number of systems available for deployment at the specified time step. Each system has a associated quantified performance it contributes to a Stakeholder 1 versus Stakeholder 2 outcome. Stakeholder 1 systems impacts positively to the engagement outcome and Stakeholder 2 systems negatively. A composite is used to develop a single resulting mission score. A zero-sum game construct is used to define the ultimate utility of each stakeholder.

The problem setup is described further below:

**Stakeholders:** Two non-cooperative stakeholders, or players.

**Systems:** Eight systems with four attributed to each stakeholder (Table 2.4).

**System Life-Cycle:** Each system follows the same life-cycle (Figure 2.7). The owning stakeholder can make a decision to develop he system (equivalent to RDT&E) to start a system along it's life-cycle. The system is available for acquisition once the development time has passed. The owning stakeholder can then make the decision to acquire a system for use (equivalent to system production). The system is deployed once the acquisition

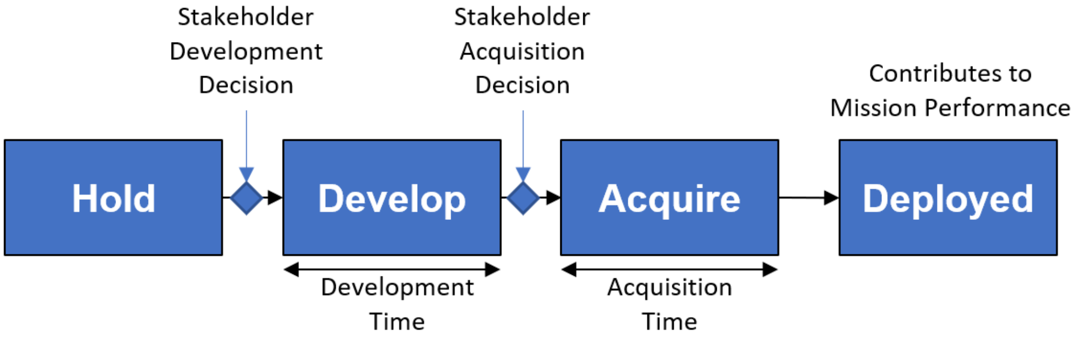


Figure 2.7: Intuitive Example: System Life-Cycle

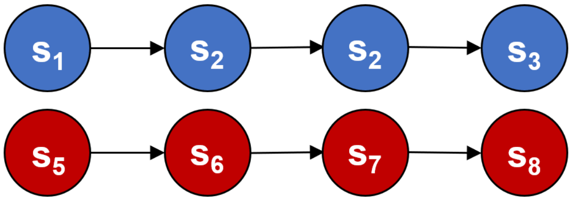


Figure 2.8: Intuitive Example: System Progression

time has passed. Additionally, Each system has a prerequisite system as defined in Table 2.6. The prerequisite system (columns) must be developed and ready for acquisition before the the system-of-interest (rows) is available for development. A graphical representation is presented in Figure 2.8. The initial conditions are set to have System 1 and System 5 already developed and ready for acquisition.

**System Performance:** Each deployed system can contribute to the scored mission level outcome. The individual performance ($p$) of each deployed system contributes in accordance with 2.4 to the individual stakeholder *mission power* ($q_h$).

$$q_h = \sum_{m \in M} n_m p_m^c \tag{2.1}$$

where $M$ is available systems for development, $n_m$ is the number of $m$ systems available, $p_m$ is the system power, and $c$ is a scaling constant.

Table 2.5: Intuitive Example Problem: System Definition

| | **Development Time (time steps)** | **Acquisition Time (time steps)** | **Mean Performance** |
|---|---|---|---|
| **System 1** | 4 | 2 | 3 |
| **System 2** | 4 | 2 | 9 |
| **System 3** | 4 | 2 | 27 |
| **System 4** | 4 | 2 | 81 |
| **System 5** | 4 | 2 | -3 |
| **System 6** | 4 | 2 | -9 |
| **System 7** | 4 | 2 | -27 |
| **System 8** | 4 | 2 | -81 |

**System Definition:** Each system is defined by a development time, acquisition time, and performance (Table 2.5).

**Stakeholder Decisions (Actions):** Each Stakeholder has a choice to develop a new system or acquire a previously developed system. Only a single system can be under development or acquisition at a given time. A stakeholder can choose a single development or acquisition when no development or acquisition is in progress. The development and acquisition decision opportunities otherwise follow what is outlined in the system life-cycle (Figure 2.7).

**Stakeholder Utility:** The example problem is defined by a single mission. All systems contribute to a single mission with no reallocation of assets to alternate missions. Each system contributes to a single stakeholder's *mission power*. The mission utility of Stakeholder

Table 2.6: System Progression Matrix

| | **System 1** | **System 2** | **System 3** | **System 4** | **System 5** | **System 6** | **System 7** | **System 8** |
|---|---|---|---|---|---|---|---|---|
| **System 1** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **System 2** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **System 3** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **System 4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **System 5** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **System 6** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **System 7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **System 8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 2.7: State 1 Stakeholder Decision Matrix

| | | **Player 2** | |
|---|---|---|---|
| | | **Acquire S5** | **Develop S6** |
| **Player 1** | **Acquire S1** | State 4 | State 3 |
| | **Develop S2** | State 3 | State 2 |

1 ($u_{h1}$) is the difference between the stakeholder *mission power* (Equation 2.5). The zero-sum definition of the game mean that the mission utility of Stakeholder 2 ($u_{h2}$) is opposite that of Stakeholder 1 (Equation 2.6).

$$u_{h1} = q_{h1} - q_{h2} \tag{2.2}$$

$$u_{h2} = -u_{h1} \tag{2.3}$$

$$q_h = \sum_{m \in M} n_m p_m^c \tag{2.4}$$

$$u_{h1} = q_{h1} - q_{h2} \tag{2.5}$$

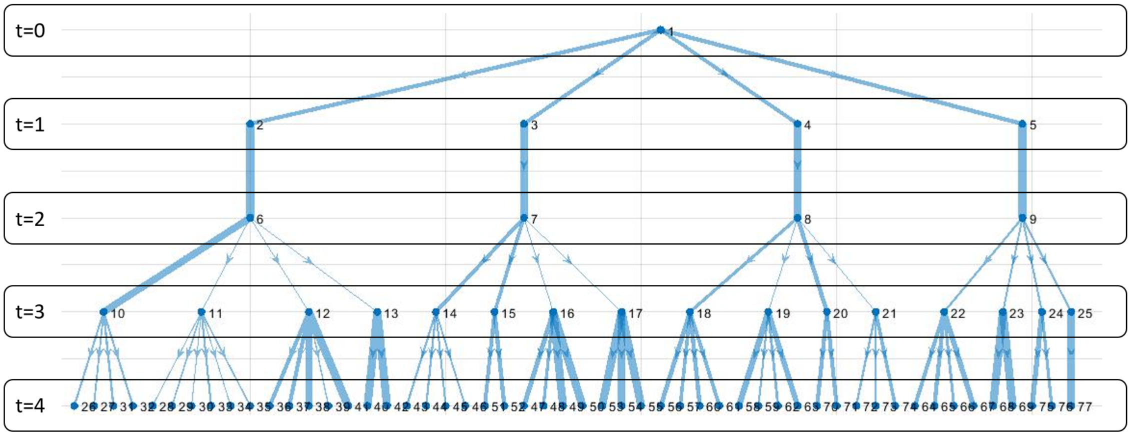


Figure 2.9: Stakeholder 1, Four Step, Multi-Stage Decision Space

$$u_{h2} = -u_{h1} \tag{2.6}$$

*Decision Trade Space Characterization*

First examining the initial state decision space will allow the full space to be better understood. The initial conditions of the example problem define System 1 and System 5 to be ready for acquisition. This means that the single step decision space is defined by Table 2.7. Each stakeholder has two options: acquire the only developed system or develop the next system with a higher performance. Note the development time is shorter than the acquisition time for each system (Table refintExpSysDef). Each set of stakeholder decisions leads to a specified resulting state (state numbers referenced . For single play through the multi-stage game, a decision table exists at each each time step with the specifics dependent on previous decisions made by each stakeholder. After every decisions, a new set of available systems for use or new decision choices are available.

The multi-stage decision space is characterized by two aspects at a given point in time: the available systems for deployment (the state) and the current available development/acquisition decisions (the actions). A graph can be developed showing the sequential state-to-state transitions, Figure 2.9. Each node represents a single state (State 1 being the

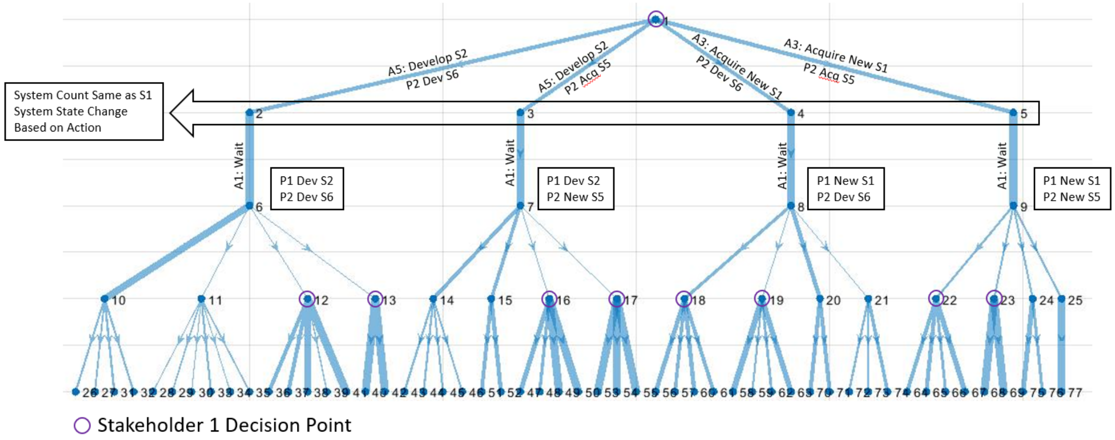
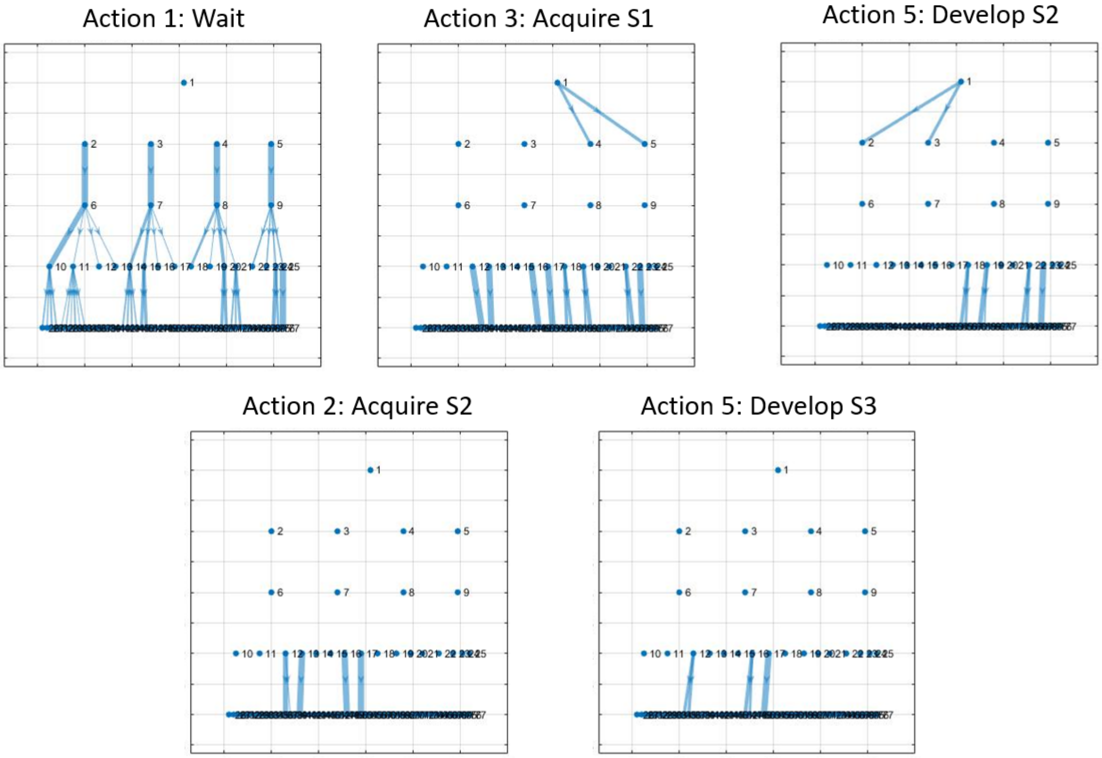


Figure 2.10: Stakeholder 1, Four Step, Multi-Stage Decision Space

initial state). Each vertex represents the results of a set of decisions by the stakeholders. The stakeholder decision are over laid in Figure 2.10.

A view of the decision space can be made for each stakeholder. The decision space, or available actions at given states, for Stakeholder 1 is presented in Figure 2.11. The problem setup dictates only a single system can be developed or acquired at a given time. The first action (*wait*) is the result of no action being available. The initial state actions of acquiring a System 1 or developing a System 2 are easily discerned (Actions 3 and 5 respectively). If System 2 is not developed during the initial state and Acquiring System 1 is selected, the opportunity to develop System 2 is available at time step 3. Alternatively, if System 2 development is selected, the opportunity to acquire System 2 is available at time step 3 along with the development of System 3.

The decision trade space also entails the resulting performance in addition to the state-action space described above. The performance is measured in *stakeholder mission utility* measured as described above. This utility is measured at every point in time and is the resulting performance feedback from all previous decisions. This consolidated measurement can be viewed as the stakeholders *reward* at any given time step. For Stakeholder 1, the reward for the entire decisions space previously defined is presented in Figure 2.12. Many simulation episodes were run to sample the decision space and develop the reward history.



Action 1: Wait

Action 3: Acquire S1

Action 5: Develop S2

Action 2: Acquire S2

Action 5: Develop S3

Figure 2.11: Stakeholder 1 Individual Action Graphs

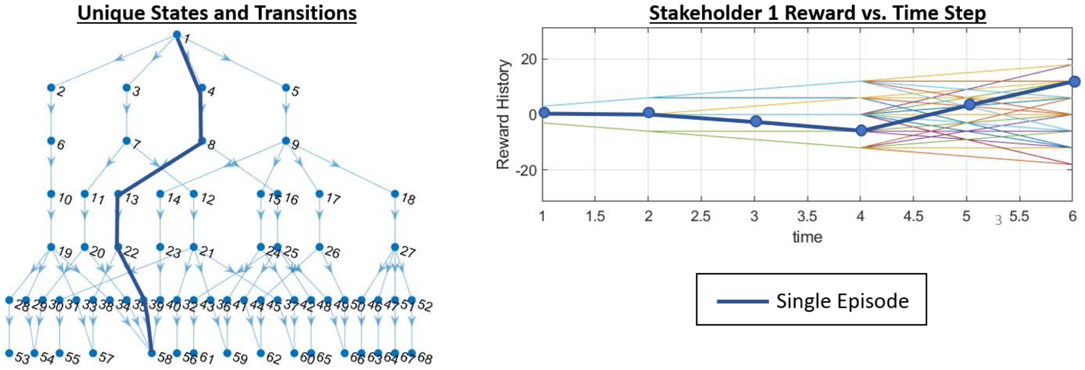


Figure 2.12: Example Decision Space Structure and Resulting Stakeholder Reward

A single episode, or path through the multi-stage game, is highlighted. The reward history is shown as a function of 6 time steps. The reward symmetry about the zero reward axis is a result of the zero-sum game definition.

### *Adding Complexity*

Thus far in the example problem development there has been no inclusion of uncertainty. The addition of uncertainty increases the complexity of the problem. Two sources of uncertainty can be used to demonstrate the impact (Table 6.9). First is temporal uncertainty and the second is performance uncertainty. Temporal uncertainty represents uncertainty in the RDT&E timeline or production timeline for a given system. It is represented as a uncertainty in the development or acquisition time for a given system. Performance uncertainty represents uncertainty in the resulting utility of a given system and it's impact on the a capability or a mission level outcome. The performance uncertainty is represented by by variation in system's contribution to stakeholder *mission power*.

An example of the impact of independently adding each uncertainty category, and then both, to the decision trade space is represented in Figure 2.13. Adding temporal uncertainty alone modifies the number of states but ultimately does not modify the bounds of potential stakeholder reward. The state increase is due to the added combination of available

Table 2.8: Intuitive Example Problem: System Definition with Uncertainty

| | **Mean Development Time (time steps)** | **Development Time Uncertainty (time steps, $3\sigma$)** | **Mean Acquisition Time (time steps)** | **Acquisition Time Uncertainty (time steps, $3\sigma$)** | **Mean Performance** | **Performance Uncertainty ($3\sigma$)** |
|---|---|---|---|---|---|---|
| **System 1** | 4 | 2 | 1 | 1 | 3 | 0.6 |
| **System 2** | 4 | 2 | 1 | 1 | 9 | 1.8 |
| **System 3** | 4 | 2 | 1 | 1 | 27 | 5.4 |
| **System 4** | 4 | 2 | 1 | 1 | 81 | 16.2 |
| **System 5** | 4 | 2 | 1 | 1 | -3 | 0.6 |
| **System 6** | 4 | 2 | 1 | 1 | -9 | 1.8 |
| **System 7** | 4 | 2 | 1 | 1 | -27 | 5.4 |
| **System 8** | 4 | 2 | 1 | 1 | -81 | 16.2 |

systems and system states over time due to the temporal uncertainty. Adding performance uncertainty alone modifies the structure of the reward without directly impacting the resulting state space. The reward structure change is due to variations along the otherwise deterministic reward paths.

The simple case outlined here demonstrates the significant impact of uncertainty on problem complexity. Additional complexity exist beyond uncertainty and what has been defined in the example problem. Remaining complexy includes allocating systems and resources to multiple missions, removal of systems from availability as they reach end of life, and budget limits impacting available decisions.

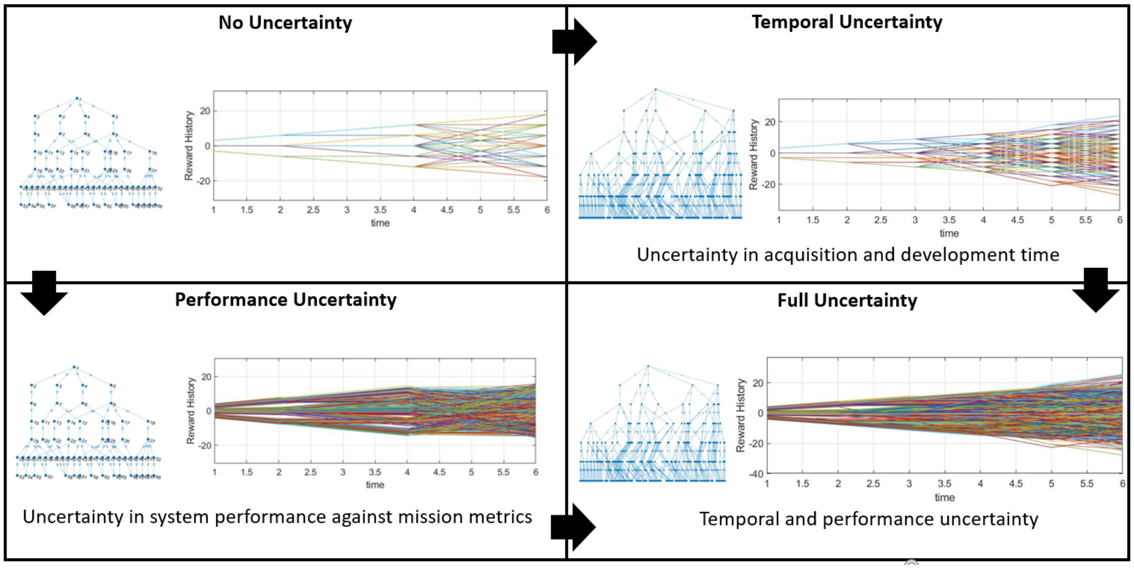


Figure 2.13: Adding Complexity

### 2.3.2 Methodology Requirements

Previously in this chapter, the SoS problem was defined and characterized. The force structure planning problem as defined in Chapter 1 can be directly mapped to the SoS problem described in this chapter. The force structure problem specifically has the following unique aspects that must be accounted for in any applicable methodology:

**Technology Investments** are made by individual stakeholders and represent strategic decisions. The payoff via technology insertion on a future system is subject to uncertainties in the time till system availability, the cost it takes to mature, and the final impact on system performance. The decisions to invest money at a given point in time have a slow response to long term system development and subsequently SoS performance. Often it could be a decade before a specific technology moves from a TRL 1-3 to a TRL 6+ maturity.

**Budget Constraints** are the primary factors that limit military stakeholders. It is the driving force behind the need to make recapitalization, technology refresh, and reallocation trade-offs. Budgets constrain both strategic (technology investments) and tactical (recap, tech refresh, and reallocate systems) decisions.

**Recapitalization** is the first of three primary defense stakeholder decisions. This involves investment in a new system which can be evolutionary or revolutionary. Evolutionary incremental development is the most predictable and usual development method. Moving away from one-for-one system replacements is key to fully evaluating higher cross-stovepipe-cutting trades.

**Technology Refresh** is the second of three primary defense stakeholder decisions and consists of utilizing existing systems while adding new capabilities via technology insertion into existing platforms. This option may take advantage of new available technologies enabling improved system performance and mission capabilities while at the same time minimizing the costs by extending the life of existing systems.

**Reallocation** represents the last of the primary defense stakeholder decisions addressed in this body of work. Reallocating systems between missions and objectives becomes the final options when budgets are constrained and technology insertion and recapitalization are not options but a certain level of mission utility is required. Reallocation goes beyond DOTmLPF-P solutions and results in moving assigned assets from one mission need to another. It can involve moving control of assets from one stakeholder to another with or without a two-way transaction.

**Multiple Stakeholders** can be interested in the same mission and the same available assets. It is crucial to capture the interest that a single stakeholder has in each given mission. Additionally, it is crucial to capture the control a stakeholder has over the development of constituent systems that, as a part of a SoS, provide mission level utility.

**Multiple Missions** need to be balanced during defense planning. The need to balance between missions that span stakeholders and span the portfolio of ultimate needs of the

United States. One example of this is balancing between ballistic missile defense, strike capabilities, and naval power projection in both technology development, system development, and system allocation.

**Uncertain Future Scenarios** must be considered with regard to the applicable red force structure or geographic engagement. The red force can be approached similarly to the blue force structure over time with the evaluation of either force structure at a given time a product of the opposing force. Inherently, there is a need to capture the uncertainty of the state of the scenario dictated by external forces in which the SoS will operate.

**Uncertain Future Budgets** can drive the particular policy or decision chain selected by stakeholders. In addition to budget constraints, taking into account the impacts of budget uncertainty over time (e.g. continuing resolutions, sequestration) is essential for any long-term planning. Exploration of the impacts due to risk posture and lack of knowledge can significantly improve the helpfulness of the strategic planning analysis.

The problem formulation leads to specific capabilities that need to be present in any method used to explore the planning and evolution of a military at the highest level, or called here the force structure level planning. From the above aspects of the general System of Systems evolution problem and the specific defense oriented SoS problem the following required aspects of a methodology have been identified:

**Multi-Stakeholder Decision Making:** At each decision cycle there are multiple stakeholders that make decisions separately and cooperatively. Each stakeholder is attempting to maximize its return on investment against the missions and objectives. There is a significant decision trade space that is generated from individual stakeholder decisions to allocate financial and system resources and from combining the individual decision spaces in order to develop a single step action outcome.

**Evolutionary Feedback Loop:** Capturing just the multi-stakeholder decision making is not enough. The feedback loop is key to capture the evolutionary loop described in previous Figures 2.6 and 2.5. Capturing the feedback loop entails capturing the long term impacts of both the strategic technology investment decisions and the tactical system decisions. It also includes the impacts due to the change in the environment both internal and external.

**Technology and System Development:** It is important to track the ongoing development of technology based on TRL, capture the anticipated cost, and anticipate the final capability after implementation. Additionally, the system acquisition life-cycle must be included. The choice to initiate system development, refresh, or retirement is crucial to representing the decision trade-space along with how the decisions ultimately manifest themselves in a mission capability when new assets with new technologies become available.

**Capturing of Uncertainty:** When attempting to provide a road map or plan for strategic decision making, it is imperative to consider the impact of uncertainty from all sources that will affect planning. This allows for the testing and understanding of how robust or flexible stakeholder decisions are with respect to potential future scenarios.

**Architecture Representation and Evaluation:** The ability to first define and describe an architecture is essential to developing alternatives and quantitatively evaluating them. Evaluating a given architecture against a specific objective is essential to determining the capability of a SoS once alternatives have been developed.

**Environment and Scenario Representation:** Capturing the external environment that is not impacted by the SoS itself is key to defense planning. The external environment constitutes scenarios that will remain static with regards to the evolution of the defense SoS.

**Multiple Mission Objectives:** When looking at multiple SoS (as the force level strategic planning problem looks at) it is key to understand the impact any changes will have to multiple missions. Mission utility is the measure used for stakeholders to evaluate the current success of a given state. The trade between reallocating systems from one to another as well as the ability to capture the impact of prioritizing the allocation of resources to one mission over another is needed when looking at a military force structure as a whole. Being able to analyze multiple missions in a method enables the necessary trades between resource allocation and mission capability.

**Defined SoS Engineering Reference Process:** A method addressing the long-term evolution of a SoS needs to be based on a defined SoSE process and align to a defined SoS life cycle. A clear process of adding and evaluating new systems is essential to outlining and developing a model.

# CHAPTER 3

# BACKGROUND, OBSERVATIONS, AND RESEARCH OBJECTIVE

System of Systems Engineering (SoSE) is no longer in its infancy, and the definitions and taxonomies above are familiar ideas. Even though SoSE has been explored theoretically and fully characterized, gaps still exist regarding quantitative methods for directed SoS planning.

This chapter explores current analysis and planning methods potentially applicable to the strategic force planning problem from the viewpoint of a stakeholder within the SoS construct. The following planning methods may or may not be derived specifically for a SoS in mind. Applicable contributions and gaps for each method will be identified and compared to the specific needs outlined in Chapter 2. The identified gaps yield the Research Objective of this work (Section 3.10) and lead to the posed Research Questions outlined in Chapter 4.

## 3.1 System of System Planning Models

A key need demonstrated for the stakeholder planning problem is to adhere to a defined SoSE reference process. There has been a significant body of work addressing not just the system development process (SE Vee) but also the SoS development process (Trapeze and Wave Models).

### 3.1.1 Systems Engineering Vee

The Systems Engineering Vee diagram has become ubiquitous with Systems Engineering itself. It is the foundation that allows complicated systems to be appropriately managed from conception through disposal. Typically, the process is used during the development and manufacturing of specific systems. The method is clearly defined, though the specifics

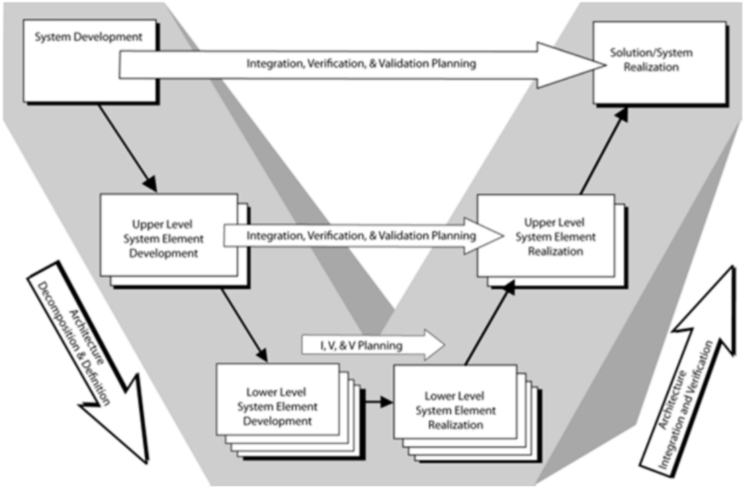

System Development
Integration, Verification, & Validation Planning
Solution/System Realization
Upper Level System Element Development
Integration, Verification, & Validation Planning
Upper Level System Element Realization
I, V, & V Planning
Architecture Decomposition & Definition
Lower Level System Element Development
Lower Level System Element Realization
Architecture Integration and Verification

Figure 3.1: INCOSE General Systems Engineering Vee Diagram [51]

of implementation may vary slightly from organization to organization. It provides a clear process that has been honed repeatedly over time [51]. The SE Vee process works very well for developing well defined systems against well defined static requirements as seen with Complicated Systems.

### 3.1.2 Trapeze and Wave Model

As previously established, a large net-centric or inter-connected military is an example of a System of Systems. As a military plans it's future force structure it is implicitly planning the future state of a System of Systems. With this observation, the Office of the Under Secretary of Defense for Acquisition, Technology and Logistics developed the Systems Engineering Guide for Systems of Systems formalizing a Systems Engineering process to use to address the continuous planning, development, and deployment of a formalized System of Systems as depicted in Figure 3.2. [52, 53, 54]

The Trapeze model outlines the seven core elements of SoS Systems Engineering. Translating Capabilities addresses taking a general capability expected to be provided by the SoS of interest and translating it into inputs to other functions (SoS requirements, system requirements, monitoring, and assessing). The idea of evolution is key to the Trapeze

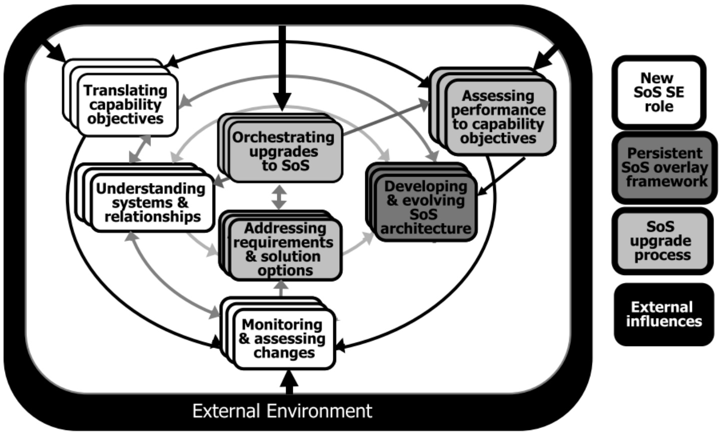

Translating capability objectives
Orchestrating upgrades to SoS
Assessing performance to capability objectives
Understanding systems & relationships
Developing & evolving SoS architecture
Addressing requirements & solution options
Monitoring & assessing changes
External Environment
New SoS SE role
Persistent SoS overlay framework
SoS upgrade process
External influences

Figure 3.2: System of System Trapeze Model [52]

model as seen in the described continuous monitoring and assessing of performance while understanding the relationships between systems. Ultimately, understanding the SoS allows new architectures to be developed against evolving requirements enabling the orchestration of upgrades [52]. The key concepts of the model are the evolutionary aspect of assessing, monitoring, evaluating, and upgrading with the continually evolving environment. Any analytical method addressing the planning or evolution of a SoS should incorporate a similar process. Dahmann expanded on the Trapeze Model to create the Wave model as depicted in Figure 3.3. Similarly, it depicts the evolution of a SoS over time through continuous monitoring, assessing, evaluating, and upgrading. [49, 50]

The Trapeze Model is a process that was developed from analyzing the use of SoS within the United States military. It is consolidated and released by a central authority. The developed process helps to standardize the way in which SoS evolution is viewed. Any work that addresses the planning and evolution of a SoS should address where it falls and fits within this process. The process and framework do not in and of itself determine how to evolve a SoS, rather it provides a method to follow. Similar to the SE Vee discussed above, the method dictates a process but does not specify how to achieve each step. Like tailoring the SE Vee to any system level development and manufacturing with specific implementations, the Trapeze should be used to guide the development of any SoS planning

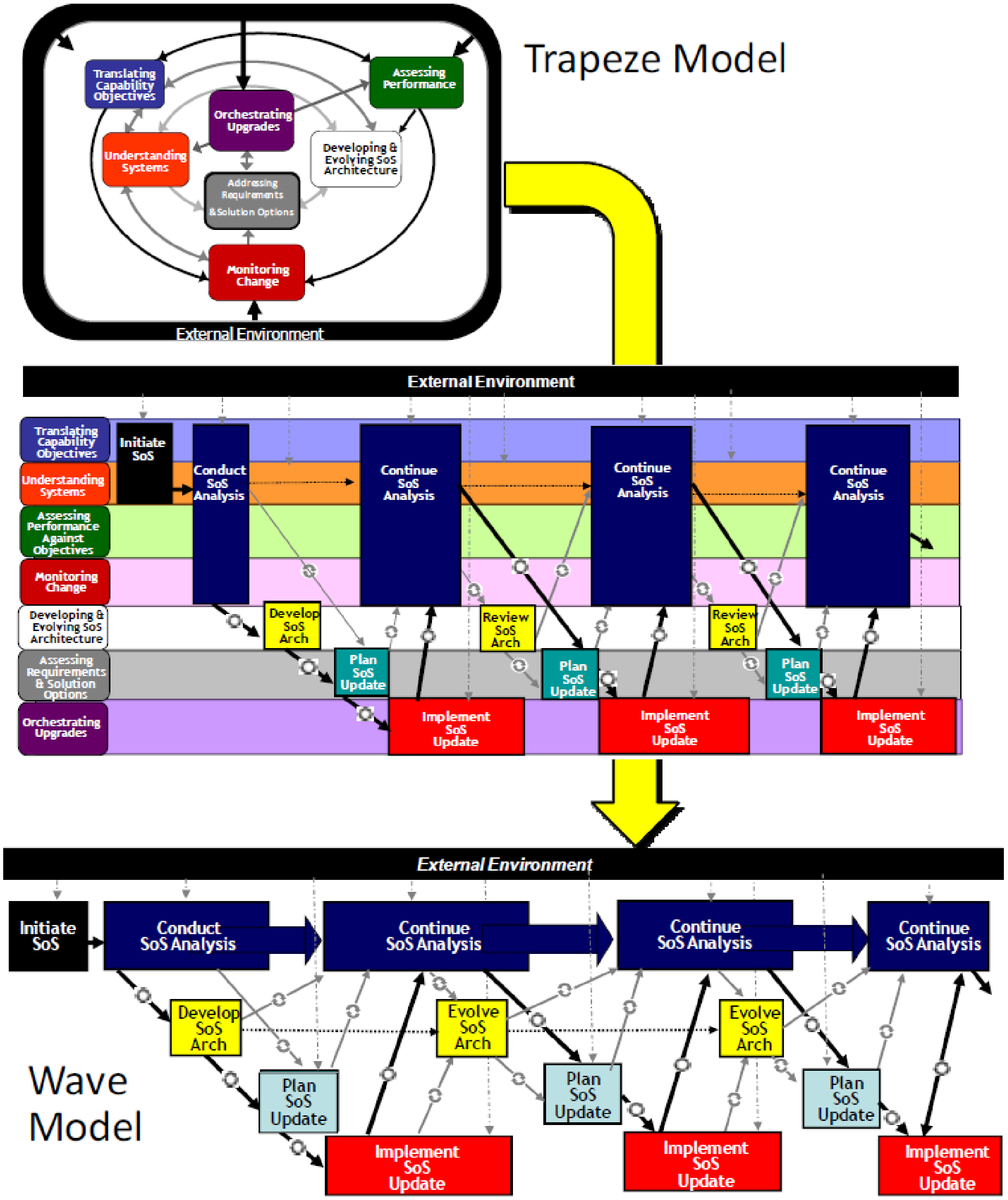


Figure 3.3: Relationship Between the Trapeze and Wave Models [49]

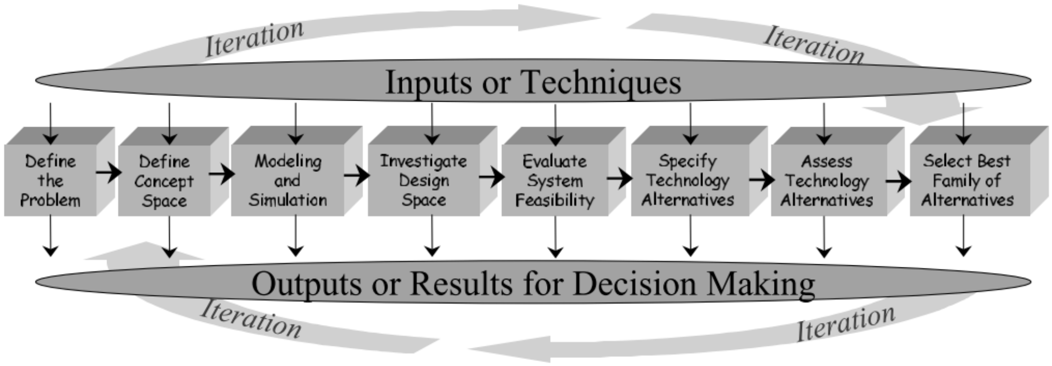

Iteration
Iteration
Inputs or Techniques
Define the Problem
Define Concept Space
Modeling and Simulation
Investigate Design Space
Evaluate System Feasibility
Specify Technology Alternatives
Assess Technology Alternatives
Select Best Family of Alternatives
Outputs or Results for Decision Making
Iteration
Iteration

Figure 3.4: Technology Identification Evaluation and Selection Model [57]

and evolution method.

## 3.2 Technology Evaluation

Technology evaluation is used to determine the most promising areas of investment. Technology evaluation methods are applied early in the technology development life-cycle based on potential future impact. A number of methods have been developed to handle the relative rating and evaluation of new technologies. Each method below addresses the problem from a different perspective.

### 3.2.1 Technology Identification Evaluation and Selection Method

The Technology Identification Evaluation and Selection (TIES) method is well established and outlines a systematic process for evaluating future technologies to be added to a system.[55] The TIES approach provides a structured method to explore the concept, design, and technology space. Combined with a unified trade-off environment (UTE) the design, technology, and requirements spaces can be explored in near real time [56]. The explicit TIES process steps are described in 3.4 with more detail in 3.5.

TIES uses K-factors to represent the impact of specific technologies on an existing baseline design. From alternative selection, to baseline design, through technology evaluation, each step optimizes the results. The process is explicit and repeatable while providing a traceable and quantitative method. Uncertainty development can be included utilizing robust design techniques with distributions assigned to technology impacts as a function of TRL.

If identifying a system-level alternative is not time consuming, the resulting process is manageable. When expanded to a SoS (not just a single system) the churning on architecture alternatives increases the needed evaluation time. Additionally, with SoS, it is key to consider the non-linearly combined impact of systems and technology.

### 3.2.2 Technology Impact Forecasting Method

The objective of the Technology Impact Forecasting (TIF) method is to identify significant impact areas for future technologies. The TIF method allows an exploration of potential futures not constrained by the current technology state or expected progression. It uses independent K-Factors combined with the Response Surface Methodology (RSM) to project the impact of potential technologies on a baseline system. [58, 59] It does not capture the dependencies between impacts of technologies, rather it captures the individual impacts a technology could have. Another assumption is having knowledge of intermediate variables (system level technology impacts) to which K-factors are applied. The TIF process is detailed in Figure 3.5. [55]

### 3.2.3 Capability Based System Technology Evaluation

Capability Based System-of-System Technology Evaluation methods have the same goal as the system technology evaluation methods (such as TIES and TIF), which is to identify the most promising technologies for future development. For the capability-based-methods, the performance of more than a single system is used to evaluate technology. [61]

Biltgen developed a comprehensive process to evaluate technologies against their mis-

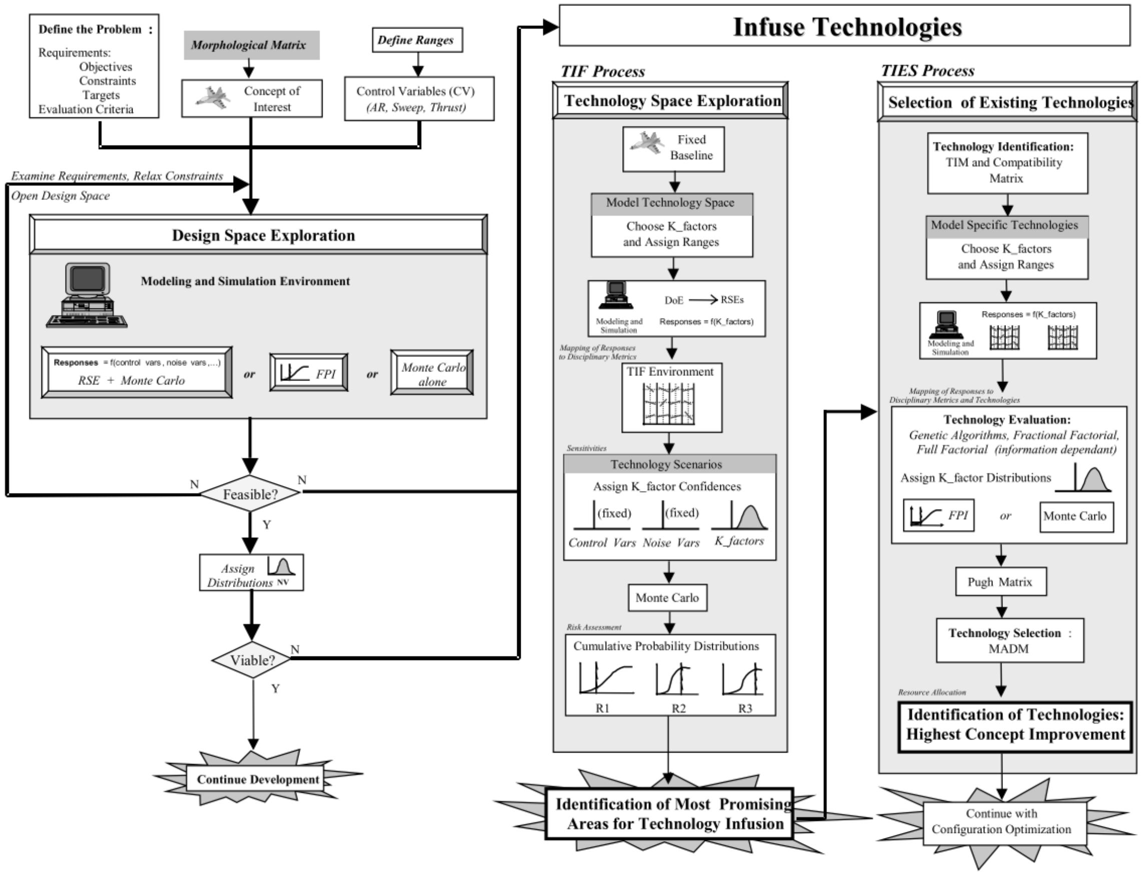

Define the Problem :
Requirements:
Objectives
Constraints
Targets
Evaluation Criteria
Morphological Matrix
Concept of Interest
Define Ranges
Control Variables (CV)
(AR, Sweep, Thrust)
Examine Requirements, Relax Constraints
Open Design Space
Design Space Exploration
Modeling and Simulation Environment
Responses = f(control vars , noise vars ,...)
RSE + Monte Carlo
or
FPI
or
Monte Carlo alone
N
Feasible?
N
Y
Assign Distributions NV
Viable?
N
Y
Continue Development
Infuse Technologies
TIF Process
Technology Space Exploration
Fixed Baseline
Model Technology Space
Choose K_factors and Assign Ranges
DoE → RSEs
Modeling and Simulation
Responses = f(K_factors)
Mapping of Responses to Disciplinary Metrics
TIF Environment
Sensitivities
Technology Scenarios
Assign K_factor Confidences
(fixed)
(fixed)
Control Vars
Noise Vars
K_factors
Monte Carlo
Risk Assessment
Cumulative Probability Distributions
R1
R2
R3
Identification of Most Promising Areas for Technology Infusion
TIES Process
Selection of Existing Technologies
Technology Identification:
TIM and Compatibility Matrix
Model Specific Technologies
Choose K_factors and Assign Ranges
Responses = f(K_factors)
Modeling and Simulation
Mapping of Responses to Disciplinary Metrics and Technologies
Technology Evaluation:
Genetic Algorithms, Fractional Factorial, Full Factorial (information dependant)
Assign K_factor Distributions
FPI
or
Monte Carlo
Pugh Matrix
Technology Selection :
MADM
Resource Allocation
Identification of Technologies:
Highest Concept Improvement
Continue with Configuration Optimization

Figure 3.5: Technology Impact Forecasting Model [60]

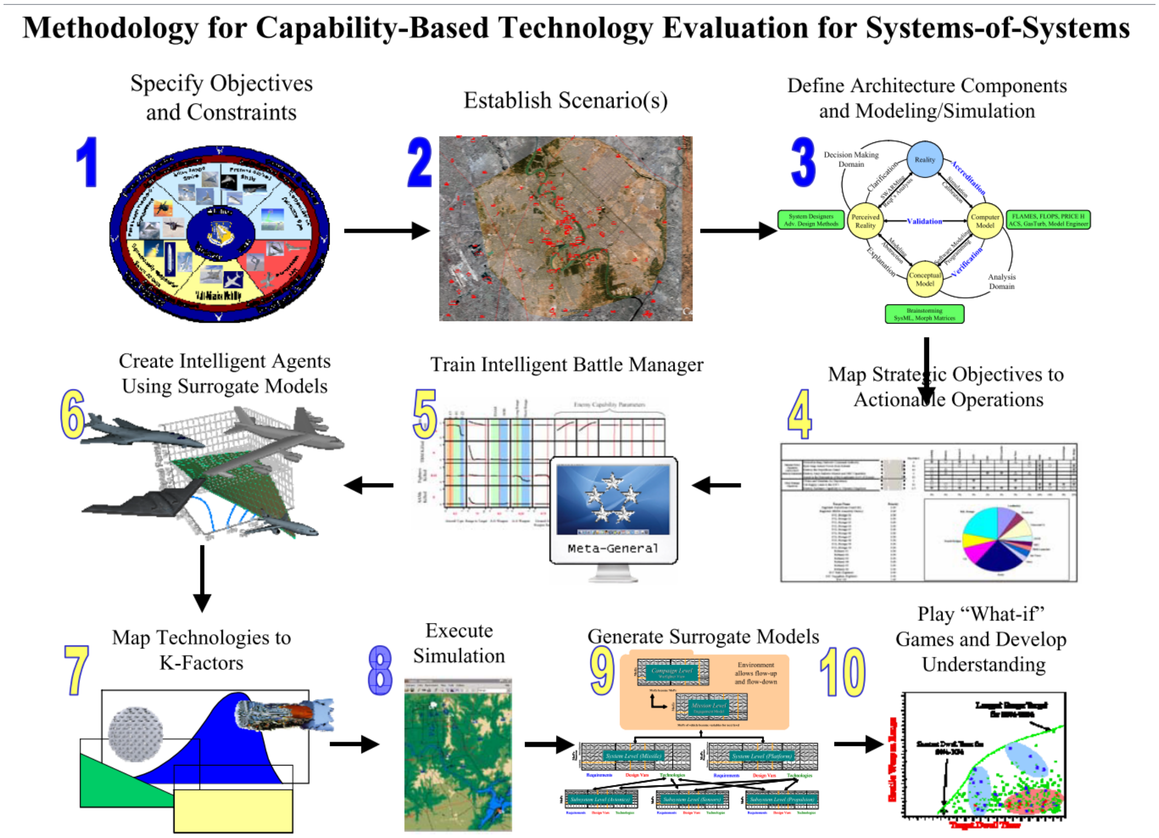

Methodology for Capability-Based Technology Evaluation for Systems-of-Systems
1
Specify Objectives and Constraints
2
Establish Scenario(s)
3
Define Architecture Components and Modeling/Simulation
Decision Making Domain
Reality
Perceived Reality
Computer Model
Conceptual Model
Validation
Verification
Accreditation
Analysis Domain
System Designers Adv. Design Methods
FLAMES, FLOPS, PRICE H ACS, GasTurb, Model Engineer
Brainstorming SysML, Morph Matrices
4
Map Strategic Objectives to Actionable Operations
5
Train Intelligent Battle Manager
Enemy Capability Parameters
Meta-General
6
Create Intelligent Agents Using Surrogate Models
7
Map Technologies to K-Factors
8
Execute Simulation
9
Generate Surrogate Models
Campaign Level
Mission Level
Environment allows flow-up and flow-down
System Level (Missile)
System Level (Platform)
Requirements
Design Vars
Technologies
10
Play "What-if" Games and Develop Understanding
Longest Range Target
Shortest Dwell Time
Target Dwell Time

Figure 3.6: Capability Based System Technology Evaluation Methodology [61]

sion level capability (Figure 3.6). The method addresses the trade-off between technologies using mission capability rather than system performance. System tactics are used in the evaluation of technologies at the mission level (e.g. training a battle manager). The inclusion of tactics and behavior is often not addressed in system and technology evaluation and plays a large role in the resulting synergies between platforms. [61]

The scalability of the method largely depends on the complexity of the Modeling & Simulation (M&S) but often proves difficult. To expand from a single Analysis of Alternatives (AoA) to the challenge of evaluating the evolution of a SoS proves difficult.

## 3.3 System of System Architecture Development Methods

As complexity increases, it becomes important to use a defined method to represent and describe architectures. A number of efforts have been put forward to grow traditional system architecture modeling to accommodate SoSE needs.

### 3.3.1 Representing Architectures

The idea of architecting a system, and even a SoS, is not a new concept. The use of standard descriptive modeling techniques has become common. Descriptive modeling is the practice of capturing the structure and behavior of a System of Interest (SOI) whether it is a software architecture, system architecture, or a SoS architecture [32]. Common descriptive modeling techniques used to describe system and SoS level architectures include UML [62], SysML [63, 64, 65], and DoDAF [66, 67].

### 3.3.2 Descriptive Architecture Modeling

Developing descriptive models allows for the documenting and communication of architectures. It does not explicitly provide quantitative measurements. The ARCHITECT method developed by Griendling looks to generate and evaluate architectures using executable descriptive models. This provides the capability to generate architecture alternatives and evaluate them against mission requirements [68]. Others have developed similar methods or have used executable models for SoS evaluation all based on common descriptive modeling techniques [40, 69, 70, 71, 72, 73].

Software architecting has long been a field of study and application. Many common patterns have been identified when architecting software [74, 75]. Kalawsky expands this idea to the SoS problem. A process for developing and applying the patterns through the evolution of a SoS was defined. Patterns are developed and applied as the SoS is evolved. [76]

The Comprehensive Modelling for Advanced Systems of Systems (COMPASS) is a European consortium focused on developing Model Based Systems Engineering techniques for SoS [77, 78, 79, 80, 81, 82]. The goal is to enable the architecture development and evolution of large complex systems.

## 3.4 Optimal, Robust, and Flexible Design

Historically, design has focused on producing an optimal result with an eye toward maximizing performance within a design space through traditional optimization techniques (Lagrangian, gradient decent, stochastic optimization, genetic algorithms etc.) even with respect to SoS. Two separate additional design approaches address uncertainty in performance and requirements. The first, Robust Design, attempts to find a system design that immunizes its performance by ensuring that variations in requirements or environment have little impact on the performance [83]. The second method, Flexible Design, also works to address the impact of changing environments and requirements. Instead of immunizing against uncertainty, flexibility is "the capability to easily modify a system after it has been fielded in response to a changing environment or changing requirements". [84] LaFluer expands this technique to look at the flexibility in planning for future space missions [85] with small state and decision spaces. A comparison of each technique can be seen in Figure 3.7. The concepts of robust and felixible design are key when addressing the military strategic planning problem though direct translations can be more difficult.

## 3.5 Scenario and Environment Representation

Scenario and environment representation defines the external inputs outside of a SoS and its stakeholders. Scenario-based planning has become key to strategic planning methods and provides a macro-level and backcasting (future to present) view [86]. A classic example is

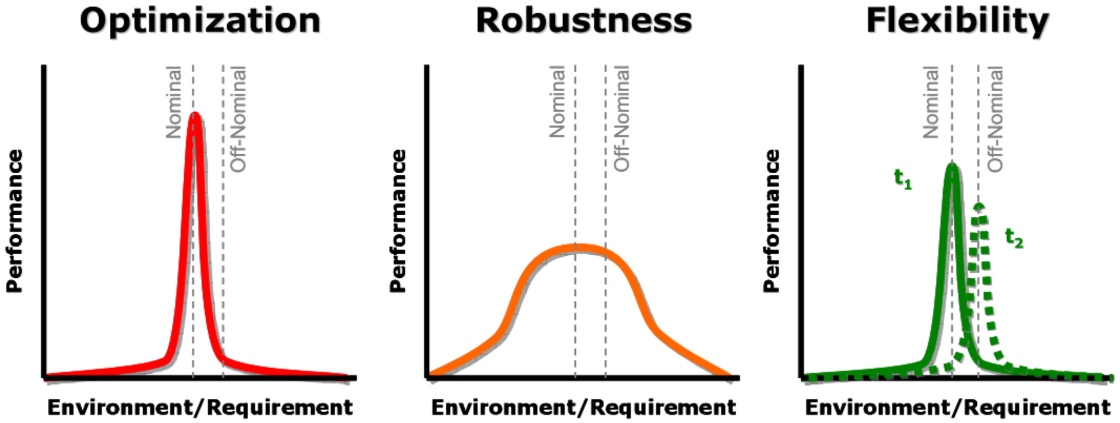


Figure 3.7: Robust Design vs. Flexible Design [84]

Political, Economic, Social-Cultural and Technology (PEST) analysis which is a qualitative method of exploring high level drivers above and beyond direct influence [87]. The Strategy Optimization for the Allocation of Resources (SOAR) developed by Raczynski is a formal top-down process for strategic planning utilizing MODM/MADM techniques. The top starts with world scenarios and an organizational vision which that is derived from lower level requirements and needs. The process is static and based on Subject Matter Expert (SME) input at each level to help derive requirements [88]. This approach ties high level needs and breaks them down in a more traditional SE process. In defense planning, defining and exploring many future scenario is key [89]. Defining and exploring future scenarios and environments with a one way flow into the area of interest, in this case a SoS, is a mature practice with long term applications inside and outside the defense community.

### 3.6 Value-Driven Design and Cost-Capability Analysis

The concept of Value-Driven Design (VDD) works to quantify the impact of a designers preferences into the design process via value maximization [90]. Value-Driven Design works to provide an objective, repeatable, and transparent method [91]:

1. Objective means that decisions should not be opinionated. Instead, every design decision should be based entirely on facts, test results, and analyses.

2. Repeatable means that, given the same facts, test results, and analyses, the same decision will always result, even if the decision is made by a different designer or a different design team.

3. Transparent means that the design process should easily yield the reasons for the decision. That is, the process should not be a black box into which data are entered and then a result is generated. Instead, a clear understandable method is required in which the engineer and everyone else can observe and critique the process.

Value-Driven Design works to combine both traditional quantitative and qualitative metrics at all levels into the design process. It combines traditional attribute based design, capability based design, business/economic evaluations. [92] It provides a point of convergence of economics, optimization, and systems engineering. [91]

VDD allows for trade-offs to be made at multiple levels of a system (component to mission) as well as cost vs. capability trades to evaluated. It provides a framework to view and execute multi-objective design. A key concept used is the value of a system expressed conceptual in Equation 3.1.

$$Value = \frac{Performance}{Cost} \tag{3.1}$$

The performance can be evaluated at any level equated to overall system effectiveness or a specific key performance parameter of a subsystem. The cost can represent the Recurring Engineering (RE) of a unit or the Life-Cycle Cost (LCC) of a system. The net result is the concept of the overall value of a given design. An extension looking at Cost as an Independent Variable (CAIV) and Cost-Capability Analysis has become common [93]. The concept looks at evaluating the Pareto efficient frontier of cost versus capability (or Value) as part of the design process. A conceptual example is presented in Figure 3.8.

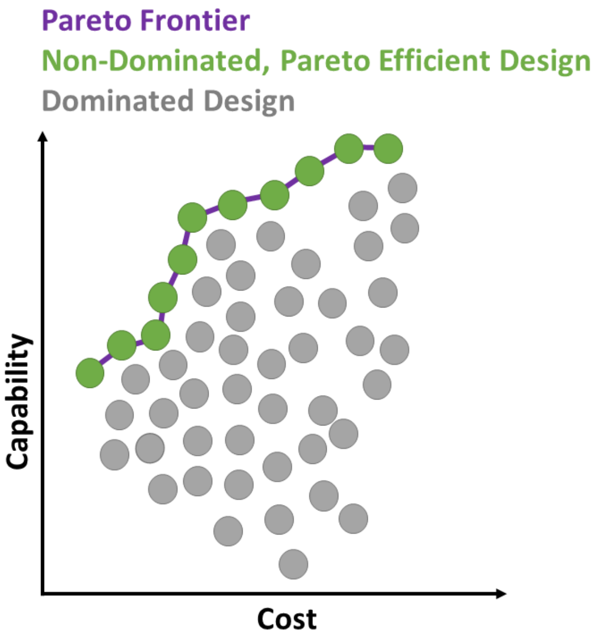


Figure 3.8: Cost-Capability Analysis (CCA) Pareto Efficient Frontier Example

## 3.7 System of Systems Analytic Workbench

A recent approach to analyzing and planning within the Systems of Systems community comes from the DoD's Systems Engineering Research Center (SERC) and focuses on the development of an Analytical Workbench toolset. The fundamental idea behind its development is that no single analytical approach can be applied to all SoS analysis. The Analytic Workbench is a collect methods that can be selected and applied as needed by a SoS analyst. The collected methods are designed to be domain-agnostic and inter-operable in order to simplify the barrier of entry for analysts. A number of examples are available demonstrating the use of each collected method, including combined use, to analyze SoS problems.

The SoS Analytical Workbench (Figure 3.9) fits within a process developed around the Wave Model previously introduced (Section 3.1.2). At each step, whether in review, update or implement, the appropriate tools can be selected to help guide SoS decision making. Higher fidelity M&S methods can be used to evaluate any decisions made. Results from

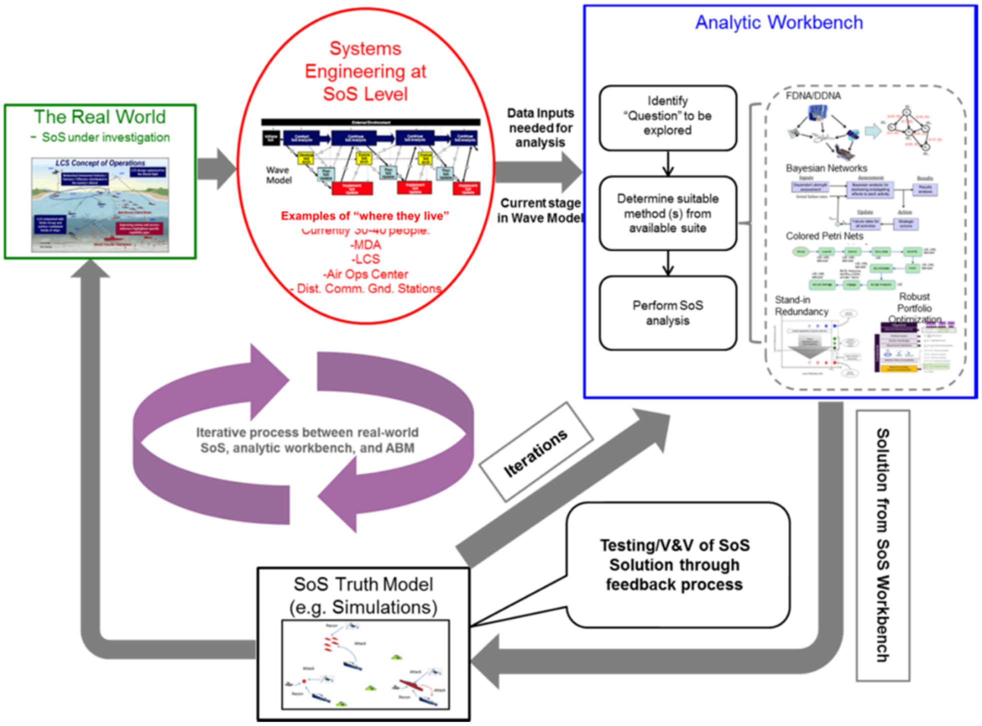


Figure 3.9: SoS Analytic Workbench

the real world can be fed back once a decision is made and implemented. This higher level, generic process can be applied to almost any continued system development or to the evolution of a SoS. The discriminators of this method lie in how the Analytical Workbench will be used, a process in and of itself. [94, 95, 96]

The process starts with archetypal questions which are used to guide the method selection from the SoS Analytic Workbench toolset ("what do you want to know?"). Questions are mapped to the available methods (FDNA/DDNA, Bayesian Networks, Robust Portfolio Optimization, Colored Petri Nets, Stand-in Redundancy). Specific inputs are defined and used for each of these methods. The analysis uses a simplified model of the SoS and is then V&V'd using a 'Truth Model' or, in usual methods, an Agent Based Model. Many of the tools and methods within the process that have been collected [97] are similar to those previously described [98, 99, 95, 100].

Current significant components of the SoS Analytic Workbench include both standard and new techniques:

**Robust Mean Variance Portfolio Optimization** is used to balance the rewards of acquisition with the risks of development time. It uses a defined SoS hierarchical network description which is common across the workbench's techniques. Each node in the graph has capabilities (payoffs) and requirements (costs). The SoS level performance is the investment portfolio performance and the risks are developed from common SoS risk postures. The risk and reward formulation based on common financial portfolio optimization also takes into account the compatibility of systems and the satisfaction of system requirements. [94, 101, 95, 100]

**Functional Dependency Network Analysis and Developmental Dependency Network Analysis** are both based on graph theory and network analysis but are used to measure two '-ilities' [100] of a SoS. FDNA is used to represent dependencies predecessor and successor systems or capabilities (as capabilities may not always entail a single system) using an acyclic graph. Links represent operational or developmental dependencies through weightings Strength of Dependency (SOD) and the Criticality of Dependency (COD) between nodes. FDNA allows a look into what happens when failure in the SoS occur. The models can be made stochastic through probability distributions linked to failures and the impact on the overall SoS performance. Development Dependency Network Analysis (DDNA) is used to analyze the effects of development delays, in a similar manner to SOD and COD, except the links are tied to a PERT network and don't represent an operational scenario. Inputs include beginning and end time for each system and dependencies. Similar network methods can be used to analyze an operational SoS (FDNA) or the development of a SoS (DDNA). [99, 100, 95, 102] PERT analysis is commonly used in industry for schedule analysis.

**Bayesian Networks** (BN) are used to analyze the operational domain (as opposed to the development domain) utilizing Directed Acyclic Graphs (DAG) with probability weightings on edges. The BNs are used to analyze the impact of operational failures in the Analytic Workbench. This allows the resilience of the SoS to be evaluated. [99, 98, 95]

**Stand-In Redundancy** is an evaluation method for a SoS overtime using two axis of measurement: Level of Performance and Level of Reliability. Level of Performance is a measure of capability of the SoS such as a mission level metric. The Level of Reliability measures the instantaneous probability of failure (e.g. gradually reduces over time once deployed due to probability of failure). [103, 104, 94, 98, 105]

**Approximate Dynamic Programming** is a common non-linear programming method which is used in the Analytic Workbench to "introduce computational strategies that can provide objective, multi-stage decisions that balance impacts of near-term and long-term SoS architectural decisions" made by stakeholders [98, 102]. The basic idea is to use ADP to evaluate an architectural decision tree where decisions are made based on a policy. [106, 97, 95, 100, 98] The decision tree formulation is not fully made but, if created, ADP would be an acceptable technique for finding optimum solutions for a small trade-space of future possibilities unconstrained by stakeholder decisions. Current implementations for SoS architecture development do not address uncertainty.

**Petri Nets** are a common Discrete Event Simulation (DES) modeling technique, especially in failure analysis. Petri-Nets are used by the Analytic Workbench to quickly evaluate a SoS performance without a complex simulation. [70, 99, 98]

**Approximate Dynamic Programming and Transfer Contract** : Fang introduces the idea of tackling the curse of dimensionality that grows with analyzing sequential decision making for SoS stakeholders. Additionally, the concept of a transfer contract is used to exchange payment for shared resources. A decentralized planning method is used based on the transfer contract approach. Uncertainty is added by defining system capability at a given time as a probability distribution with repeat runs to sample. [106, 102]

The work surrounding the SoS Analytic Workbench has progressed the areas of SoS architecture representation, evaluation, and evolution. It has begun to address the evaluation of SoS over time and the impact of stakeholder decisions over time [107, 102, 108]. A key missing aspect is decision making under extreme uncertainty.

## 3.8 Adaptive SoS Architecture Evolution

A key component of the defense planning problem is generating and evolving SoS architecture from a current state to a future state. Agarwal developed a evolutionary methodology using meta-architectures to represent architectures and enable genetic algorithms to optimize against an architecture evaluation process [109, 110, 111, 112, 113, 114]. Work has been done using dynamic programming and ADP to solve for the optimum potential future architecture [102]. The Cognitive Evolutionary Computation (CEC) for SoS architectures focused on optimizing the evolution of a SoS using the CEC algorithm modeling divergent, convergent, and long term memory in combination with architecture definition and evaluation methods [115]. These are examples of recent developments in architecture optimization.

The field of architecture optimization for a single architecture given a definition and an evaluation method is quickly reaching maturity. Each one of these methods focuses on finding an optimum architecture for a future state. This works well for directed SoS with centralized control. An explicit assumption in the methods is the direct control of the resulting SoS by a single stakeholder. None directly represent multiple-stakeholders with multiple-missions over time. Each method works in a non-stochastic environment and, at best, uses a simple mean return to value given architectures.

### 3.9 Summary of Observations and Gaps

Throughout Chapter 3, various techniques that can be applied to the force level trade problem were presented and evaluated. The following are SoS disciplines and methods that, based on the literature search presented in this chapter, are individually mature:

**Architecture Representation, Evaluation, and Optimization:** There has been significant work in developing methods to represent, evaluate, and optimize SoS. Representation examples using existing architecture description modeling methods include DoDAF, SysML, and UML. Efforts exist to expand these methods to enable executable architectures. The executable architecture evaluations use varying levels of MS&A including Discrete Event Simulations and Agent Based Modeling. General optimization methods include genetic algorithms, simulated annealing, Lagrange, etc. These efforts have been focused on developing static or near-static representations and not on dynamic evolutionary capability.

**Addressing SoS as a Concept (definition and taxonomies)** Describing and categorizing SoS has been thoroughly explored and defined. The characteristics put forth by Maier [47, 43], DeLaurentis [45], and Gorod [46] (along with many others) have fully exhausted the characterization of a SoS. Taxonomies have been equally explored and help further help categorize SoS based on their characteristics [47, 48, 45, 46].

**Defined SoS Process and Life-Cycle** Starting with Maier's work in the late 90's [47] to the current DoD SoSE Guide [52] to including work by Dahmann [116] there has been maturation of a common SoS definition and a common SoS life cycle. The Wave Model developed by Dahmann [50] builds on the DoD SoSE Guide's Trapeze Model [52] and provides the commonly accepted SoS life cycle. The maturity of the life-cycle description allows for a common reference when addressing problems that span a single development cycle of a SoS. The accepted Wave Model is used in this work.

A comprehensive method does not exist today to address the evolution of Systems of Systems as seen during military SoS stakeholder planning. A comprehensive method requires the exploration of system acquisition, system development, technology refresh, and system reallocation trade space available to individual stakeholders. The long term view introduced by life-cycles and decision impacts, the multi-stakeholder environment, and the need to account for uncertainty in the long term planning process are not yet addressed in the presented methods. The following is a summary of the specific observations of capabilities lacking in current individual methods:

**Observation 1: Lack of Uncertainty Quantification** There is inherent uncertainty in the outcome of any decision. Capturing decision-related uncertainty is crucial to informed decision making. The defense planning problem defined in Chapter 1 has specific uncertainties that need to be represented: scenario and environment, development of technology and systems, and system performance. Each of these particular uncertainties is individually captured in the methods outlined in this chapter but not together. Addressing the defense planning problem requires capturing all the uncertainty sources and evaluating their impact. Cumulative uncertainty could increase the noise with regard to measuring future states and become an overwhelming concern.

**Observation 2: Lack of Addressing Multi-Mission, Multi-Stakeholder Aspects** Current SoS methods focus on a single set of systems with a single mission. Current methods need to be expanded to address multiple SoS and missions simultaneously. The ability to not just take the view of a single SoS stakeholder, but multiple stakeholders, is key to addressing multiple SoS and missions.

**Observation 3: Lack of Temporal SoS Influence Model** Current methods described in this chapter lack attention to the temporal aspects of a evolving system of systems. Many methods evaluate a specific grouping of systems or optimizing the construct of systems for a static point in time or for the next time step in an evolution. In reality, the development of a SoS is highly dependent on the time sequential decisions stakeholders make and the delayed feedback loop present due to the development process.

**Observation 4: Lack of Robust and Flexible Design Considerations:** Architecture evaluation and determination has largely been based on optimization techniques. Little regard has been given to the impact of uncertainty (robust design) or shifting requirements (flexible design) on the evaluation and formation of a 'new' SoS. Designing against uncertainty and shifting future requirements is a key part of the SoS defense planning process.

**Observation 5: Lack of Stakeholder Constraints** Very few methods fully take into account the constraints that are imposed on each of the Stakeholders. Looking at the individual budgets applied to system acquisition and potentially operation is one step. Developing an understanding of the impact past decisions decision is key to constraining those further in the future. This includes technology and system development delayed availability for use by stakeholders above and beyond budget constraints. Additionally, the need to not only supply a single mission capability but invest in multiple mission capabilities allows the results of the budgets constraints to fully be realized.

**Observation 6: Lack of Stakeholder Decision Space Exploration** Few of the methods identified address exploring the full stakeholder decision trade space. The decisions made by military stakeholders are not a simple yes or no. At a high level, stakeholders can decide to reallocate current resources, recapitalize or acquire new systems, or refresh current systems combined together within resource constraints.

**Observation 7: Lack of a Concrete, Multi-Cycle Evaluation** Many of the methodologies applied to stakeholder decision making outlined above focus on a single design cycle or a single stage in ongoing decision making. A key aspect of a SoS is its evolutionary characteristics. The central concept for evolution is a dynamic boundary with old obsolescent systems leaving and new systems joining. No single stakeholder controls the addition and retraction of systems. No methods address the evolutionary characteristics present in the military forces structure planning problem.

## 3.10 Research Objective

In summary, there has been a large body of work surrounding SoS Engineering for describing architectures, developing architecture alternatives, and evaluating architectures (including SoS environmental impacts). The body of work explored through this chapter and aligned with the conceptual model developed in Chapter 1 is summarized in Figure 3.10. Stakeholder decisions making is at the center of the feedback loop needed to address SoS evolution. The identified gaps show a lack of multi-stakeholder, multi-mission

Table 3.1: Existing Method Evaluation Summary

| **Current Method**<br>⊙excellent representation<br>●good representation<br>◐fair representation<br>○poor representation<br>⊗ no representation | **Multi-Stakeholder Decision Making** | **Evolutionary Feedback Loop** | **Technology and System Development** | **Capturing of Uncertainty** | **Architecture Representation and Evaluation** | **Environment and Scenario Representation** | **Multi-Mission Objectives** | **Defined SoS Process** | **Temporal Impacts** |
|---|---|---|---|---|---|---|---|---|---|
| SE Vee | ⊗ | ⊗ | ○ | ⊗ | ⊗ | ⊗ | ⊗ | ○ | ⊗ |
| SoS Trapeze and Wave Models | ⊗ | ◐ | ● | ⊗ | ⊗ | ○ | ⊗ | ⊙ | ◐ |
| TIES & TIF Methods | ⊗ | ⊗ | ◐ | ◐ | ○ | ◐ | ○ | ⊗ | ⊗ |
| Robust and Flexible Design | ⊗ | ⊗ | ⊗ | ◐ | ⊗ | ⊗ | ◐ | ⊗ | ⊗ |
| SoS Capabilities Technology Evaluation | ○ | ⊗ | ◐ | ○ | ○ | ● | ◐ | ○ | ⊗ |
| Architecture Representation Methods | ⊗ | ○ | ○ | ⊗ | ⊙ | ⊗ | ⊗ | ◐ | ○ |
| SoS Analytic Workbench | ○ | ◐ | ○ | ⊗ | ◐ | ◐ | ○ | ● | ○ |
| Traditional Scenario Based Planning | ⊗ | ⊗ | ⊗ | ⊗ | ⊗ | ◐ | ⊗ | ⊗ | ⊗ |
| MUSTDO Framework | ◐ | ○ | ⊗ | ⊗ | ● | ◐ | ◐ | ● | ⊗ |
| SoS Evolution Methods | ⊗ | ○ | ○ | ⊗ | ◐ | ○ | ⊗ | ◐ | ⊗ |

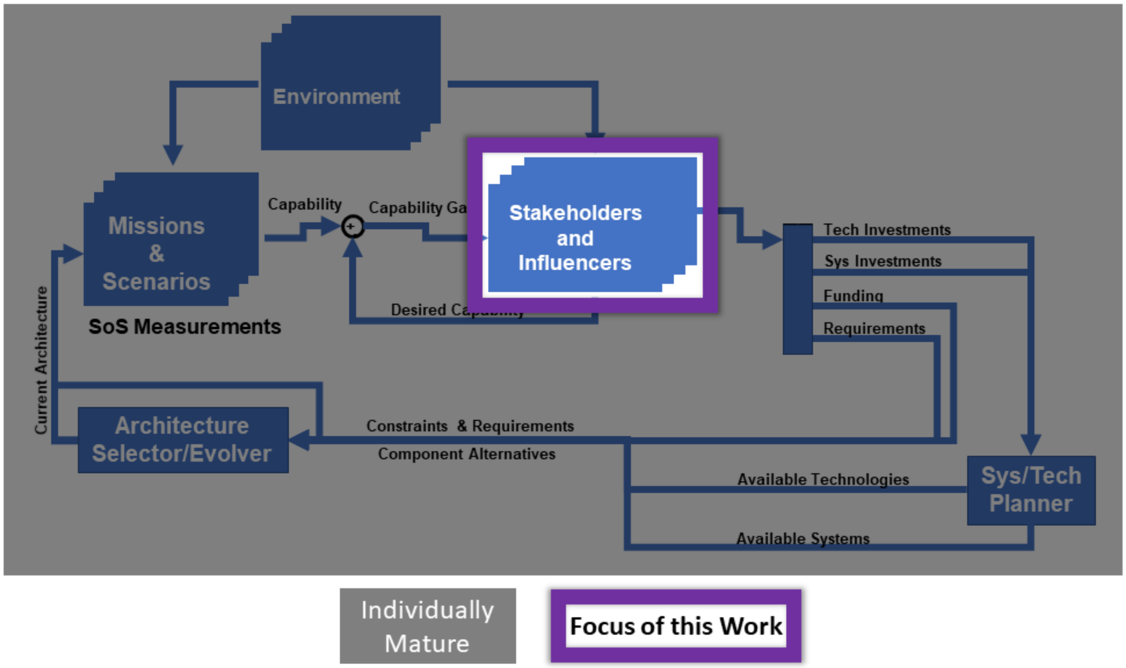


Figure 3.10: Concentration of this Work

decision maker evaluation.

It is crucial to address the problem of defense stakeholder planning given the observations of the current capability gaps. Therefore, the objective of this dissertation is:

***Research Objective:*** *To develop a new methodology that will instantiate the evolution of a System of Systems with regards to the decision making of the stakeholders accounting for the influence of the external environment, the morphing of the requirements, and the availability of resources over the lifetime of a SoS to enable individual stakeholder decision making under uncertainty.*

# CHAPTER 4
# LITERATURE REVIEW, RESEARCH QUESTIONS, AND HYPOTHESES

Chapter 3 outlined the specific capabilities needed to address the stakeholder defense planning problem as a System of Systems analysis problem. Current SoS analysis methods were outlined and evaluated in their ability to address the defined problem. Mature areas of research and definition were identified as well gaps within the existing body of knowledge.

Many aspects of SoS analysis have grown to a mature level including the representation of architectures, the evaluation of architectures, and applicable SoSE processes. But, there is a lack of multi-mission, multi-stakeholder decision-making including uncertainty associated with the SoS planning and stakeholder strategic defense planning processes. Including these aspects in a SoS planning process would enable a clear picture of the SoS evolution. The research questions define in this chapter are developed from the observed gaps previously identified and focused on addressing the Research Objective of this work.

Research Questions 1, 2, and 4 are individually motivated by Research Objective of this work. Research Question 1 (RQ1) addresses the development and capture of a decision trade-space . Researh Question 2 (RQ2) addresses the the evaluation of the decision trade-space. The investigation of RQ1 and RQ2 result in Research Question 3 (RQ3) which addresses developing a tractable and computationally solvable representation. Research Question 4 (RQ4), developed from the RQ3 investigation and the Research Objective, addresses the creation of usable decision information from decision space evaluation. Each Research Question is investigated through a search of literature, an identification of alternative approachs or solutions, and a comparison of alternative solutions existing in the present state of the art. Hypotheses are developed based on the evaluation of alternative solutions, the solutions capability to help address the defined problem, and the identified gaps in state of the art.

## 4.1 Representing and Populating the Decision Trade Space

The first step to in enabling the exploration of stakeholder decisions is to appropriately represent and evaluate the decision space. The representation encompasses how to represent the decisions a given set of stakeholders may make, the resulting SoSs and developmental states, and the utility each stakeholder gains in return. It should account for the size of the decision space and the uncertainty surrounding decision outcomes.

> ***Research Question 1:*** *How can the time-dependent decisions of multiple stakeholders be captured and combined to develop a full representation of potential outcomes for evaluation given resource constraints and uncertainty?*

RQ1 can be decomposed into two components, capturing the decision space and evaluating the decision space. The first comonent addresses how to mathamatically capture a stakeholder decision space in order to evaluate stakeholder decisions (Research Question 1.1). The second questions how to evaluat the decision space and provide usable metrics to help inform stakeholders (Research Question 1.2).

### 4.1.1 Representing the Decision Alternative Space

A generic conceptual model of defense stakeholder planning was outlined in Chapter 2 and captured in Figure 2.5. Populating the multi-stakeholder decision space is needed to create a decision making playbook in order to help inform a single stakeholder. To address Research Question 1, a trade space of all stakeholder decisions and their outcomes must be developed:

> ***Research Question 1.1:*** *How can the decision alternatives of multiple-stakeholders be captured and combined to develop a full accounting of potential outcomes for evaluation?*

The decision space (i.e available stakeholder actions) is dynamic and changes over time. The decision space is a result of the current state reached by the stakeholders and SoSs as

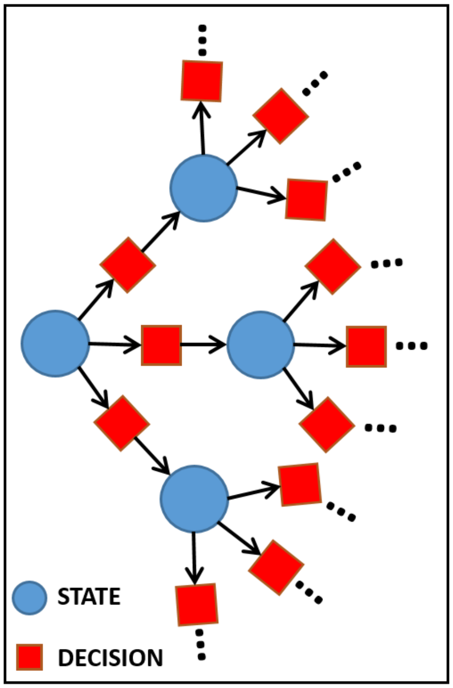


Figure 4.1: Simple Single Stakeholder Decision Tree

a whole. Three alternatives are explored below in their applicability to representing the decision and outcome space needed to address RQ1.1.

*Decision Tree*

A decision tree is commonly used to represent a player's sequential choices and outcomes with the goal of identifying an optimal policy or path through the tree, Figure 4.1. At each node a player can select specific actions that result in a transition to a new node in the graph [117]. In this paradigm, each stakeholder can be considered a player, each node, or state, represents a SoS implementation, and each action represents a stakeholder's decisions.

Throughout this work, the following definition will be used with respect to states and actions regarding decision trees:

- $\mathbf{S}$ is a set of states $\{s_1, s_2, \cdots, s_n\}$
- $\mathbf{a}$ is a set of actions $\{a_1, a_2, \cdots, a_m\}$

A “stakeholder” will be synonymous to a “player” within a decision or game framework. The specific decisions a stakeholder can make are referred to as “actions”. A given SoS composition and progress of investments is used to represent a given “state”.

Using the decision tree construct enables a number of different methods for exploitation including dynamic programming, tree search and path finding methods. Additionally, the techniques can be expanded to include stochastic attributes given a selected action at any given state along with utility functions that describe risk behavior (aversion or seeking). [117]

There are two options to handle the potential size of the decision tree, or decision trade space, when growing and representing during exploitation. The decision tree can either be pruned during growth (pre-pruned) or it can be reduced after growth (post-pruned). Pre-pruning is used while developing the decision tree and post-pruning is used while solving it.

**Pre-Pruning The Decision Tree** Pre-pruning smartly grows the decision tree during initial development. At each decision cycle for stakeholders, there will be a set of decisions available to them. Pre-pruning can limit these decisions by available resources, actual and anticipated rewards, and agent based rules.

One option is to constrain the available decisions. An example of constraints would be limits on budgets for technology maturation and system development. Additionally, previous year or time step resource commitments reduce currently available resources.

Another option is to pre-prune by not growing the entirety of the tree and limiting the growth via a set of rules or behaviors. These rules and behaviors can be adjusted to develop an understanding of what would be a full set of decision trees. An example of agent behavior rules could be prioritizing technology investment over system development, prioritizing a single mission above others, or optimizing for single decision-cycle impacts. Pre-pruning acts to limit the growth of the tree but would require multiple trees to be grown to approximate a full decision tree. Pre-pruning results in either a tree representing a subset of all possibilities or is a composite of many different trees.

**Post-Pruning The Decision Tree** Post-pruning can follow many existing algorithms including Branch and Bound, A*, D*, etc., all of which are varying algorithms that span the continuum between Depth First Search (DFS) and Breadth First Search (BFS). Theses algorithms work to identify a single optimal, or near optimal, path through a directed weighted graph. The ultimate reward experienced on a given traverse is based on transition returns.

The algorithms could be used to grow the tree but often are used once a decision tree exists. Additionally, backwards induction techniques fully solve a deterministic decision tree working backwards from the end leaves. This includes Dynamic Programming that answers the question: "What is the best action now, assuming optimal behavior at all potential future decision points?" [117]. The method is described in Equation 4.1. Any method of post-pruning requires the development of the entire graph before they can be fully applied and run. At best, the computation order is $O(|E||V|)$ where $E$ is the number of edges and $V$ is the number of vertices or nodes.

$$V^*_{t(x)} = \underset{a \in \mathbf{A}(x)}{max} [r_t(x,a) + V^*_{t+1}(x_{t+1}(a))] \tag{4.1}$$

where $V_t(x)$ is the *Value* of state $x$ with $V*$ representing the optimal value, and $r_t(x,a)$ is the return from taking action $a$ in the current state $x_t$.

### *Game Theory*

Traditional game theory is a second option to use to populate the decision and decision outcome space. Traditionally, game theory has worked to solve for an equilibrium state in 1v1 deterministic and static games where a payoff is known (e.g. Prisoners Dilemma, Hawk-Dove) [118]. These basic game theory approaches lack the needed temporal component.

The addition of a temporal component, or continued decisions, results in combinatorial games characterized by win-loss, deterministic, and known outcomes of decisions [119]. Many standard two player games (simple to complex) fall into this category (e.g. connect four, checkers, chess, baduk). Many solution techniques are similar to those above for decisions trees [118].

The above traditional game theory techniques do not address cooperative multi-player aspects. It is possible to extend the traditional 1v1 game to three non-cooperative players [120]. A number of approaches exist to address the cooperative player aspects including shapley values representing relative influence [120, 121, 122, 123, 124, 125] and coalition based decision making (to join or not to join blocks) [126, 127, 121, 122, 128, 120, 124]. Additional constructs and mechanics can be used. An example used specifically for SoS evolution coordination between stakeholders is a transfer contract method developed by Fang [70, 129]. The constructs used don't lend themselves to traditional deterministic solving methods due to the size and complexity of the state-action space [129].

In order to apply traditional game theory to develop the decision and outcome space, an asymmetric, cooperative, sequential, imperfect knowledge, discrete game must be developed and then solved. Stochastic aspects to address uncertainty are developed in a Section 4.1.2. It quickly becomes difficult to identify a deterministic solution with the growing complexity of the game type even before the addition of uncertainty. Typical player decision methods, or decision rules, used to solve games include equilibrium and minimax/maximin approaches. At each stage a decision is made using a heuristic. Techniques beyond simple heuristics are needed as rules are applied over time (sequentially), across multiple players (cooperative), and with asymmetric outcomes (varying rewards and goals). Additional ways to handle the complexity are discussed in Section 4.3.1.

### *Modeling and Simulation*

Modeling and simulation becomes relevant when deterministic analytical solutions are not available and/or a temporal based understanding of the problem is needed. Primary categories of simulations include: Discrete Event Simulation (DES), System Dynamics, and Agent-Based Modeling. System dynamics is traditionally associated with solving systems of equations and looking at non-transient and transient behavior of inputs over time. There is current work to extend the SD paradigm to apply to and solve SoS problems [130]. DES are built around cause-effect relationships without constant temporal representation. Agent-Based Modeling traditionally relies on time-step based simulation (though DES can be used) but is defined by the model of an agent continuously sensing it's environment, evaluating actions, and acting on it's environment.

Agent-Based Modeling is the most applicable simulation type to the problem. It allows a construct or framework to be developed along with agent decision methods as inputs. These inputs can be statically held during the dynamic simulation execution. Rules can be given to each agent similar to heuristics used at each step in a sequential game. The execution allows a brute force exploration of potential outcomes and, in this case, a variation of behaviors to explore within a scenario. The brute force runs do not result in exact state/action pairs as a function of time and therefore pose challenges to consolidate into a play book.

### *Truth Model Development*

The explored decisions space representation alternatives offer benefits but each has associated downsides (Table 4.1). The Decision Tree paradigm offers a construct utilized previously in SoS evolution work [129]. Each action vector, composed of individual actions by each stakeholder, can be used to play out an outcome and develop a subsequent state, defined by the current definition of all SoS under consideration. The Game Theory approach offers a fundamental view of decision making at each individual state based

Table 4.1: Decision Trade-space Representation Alternatives

| **Alternative** | **Pro** | **Con** |
|---|---|---|
| **Decision Tree** | Appropriate construct unfolding SoS decision-cycle; | Does not scale well |
| **Dynamic (Sequential) Game Theory** | Appropriate construct for individual decisions | No temporal memory or player learning over time<br>Issues scaling closed form solutions as complexity grows |
| **Agent Based Modeling** | Simple problem setup | Partial sampling of decision space<br>Decision rules determination<br>Difficult to aggregate state-action pairs |

on the anticipated return (mission utility) for individual stakeholders. Game Theory has been used in previous SoS evolution problems [34, 131]. A more exploratory method than Game Theory is needed due to the complexity of the decision space and representation of the defined defense planning "game". Agent-Based Modeling provides that framework to develop agents with heuristic decision making and play out scenarios via simulation. ABM does not allow for a full accounting of the decision space but can help accommodate the complexity. Forming discrete states can be difficult when approaching continuity via time step simulation. Discrete Event Simulation (or potentially course time steps) can result in discrete steps. A Discrete Event Simulation with hard time steps combined with agent representations can result in a simulation that can both be sampled (instead of fully determined) and will result in discrete state-action points. A final comparison of the alternatives is summarized in Table 4.2.

A fundamental assumption and input of this problem is the development of the scenario and agent models. The resulting analysis of the decision space is rooted on the assumptions and understanding of the development. The validity of this model is assumed during the body of this work and will be subsequently referred to at the Truth Model from which all decision evaluation is done.

Table 4.2: Decision Trade-Space Representation Comparison

| **Trade-Space Representation Alternatives** | **Temporal State Nature** | **Ease of Setup** | **Full Characterization** | **Scalability** | **Uncertainty** | **Action Space Representation** |
|---|---|---|---|---|---|---|
| Decision Tree | ● | ◐ | ● | ○ | ⊗ | ● |
| Dynamic (Sequential) Game Theory | ○ | ○ | ◐ | ○ | ⊗ | ● |
| Agent Based Modeling | ◐ | ● | ○ | ● | ⊗ | ○ |

⊙excellent, ●good, ◐fair, ○poor, ⊗ none

### 4.1.2 Capturing Uncertainty

Uncertainty can dominate future outcomes of actions and comes in many forms. Future outcomes of actions made in the present, especially at a significant distance forward in time, can be dominated by compounded uncertainty and have results that often are indeterminable. Stakeholder strategic actions with respect to defense stakeholder planning include deciding to invest in maturing a specific technology, initiating an acquisition, or deciding to re-appropriate existing resources to a new mission. The cumulative result of each of these decisions made in the present on the resulting force capabilities is subject to uncertainty as defined in Chapter 3.

Will the new technology result in the system capability needed? Will the system acquisition schedule and capability results be as expected? Will the system acquisition result in the capabilities needed in the time frame expected? Will the developed technology and acquired system even perform as expected when combined with the rest of the force structure? What will the force structures ability to execute a mission at a given time even be?

The results for RQ1.1 leave the following still open:

> ***Research Question 1.2:*** *How can the uncertainty existing in the multi-player, multi-objective decision space be represented, captured, and accounted for?*

In this section, representation of uncertainty is explored and characterized. Subsequently, three options to handle the uncertainty within the framework described in Section 4.1.1 are explored in light of the characterization.

### *Representing Uncertainty*

There are two fundamental classifications of uncertainty, epistemic and aleatory:

**Epistemic Uncertainty:** This is commonly referred to as systematic uncertainty. This references the degree of uncertainty strictly formed as a function of the chosen model or representation. This could be due to the selection of a lower fidelity model or due to a model developed around incomplete knowledge of the underlying phenomenon. With more information and a better model (higher fidelity) the uncertainty can be reduced. Conceptually, a perfect model will have zero epistemic uncertainty.

**Aleatory Uncertainty:** This is commonly referred to as inherent uncertainty in a non-deterministic event. With increased knowledge or a better model, there is no reducing aleatory uncertainty. It is the inherent randomness in the world.

The purpose of classification is to enable representations. In the framing of the decision representation in Section 4.1.1, a definition of a Truth Model was determined as an input to this work. This Truth Model is to be used to measure the methods explored and selected for the methodology (Chapter 5).

The classification of epistemic and aleatory, or representation and fundamental, uncertainty enables the ability to appropriately address each one. In the outlined problem of stakeholder defense structure planning, the uncertainty in the future is categorized as fundamental with only time (e.g. continued progress of technical maturation and fundamental

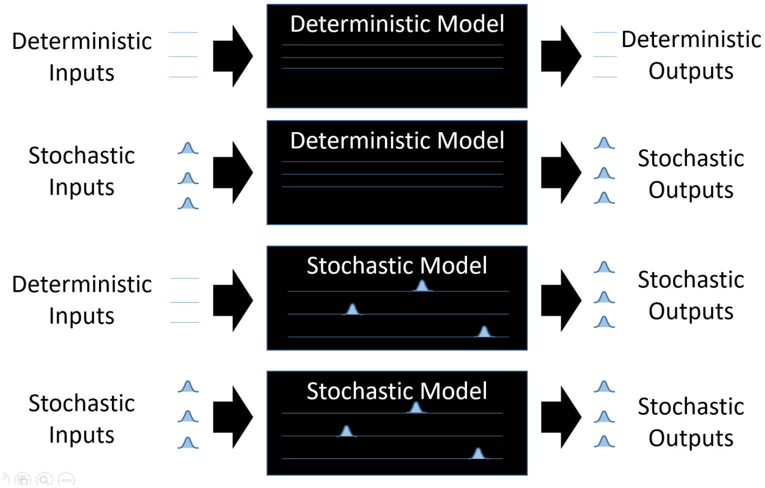


Figure 4.2: Representing Uncertainty in Models

understanding of technology) culling the uncertainty. The fundamental uncertainty is the uncertainty needed to be represented within the Truth Model. The Truth Model will also represent epistemic uncertainty.

A final classification of uncertainty comes with a representation built on the Truth Model (e.g. meta-model used for evaluation purposes). Any derived representation or analysis regarding the Truth Model fundamentally is subject to epistemic uncertainty. This uncertainty is added stochastic error introduced when abstracting the Truth Model. This aspect is further explored in subsequent sections that address meta-models for the described Truth Model (Section 4.1.1).

*Capturing Uncertainty in a Model or Simulation*

The goal with capturing either case of uncertainty is to quantify and represent it in a model. When modeling, there are inherently four options to capture the impact of uncertainty as depicted in Figure 4.2.

Fundamentally, modeling uncertainty means dealing with repeat runs with a stochastic model or Monte Carlo runs with a deterministic model. If there is uncertainty in both the inputs and the model itself, there is no fundamental way to distinguish between the two im-

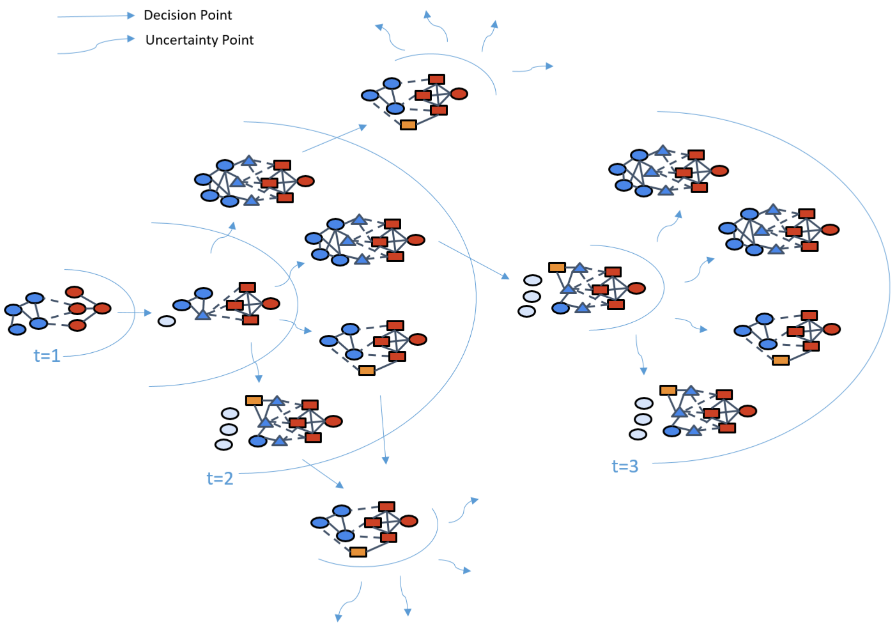


Figure 4.3: Example Multi-Player Action with Uncertain Outcomes

pacts. Potentially, less accurate methods can be used, for example, when using the Monte Carlo method, repeat runs can be recorded with means and variances as appropriate. Ultimately, any randomness used to incorporate uncertainty manifests itself as a distribution of the output metrics. It is common to represent the stochastic outputs as a mean and variance of a normal distribution that assumes the Central Limit Theorem applies.

### *Markov Decision Process*

Game trees and complex games can be extended to include stochastic outcomes using Markov Decision Processes. MDPs make two inherent assumptions: (1) time-separable and (2) additive. The first assumption means that cost and rewards are independent of time and only a function of state. The second assumption emerges from the additive property of rewards received from each state. [132] Both are inherently derived from the Markov property of MDPs but should be noted in comparison to this stakeholder defense problem which has temporal components.

A Markov Decision Process (MDP) adds another layer of complexity to the idea of a decision tree with variable outcomes. It adds the idea of a state variable space that defines each state. Additionally, the actions able to be executed are time dependent. This work looks at finite-horizon models. Fundamentally, MDPs are build on Equation 4.2. The next state is ultimately represented as a stochastic function of the current state and the current action. Application to the SoS problem is visually captured in Figure 4.3.

$$p(x_t + 1|h_t, a_t) = p(x_t + 1|x_t, a_t). \tag{4.2}$$

where $\mathbf{X}$ is a set of state variables $\{x_1, x_2, \cdots, s_k\}$, $\mathbf{A}$ is a set of actions $\{a_1, a_2, \cdots, a_j\}$, and $(x_t, a_t)$ represent state-action pairs available at $t = \{1, 2, 3, \cdots\}$, and $\mathbf{h_t}$ is a set of the history up to time $t$, $(x_0, a_0, x_1, a_1, ...x_{t-1}, a_{t-1}, x_t)$.

Expected values for the state-value and action-value functions are depicted in Equation 4.3 and Equation 4.4 respectively.

$$V_\pi(s) = \mathbf{E}_\pi[G_t|s_t = s] = \mathbf{E}\Big[\sum_{s \in S} \gamma^k R_{t+k+1}|S_t = s\Big] \tag{4.3}$$

$$Q_\pi(s, a) = \mathbf{E}_\pi[G_t|s_t = s, a_t = a] = \mathbf{E}\Big[\sum_{s \in S} \gamma^k R_{t+k+1}|s_t = s, a_t = a\Big] \tag{4.4}$$

There exist a number of well defined extensions to the vanilla MDP. Partially Observable MDPs (POMDP) combines the idea of a deterministic Hidden Markov Model with an MDP. An agent making decisions within the environment does not have perfect knowledge of states and represents the truth-states with belief-states. The agent has a “belief” of the current state and makes decisions based on that belief. Extending this further is the concept of Decentralized POMDP (DEC-POMDP). The DEC-POMDP construct incorporate the stochastic and partial observations with distributed agent decision making. Solving of DEC-POMDP in an efficient and scalable manor is a focus of current research [133, 134, 135, 136, 137, 138].

**Stochastic Dynamic Programming** Traditional methods used to solve a Markov Decision Process or Stochastic Decision Tree can be used to solve their determinsitic counterparts in a similar manner. Stochastic Dynamic Programming answers the same question as Dynamic programming but utilizes the expected return for each action given the transition probabilities to a new state (Equation 4.5).

$$V^*_{t(x)} = \underset{a \in \mathbf{A}(x)}{max} [r_t(x, a) + \sum_{x' \in \mathbf{X}} p(x'|x, a) V^*_{t+1}(x')] \tag{4.5}$$

where $V_t(x)$ is the *Value* of state $x$ with $V*$ representing the optimal value, and $r_t(x, a)$ is the return from taking action $a$ in the current state $x_t$.

### *Stochastic Games*

The formulation of repeated and stochastic games quickly converges on an MDP. Similar solution methods can be used to address repeated stochastic games as were previously mentioned for an MDP [118, 139, 140, 141]. This representation may also include the application of standard decision making techniques from game theory in combination with expected returns [139]. With non stochastic games, the evaluation begins to quickly grow uncontrollably as the state and/or action space grows and other approaches are needed [142]. The problem space that MDPs and repeated-stochastic games converge on can be call Decision-Theoretic Planning [132].

### *Representing Uncertainty and Selected Representation*

When looking at the variability due to a given policy, Prashanth attributes it to two types of uncertainties: “(i) uncertainties in the model parameters, which is the topic of robust MDPs and (ii) the inherent uncertainty related to the stochastic nature of the system, which is the topic of risk-sensitive MDPs”[143]. The first of the two is out of scope of this work and is subjected to the development of an acceptable Truth Model 4.1.1. The second source is key to this body of work and will define how uncertainty is handled (see Section that addresses risk-sensitive policy generation). Uncertainty is introduced through the definition of the Truth Model which includes definition of all stochastic variables. Specifically, the Truth Model can be represented using a high dimensional MDP. Further sections address how uncertainty is handled when developing the decision trade space and subsequently evaluating it.

## 4.2 Evaluating Stakeholder Decisions

It is necessary to establish metrics to evaluate the decision trade space previously defined. Measuring the impacts of decisions quantitativily is necessary for evaluating decisions. This yields the following research question:

> ***Research Question 2:*** *How can the decision space be evaluated and sequential decision alternatives be compared?*

Three metrics for evaluating the decision space are presented: (1) risk, (2) volatility reduction, (3) opportunity cost. Risk based decision making allows for the exploration of stakeholder risk postures and helps tie it to concrete policies/strategies and their potential outcomes. State-action importance is key to understanding significant decision points. Volatility decreases across a state (given the available sets of actions) shows the culling of uncertainty and importance of that decisions point. Measuring the impact that a decision has on the precluding of future outcomes allows a relative measure of action impact. Measuring the relative opportunity cost of any given action set will provide important insight to decision significance.

### 4.2.1 Evaluating Risk and Reward

It is imperative to enable robust planning for defense stakeholders. Representing uncertainty is the first step to enabling robust planning (Section 4.1.2). The second is risk-based evaluation techniques and policy development. Evaluating stakeholder risk is crucial to developing risk based policies that are necessary for robust playbook creation. Thus, the following research question is posed:

> ***Research Question 2.1:*** *How can the risk and reward of a stakeholder's individual decisions be assessed and compared?*

Understanding the risk associated with a specific decision is just as important as understanding the relative significance or classification of decisions. In financial engineering, a traditional method of evaluating portfolios of assets is to utilize mean-variance theory. This theory takes past data to estimate values of volatility (or risk) and mean-return (or reward) for making the decisions to purchase and hold particular assets. The concept applies to individual asset investments as well as portfolios of assets. The constructed volatility and return is indexed against a zero-risk (or no volatility) assets, typically an assumed bank holding rate of return. This method can be used to evaluate various investment decisions against each other based on a decision maker's risk tolerance vs. desired returns. [144]

A similar method can be used to develop risk measurements for decisions being presented to a stakeholder at any state. Each decision can be viewed as a portfolio of assets held by the stakeholder with each individual action resulting in a portfolio selection. At any given point in time, a volatility measurement can be taken with regards to the future states within the MDP (similar to the Black-Scholes method where a pricing lattice is used for asset valuation [144]). A risk measurement can be taken based on the volatility of future states in addition to the expected reward.

Other methods have been developed over time to measure risk based on the mean and volatility of a return. Examples of statistical properties include value-at-risk (VaR), conditional value-at-risk (CVaR), or exponential-utility [145]. Value at risk looks for the single metric of value based on an acceptable probability margin [146, 144]. CVaR is defined as the expected loss (or return) for a given worst case probability [144]. These options are similar but vary in risk-acceptance definition. VaR and CVaR look at a single metric rather than multi-criterion such as mean-variance portfolio theory. There is no effort to build a set of decisions when using VaR or CVar as exists when using portfolio theory. Some work has been done to apply the VaR and CVaR approaches to SoS [101] but it is not yet mature.

Thus far, traditional risk consideration techniques have been derived from investment science techniques. The field of risk-sensitive MDPs is an alternative that combines MDPs that traditionally look at the expected return and incorporates a sensitivity to the volatility of the return as well [145]. Methods involving MDPs have traditionally excluded this topic as the expected return provides significant computational issues alone, without the addition of variance inputs [147]. Typical risk-sensitive metrics work to calculate a variance of return in addition to an expected return for a given state-value or action-value [147, 148]. The sampling can be computationally expensive and done brute force with a MC sampling method [147]. Other methods build on risk measurements to encompass full RL techniques like the multi-level time scale optimization method using Lagrangian policy optimization with a lower level actor-critic setup with returns constrained by variance [143]. There are two commonalities of traditional risk-sensitive approaches: (1) They require specific sampling methods to ensure a variance measurement can be made and (2) they don't account for the volatility directly but only indirectly through a cost function placed on the reward.

A summary of each method can be found in Table 4.3 with a clear gap emerging. A portfolio approach for analyzing an action-portfolio at each state is necessary to develop policies for each state. Relative comparisons and indirect representations provided by risk-sensitivty methods are not sufficient. A direct comparison of the return and volatility necessary to fully attribute the results to a commonly defined (absolute) risk tolerance.

Using the explored techniques, a hybrid approach can be formed. The mean-variance approach allows for these needs on a per state basis to be addressed. Combined with the MC methods of the more brute force risk-sensitive MDP approaches allows for the appropriate variance metric to be calculated at the expense of computational need. The increased computation time can be accepted as long as it is applied to the reduced meta-model, not

Table 4.3: Risk-Reward Policy Evaluation Options

| Alternative | Pro | Con |
|---|---|---|
| Mean-Variance Portfolio Theory | Accounts fully for variance<br>Allows direct scaling of risk-sensitivity<br>Supports portfolio selection | Not yet applied to MDP problem |
| VaR | Comparative metric between similar risk profiles (return open) | Single metric based<br>Indirect scaling of risk-sensitivity |
| CVaR | Comparative metric for similar return profiles (risk open) | Single metric based<br>Indirect scaling of risk-sensitivity |
| Risk-Sensitive MDP | Incorporates risk into existing MDP problem definition | Requires significant sampling for variance measure<br>Variance is not fully accounted for<br>Variance is not directly scalable, 'k-factor' like metric |
| Mean-Variance Risk-Sensitive Policy Generation Hybrid | Allows for direct scaling of risk-sensitivity<br>Enables action-portfolios to generate policies<br>Directly accounts for return vs. volatility | Requires brute force sampling (computationally costly) |

the full Truth Model MDP.

The hybrid approach if fully detailed in Section 5.4.1 as part of the methodology. The approach first samples the mean and variance of an individual stakeholder return for all available actions for each state. A action Pareto frontier is established and a risk-tolerance level ($\xi$) is used to select a position on the frontier. The risk-tolerance based point yields a relative weighting for efficient actions. This risk-tolerance level based weighting is used to create a risk-tolerance based policy. A direct comparison of methods can be seen in Table 4.4.

> ***Hypothesis 1*** *If the risk-tolerance level of a stakeholder is varied as an input to the Return mean-variance risk-based policy algorithm, then the Pareto efficient actions will be identifiable.*

Table 4.4: Risk-Based Policy Approach Alternatives

| **Risk-Based Policy Methods**<br>**Alternatives** | **Stochastic Based Risk** | **Full Variance Representation** | **Absolute Variance** | **Risk Sensitivity Representation** | **Existing Solution Method** |
|---|---|---|---|---|---|
| Mean Variance Method | ● | ◐ | ◐ | ● | ○ |
| Value at Risk | ● | ○ | ○ | ○ | ◐ |
| Conditional VaR | ● | ○ | ○ | ○ | ◐ |
| Risk-Sensitive MDP | ◐ | ○ | ○ | ◐ | ● |
| Hybrid Mean-Variance/Risk Sensitive Method | ● | ● | ● | ● | ⊗ |

⊙excellent, ●good, ◐fair, ○poor, ⊗ none

### 4.2.2 Evaluating Decision Importance

In addition to developing specific policies that an individual stakeholder can follow, it is important to identify when an important decision point is reached. An important decision point can be identified by the reduction in volatility of a future return. Each state acts as a decision point and actions result in a maintenance or a decrease in outcome volatility. Thus there is the following research question:

> ***Research Question 2.2:*** *How can the volatility before and after a given state be measured in order to identify significant decision points?*

A given state's importance can be determined by evaluating the measure of uncertainty reduction. Any decision by a stakeholder (or action by a player in the Truth Model) will yield a given distribution of future stakeholder mission utility (or reward). More specifically, the significance of a state can be determined by the relative remaining uncertainty after each action is taken. The state can be identified as important if there is a significant difference between uncertainty measurements for each action. There are two identified potential methods of measuring relative uncertainty depicted in Table 4.5.

Measuring the entropy of a data set has been used across many fields including information theory. Specifically, "graph entropy" has been used as a measure to characterize structure, complexity, and noise of data sets [149]. The generic equation that defines entropy is captured in Equation 4.6.

$$E(G) = \sum_{i \in G} -P_i log_2 P_i \tag{4.6}$$

where $P_i$ is the probability of occurrence for value of state $i$ in graph $G$.

When a single state is analyzed, the relative entropy between actions can be analyzed to demonstrate the importance of the decision. If entropy is high, then there is a significant impact due to a decision or outcome along that specific policy. Typically, a measure is made before and after a split is developed in a decision tree. As the tree grows and is used to classify data, the entropy of the data characterized at each node is reduced. The entropy of a general data set can be calculated using the probability of occurrence of individual items (or characters) and the traditional formula, seen in Equation 4.6 [150]. Similar approaches to general architectures of systems have been used to measure system complexity in truly complex systems: manufacturing, power grid, and railway transport systems. [151]

The alternative is to use traditional variance calculations of similar MC samples necessary for the entropy measurements. The variance provides an absolute measure of uncertainty whereby a single measure relative uncertainty is necessary. Traditional variances cannot be equally applied across separate metric types or measurements and are impacted by the scale of the metric. A relative comparison and rating of the options is summarized in Table 4.6.

> ***Conjecture 1:*** *The relative importance of each state can be evaluated using an entropy calculation of the overall stakeholder utility of finite time horizon MC*

Table 4.5: State Significance Options Options

| **Alternative** | **Pro** | **Con** |
|---|---|---|
| Variance of Return | Quantifies uncertainty measurement | Absolute measure of uncertainty<br>Requires MC samples per state |
| Entropy | Relative measure of uncertainty<br>Applicable to non-normalized returns | Simple calculation given samples<br>Requires MC samples per state |

Table 4.6: State Significant Determination Methods

| **Risk-Based Policy Methods** | **Relative Volatility Measurement** | **Computationally Simple** | **Reasonable Computation Time** |
|---|---|---|---|
| Variance of Return | ○ | ◐ | ○ |
| Entropy Method | ● | ◐ | ○ |

⊙excellent, ●good, ◐fair, ○poor, ⊗ none

*samples before and after each action using the meta-model.*

### 4.2.3 Evaluating Opportunity Cost

Identifying the importance of a specific decision follows identification of significant decision points. Action significance is a factor of opportunity cost where decision-point-significance concentrates on volatility. Stakeholders must understand when they are making a decision that precludes returns in one area versus returns in another. They must understand when they are trading future mission utility between two separate missions. This need yields the following research question:

***Research Question 2.3*** *How can the opportunity cost between individual stakeholder metrics for a given decision be assessed and compared?*

When considering and planning, it is imperative to identify opportunity costs. Traditionally, opportunity cost for a given decision compares a deterministic return from an action selection to the best of all other potential actions. This results in a static, deterministic measurement for a single return-metric.

A stochastic, multi-metric approach to opportunity cost is necessary to apply to the stochastic, multi-stakeholder, multi-mission problem previously outlined. The stochastic approach will compare the mean and variance for each individual stakeholder metric against that of all other actions (similar to comparing an actions outcome to the best of all others). Opportunity cost between metrics can be measured in relative shift of mean and variance of individual actions as outlined in Figure 4.4. The difference between the two alternatives is outlined in Table 4.7 with a relative scoring outlined in Table 4.8.

***Conjecture 2:*** *Individual stakeholder metric opportunity cost can be identified by comparing the mean and variance of stakeholder utility metrics.*

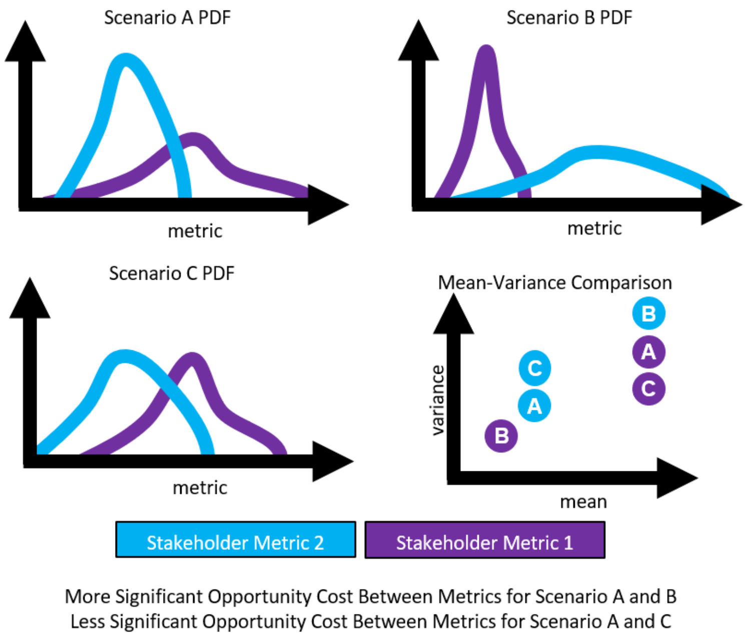


Figure 4.4: Stochastic Opportunity Cost Comparison

Table 4.7: Opportunity Cost Evaluation Alternatives

| **Alternative** | **Pro** | **Con** |
|---|---|---|
| Traditional Opportunity Cost | Simple and standard measurement | Single metric opportunity cost evaluation<br>Will require additional sampling<br>Will require additional sampling (no closed for approach to problem) |
| Stochastic Opportunity Cost | Compare two metric opportunity cost | More complex calculation and comparison<br>Will require additional sampling (no closed for approach to problem) |

Table 4.8: Opportunity Cost Calculation Methods

| **Risk-Based Policy Methods** | **Handles Multiple Metrics** | **Standard Measurement** | **Simple Calculation** |
|---|---|---|---|
| Traditional Opportunity Cost | ⊗ | ● | ● |
| Stochastic Opportunity Cost | ⊙ | ⊗ | ◐ |

⊙excellent, ●good, ◐fair, ○poor, ⊗ none

## 4.3 Reducing Dimensionality of Decision Trade-Space

Consider a simple single stake-holder responsible for a single mission, equivalent to a single SoS. It is easy to visualize a simple decision tree based on the single stakeholder's decisions to add or subtract systems over time as shown in Figure 4.1. Each state represents a stable point in the SoS with each action adding or subtracting from it.

In reality, the single stakeholder has many actions consisting of all combinations of technology refresh, asset recapitalization, and asset reallocation that fit within resource constraints. This is similar to the problem of attempting to fit as many high value objects of varying 3-D dimensions into a backpack. If the problem is expanded to include multiple stakeholders who make decisions which impact the performance of multiple missions (or multiple SoS) then the tree will grow beyond a usable form. If there is even a low number of potential actions for each stakeholder at a given node the tree size becomes unmanageable.

Powell identifies three places that Decision Trees and Markov Decision Processes are subject to the Curse of Dimensionality: the state space ($S_t = \{S_{t1}, S_{t2}, \cdots, S_{tn}\}$), the outcome space (probability of an outcome), and the action space ($\mathbf{a}_t = \{a_{t1}, a_{t2}, \cdots, a_{tn}\}$). [152] These three aspects of developing a tree are key drivers in unreasonable growth and are present in the force level planning problem. Each needs to be mitigated in order to maintain the feasibility of the problem. All three of these are much higher than traditional problems represented and solved via the MDP structure.

Two categories of options were previously presented to manage the size of the generated graph: pre-pruning during growth or post-pruning after growth (Section 4.1.1. Neither of these options present a usable solution to dealing with the significant size of the action and state space of the presented problem. This section investigates alternative methods that can be employed to deal with such large action-state spaces generated via methods explored in Section 4.1.1 and 4.1.2 to address Research Question 1.3:

> ***Research Question 3:*** *How can the inevitable resulting dimensionality of the multi-player decision space be reduced to a digestible and actionable trade-space?*

### 4.3.1 Reinforcement Learning Concepts

Reinforcement Learning (RL) has grown in popularity with the success of Deep Blue [153], AlphaZero [154, 155], and even multi-player video game AI players [156, 157, 156, 158]. Small state and action space games such as tic-tac-toe and connect four have been fully solved [159]. More complex games such as chess and go consisting of larger state spaces and deterministic outcomes require RL methods [159].

All RL methods are rooted in the simple agent-environment model show in Figure 4.5. The foundation of RL methods is the development of an agent (or player) who acts on its environment and then receives a return before moving to a new state. More complex formulations include multiple agents, partial observability of states, and multiple rewards. Many RL methods assume an underlying MDP exists in a $SARSA$ formulation: $(s_n, a_n, r_{n+1}, s_{n+1}, a_{n+1})$ where $s$ is the current state, $a$ is an action, and $n$ is the current discrete step. [158]

Dynamic stochastic games can easily be represented with an MDP construct where

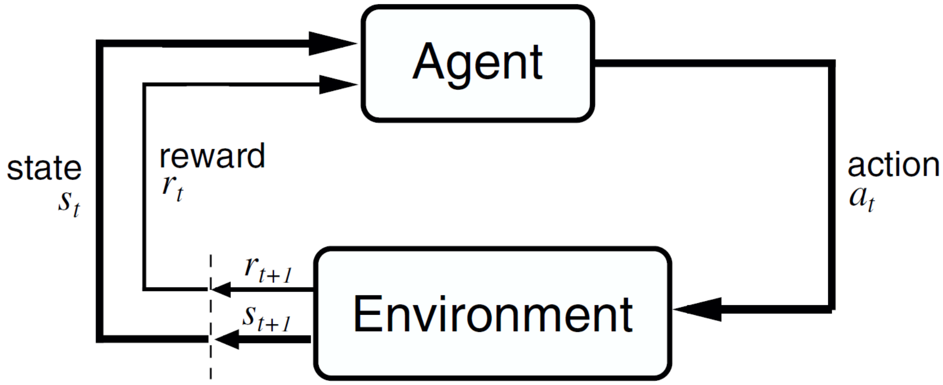


Figure 4.5: Reinforcement Learning Agent-Environment Interaction [158]

general state-value, action-value, and policies can be trained [140, 142, 160]. The paradigm can be extended to learning for large number of players [142], distributed learning for cooperative players (e.g. DSL MARL algorithm) [140], and combining cooperative and adversarial players (e.g. USCG algorithm) [160]. Many other solution paradigms exist as do representative problems to solve [161] with many solutions being problem-centric.

The general focus of reinforcement learning applied to sequential stochastic games is to solve for an optimal policy based on rewards for a set number of stakeholders in a live learning environment (simulation or real world). The resulting policies are then exploited in the real world. The problem formulations (e.g. grid world, robot and coffee) and common algorithms for solving lack the scalability to deal with the defense stakeholder planning problem. Each time step creates a number of unique states and unique actions tied to each state. Given these shortcomings, there are still RL techniques that can be applied to solve a large state and action space problem. [162, 158]

### 4.3.2 Learning Techniques and Approaches

Reinforcement Learning can be used to develop knowledge about an existing decision tree or MDP. In the most basic form, there are two intertwined objectives for RL: the first is to estimate the value of a given state or action and the second is to determine optimal policies.

Optimal policies can be described as a series of optimal action selections through a defined game. Q-learning, and its many variants, exist to solve for a $Q$ function given a policy. A policy can then be determined using the value approximation $Q$.

A $Q$ matrix, or action-value matrix, is trained instead of fully calculating the reward for each and every state by fully characterizing the tree. Each sample of the matrix selects a single state and looks to calculate the next best state and its associated value. The reward is saved then the value is calculated using Bellman's equation (Equation 4.7) for that particular state. This can be done if the Markov property holds true since the previous state does not matter.

$$Q(s,a) = r + \gamma max_{a'} Q(s',a') \tag{4.7}$$

where $s$ is the selected state and $a$ is the selected action, $r$ is the reward for action $a$ at state $s$, $\gamma$ is the discount factor, $s'$ is the state determined based on action $a$, and $a'$ is an action available at state $s'$.

Q-Learning techniques focus on developing a representation of the action-value function [163, 125]. A matrix $Q_{s \times a}$ is developed and can be used to approximate if not solve for the optimum policy. It is possible to be sub-optimal as the number of approximated entries in the $Q$ matrix is determined by the number of samples. A full sampling would yield a perfect knowledge. If a $Q$ matrix is available in multi-player games, a number of common game theory approaches can be applied at each state as players make their action selections (e.g. Nash Q-Learning[164], Minimax-Q Algorithm[164], Friend-or-Foe Q-learning [165], Correlated-Q learning (CE-Z) [166], and Nash bargaining solution Q-learning (NBS-Q) [167] as examples [164]. Given the processing power, each decision can be treated as a single step game given the estimated Q payoff function [165, 125, 163, 168, 156].

Similar methods can be applied to develop a state-value function, or $V(s)$ similar to the action-value function, $Q(s,a)$. Many techniques exist to train a model to represent either.

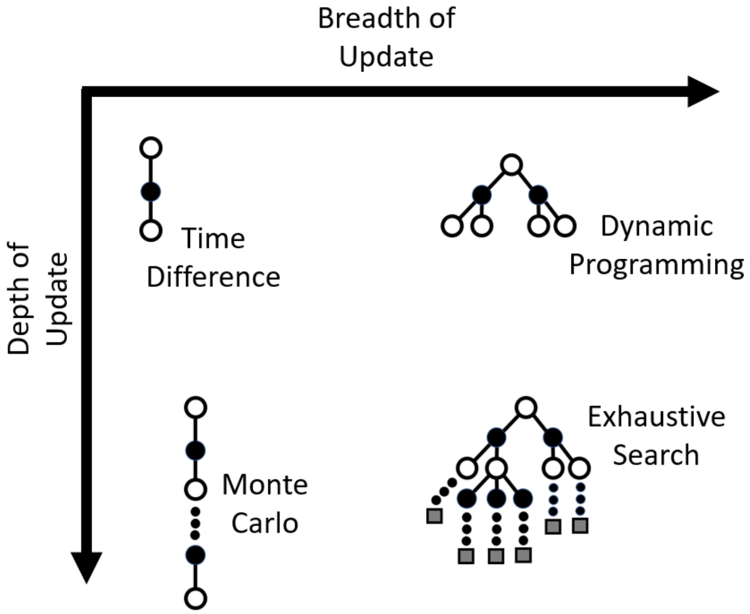


Figure 4.6: Traditional Reinforcement Learning Algorithms [158]

Each technique depends on sampling the MDP, game, or construct in different ways and updating the function under evaluation ($V(s)$ or $Q(s, a)$). Many learning algorithms can be described by the continuum that exists between depth and breadth sampling (Figure 4.6). [158]

Estimating the state-value or action-value for a given player (or stakeholder) is only half of the goal of reinforcement learning. The other half entails developing optimal policies given a set value. Many policy optimization methods exist as do the Q-learning or V-learning methods described above. Traditionally, policy optimization and value optimization are done one at a time in a sequential procedure. Actor-Critic methods use an actor to learn value while a critic simultaneously adjusts the policy to an optimal state. [158]

Lastly, the concepts of policy determination and value approximation can be achieved using function approximation. Many different approaches exist from simple tabular forms to deep neural networks representing actor and/or critic. Convolution Neural Networks can be used to extract features from state spaces and map them to value or policy decisions. Recurring Neural Networks (or Long Short-Term Memory for more stability) can be used establish value or policies for sequential games. Function approximation methods are commonly used amonst the more challenging problems facing reinforcement learning today. [158]

### 4.3.3 State Factorization

A common issue when applying MDPs to real world problems is a substantial increase in state space [169, 170]. The defense stakeholder planning problem characterized in Chapter 2 has a state space that grows substantially with the number of time steps considered. This section explores techniques to reduce the state space given an intractable MDP.

The classic example of factorization is to use a two-stage temporal Bayes net (2TBN) to factor state variable transitions [171, 172, 132]. The 2TBN factorization results in a Conditional Probability Table (CPT) for each state variable to its next state as a function of an action [132]. This helps represent a large state-to-state transition matrix as a set of smaller matrices but does assume independence among state variables. Algorithms have been and continue to be developed for standard value and policy iterations for single and multiple decision makers [173, 174, 175, 176].

The large number of state and action spaces have become a classic issue in multi-agent MDP, POMDP, and Dec-POMDP. Kumar *et al.* takes a value-factorization approach in addition to state-factorization which does not provide enough relief [177]. MDP, POMDP, and Dec-POMDP solution methods have concentrated on previously discussed reinforcement learning methods. The Q-Learning with Adaptive State Segmentation (QLASS) algorithm was designed to help solve large state spaces combined with Q-learning [178]. The QLASS algorithm uses a “sensor subspace” to represent a state space. The subspace is dynamically updated based on samples and highly explored areas. The QLASS algorithm gradually builds the sensor space online until a convergence occurs and is dependent on a given policy selection method (e.g. Boltzmann distribution). [178]

Graph minimization is a necessary part of analyzing large graphs with few state variables and no actions (e.g. social network data). Hamilton looks at examining proximity of nodes to train an encoder that shrinks the overall network to a single encoded node. This method uses common encoding/decoding machine learning techniques. Another approach, when "neighborhoods" are not available, is to begin to contract node by node based on structure (e.g. single string nodes collapsed to a single node). These methods work well for graphs but fail as full state-variables are characterized and as transition probabilities need to be maintained in an MDP reduction.

### 4.3.4 Clustering Similar Decisions and States

Clustering algorithms are used to group similar items based on an $n$-dimensional feature space. Clustering is an unsupervised learning method: there is no target value (class label) to be predicted, the goal is finding common patterns or grouping similar data points [179]. An example of a common clustering method is K-Means Clustering that aims to divide a $n$-dimensional data set into $k$ clusters which commonly minimizes the following Euclidean distance function:

$$f(x) = \underset{j}{argmin} \| x^{(i)} - \mu_j \|^2 \tag{4.8}$$

K-Means can add center points stochastically or deterministically and proceeds to cluster based on minimum distance values (e.g Euclidean, Manhattan, $\cos(\theta)$). K-Means fails on classifying convex data surfaces and will not be able to classify visually obvious clusters. Many other algorithms exist to extract such features and are common in ML libraries today. Significant ongoing research exists in new algorithms applied to clustering subsets [180, 181, 182, 183, 184, 185] (e.g Stochastic/Deterministic, Hard/Fuzzy, Partition/Hierarchical, Agglomeration/Division, Incremental/Non-Incremental [182]). Clustering can be used to find commonality among states, values, and actions across the decision-state trade-space as needed to potentially reduce the dimensionality.

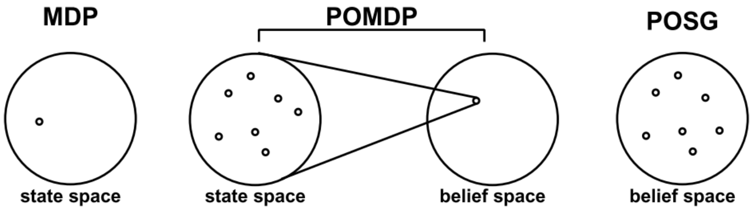


Figure 4.7: Relationship Between MDP, POMDP, and POSG [186]

A base MDP state-space can be clustered and converted to a Baysian game using POMDP techniques built on the POMDP belief-states. The believe-states can be collapsed to generate a more tractable problem than a full POMDP or POSG alone [186]. The state space reduction from state space to believe space is described in Figure 4.7 where the believe-state represents a potential tractable problem. The overall POSG is split into progressive sub-problems and converted to a Baysian game. [186]

### 4.3.5 Dealing with Dimensionality

Three primary solution options were outlined and their positive and negative aspects are noted in Table 4.9. It is clear that traditional MDP solution techniques will not scale appropriately and RL techniques have difficulty in directly representing the variance of outcomes. Ultimately, it is necessary to factor the state space into a subspace, similar to a full state space reduced to a belief space, and apply traditional and RL techniques to the constructed meta-model. A relative scoring of each alternative can be found in Table 4.10.

### 4.3.6 Tractable Representation of Decision Space

Through the observations of Section 4.1 it is clear that the most advantageous approach to reducing the dimensionality is to factor the state space to a subspace. The resulting reduced order MDP will be refered to as the meta-model of the full order MDP. Actions are held constant and transition probabilities are constructed using POMDP techniques. The state-action transitions are appropriatly concerved during the meta-model generation.

Table 4.9: Alternatives for Handling Dimensionality of State Space

| **Alternative** | **Pro** | **Con** |
|---|---|---|
| **MDP Solution Approach** | Standard and well defined solution methods<br>Focused on identifying optimal policy solutions | Fails to scale with states and action space |
| **Reinforcement learning** | Flexible and defined options (e.g. Q-Learning, Actor-Critic)<br>Focused on identifying optimal policy solutions | Fails to maintain direct variance for uncertainty |
| **Factorization** | Shrinks solution space<br>Allows uncertainty distributions to be maintained | Does not directly represent the original MDP<br>Applied evaluations must be re-applied to original MDP |

Table 4.10: Comparison of Dimensionality Handling Techniques

| **Trade-Space Representation Alternatives** | **Standard Solution Method** | **Maintains Uncertainty Impact** | **Scalability** | **Direct Representation of MDP** |
|---|---|---|---|---|
| MDP Solution Approach | ⊙ | ○ | ○ | ● |
| Reinforcement Learning Methods | ● | ○ | ◐ | ● |
| Factorization | ○ | ● | ⊙ | ○ |

⊙excellent, ●good, ◐fair, ○poor, ⊗ none

The approach is fully described in Section 5.3 and is part of the first step in the methodology addressed in this work. The original Truth Modelis sampled using MC methods using exploratory RL techniques. A reduced order state space is used to generate a meta-model MDP. This meta-model is then evaluated and metrics are reapplied to the original state space. The state space compression, evalutation, and remapping leads to the following hypothesis:

> ***Hypothesis 2:*** *If a full MDP is reduced to a meta-model MDP, the resulting risk-based policies generated will preserve the Pareto efficient action determination with reduced computation time.*

## 4.4 Stakeholder Strategy Development and Decision Making

Lastly, the construction and evaluation of the decision trade-space must be used to facilitate stakeholder decision making. A stakeholder must have the information to develop a robust playbook (robust to the significant future uncertainty provided a risk-tolerance level) that can enable decision making over time. This yields the following research question:

> ***Research Question 4*** *How can insights be developed to enable the creation of a playbook that addresses a stakeholder's decisions through an uncertain multi-stakeholder-influenced future?*

The final goal of the methodology is to develop information that can be used to facilitate the creation of a stakeholder playbook. This is done by reducing the epistemic uncertainty regarding actions and resulting possible outcomes. Analysis using the defined evaluation metrics identified (risk-based policies, state-action Return entropy, and opportunity cost) in Section 5.4 results in increased insights. Specifically, individual states and action spaces can be evaluated using the outlined metrics. This provides state based and action based rule set development as a function of stakeholder risk-tolerance. Section 5.5 outlines the development of insights that can be used to develop a risk-based playbook.

The methodology must be benchmarked against state of the art approaches. The rules and recommendations determined via analysis of the evaluation metrics provide additional insights than what is available today. Benchmarking the methodology is equivalent to comparing the provided insights against current methods. Today, optimal policies are used to evaluate decision spaces over time and to develop stakeholder strategies.

The need to provide useful information to playbook generation and benchmark the current methodology yields the following hypothesis:

> ***Hypothesis 3*** *If risked based policies, state metrics, and action metrics derived from the meta-model are used to evaluate the decision space, then more insight is provided to the relevant stakeholders than traditional optimal strategy solutions.*

### 4.5 Summary of Research Questions and Hypotheses

The Research Objective of this work identified the goal to provide a SoS stakeholder with increased information regarding decision making in a multi-stakeholder, multi-objective, and uncertain environment. Three Research Question (RQ) were derived from needs to meet the overarching goal (RQ1, RQ2, and RQ4). RQ1 addresses the need to fully populate the decision trade space. Investigation of RQ1 led to the need for a Truth Model as an assumed input to the developed methodology and using an MDP for Truth Model representation.

RQ2 was motivated by the Research Objective and more specifically by the representation selection. Three evaluation needs were investigated: risk based action evaluation, result volatility evaluation, and opportunity cost evaluation. Risk based evaluation methods were researched and a risk-tolerance based policy development method was selected as the path forward for risk based evaluation. The method provides the needed metrics but comes at a computation cost. The use of the risk-based policy development method results in Hypothesis 1. Hypothesis 1 asserts the risk-based policy development method can provide the identification of Pareto efficient actions. The investigation of volatility and opportunity cost yields two conjectures that are not further explored and become part of an assumed evaluation option.

RQ3 is motivated by the computational cost resulting from the selected risk-based policy development method. Reinforcement learning (RL) and state space reduction techniques were explored. A combination of RL methods and state space reduction was determined to be a viable path forward. RQ3 results in Hypothesis 2 which asserts the solving of a meta-model (reduced order MDP) will result in similar usable results as compared with the solution of full MDP derived directly from the Truth Model.

RQ4 is motivated by the need to provide a usable output to stakeholders and a need to benchmark the methodology outlined in this work. RQ4 results ins Hypothesis 3 which assets that the information developed using the methodology results in more information provided to a SoS stakeholder than an optimal policy solution method.

A summary of the Research Objective, Research Question, and Hypothesis can be found in Figure 4.8. Chapter 5 describes the consolidated methodology developed from the work presented in Chapter 4. Chapter 6 depicts the associated experiments developed to test each hypothesis outlined in Chapter 4.

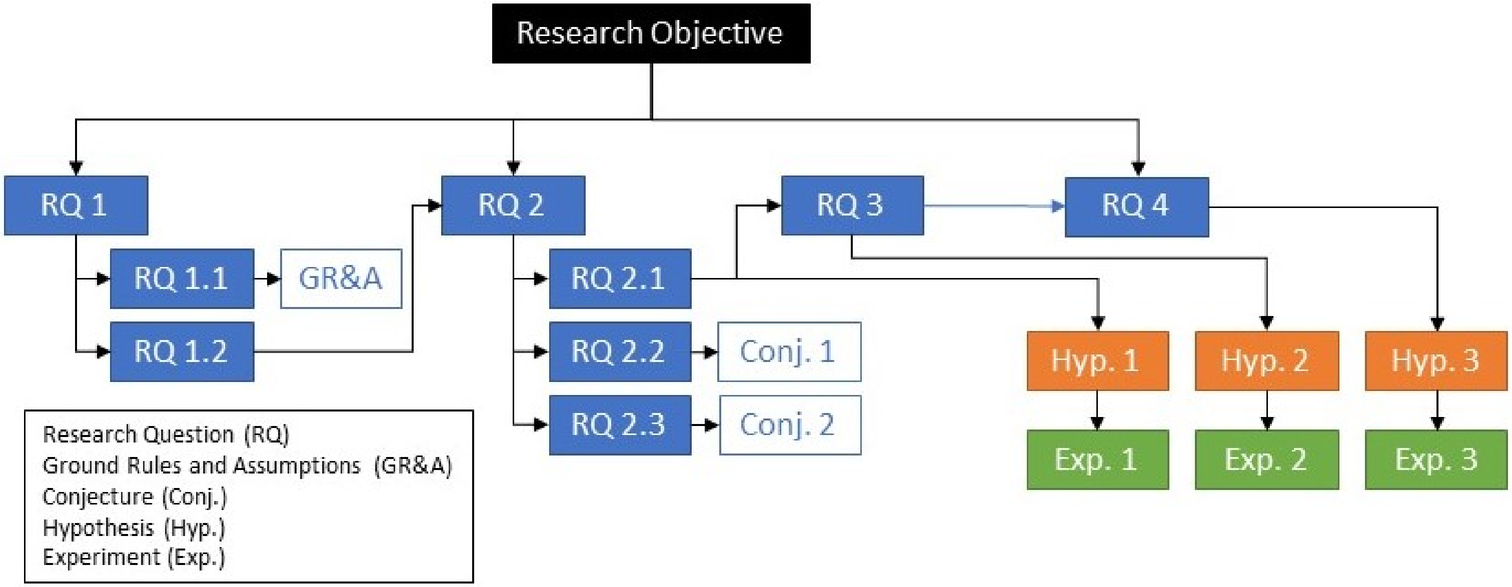


Figure 4.8: Research Questions and Hypotheses

# CHAPTER 5
# METHODOLOGY

## 5.1 Methodology Overview

The methodology consists of a single initialization step (Step 0) followed by three defined steps (Figure 5.1). Step 0 entails setting up the problem and Truth Model for evaluation using the methodology. Step 1 samples the truth model and develops a usable meta-model. Step 2 evaluates the meta-model to produce risk-based policies, state significance metrics, and action significance metrics. Step 3 uses the metrics calculated to better inform the stakeholder of interest and produce rules sets to help generate a stakeholder playbook.

The flow of information between steps is capture in Figure 5.2. Step 1 ingests unique states, unique actions, and tuples based on Truth Model MC sampling. A meta-model characterized by a transition, mean reward, and reward variance matrix are developed and passed to Step 2 along with the unique state and action definitions. Step 2 generates Risk-Tolerance Sensitivity Profiles (RTSP) and other state/action metrics which feed Step 3. The RTSP and metrics are used to generate insights in the form of risk-based state and action based rules. These rules are provided to analysts to inform decision makers.

The Georgia Tech Generic IPPD Methodology (Figure 5.3) outlines a generic decision making process. The process consists of the central column of the process. The outlined generic process can be applied to any decision making problem, including the described force structure planning problem addressed by the outlined methodology. The methodology is mapped to the the generic decision making steps in Figure 5.4. The initial step (Step 0) represents the required inputs and the information that an analyst must produce before using the methodology. This includes establishing the need, defining the problem, and establishing value. These three activities are characterized by defining: stakeholder objec-

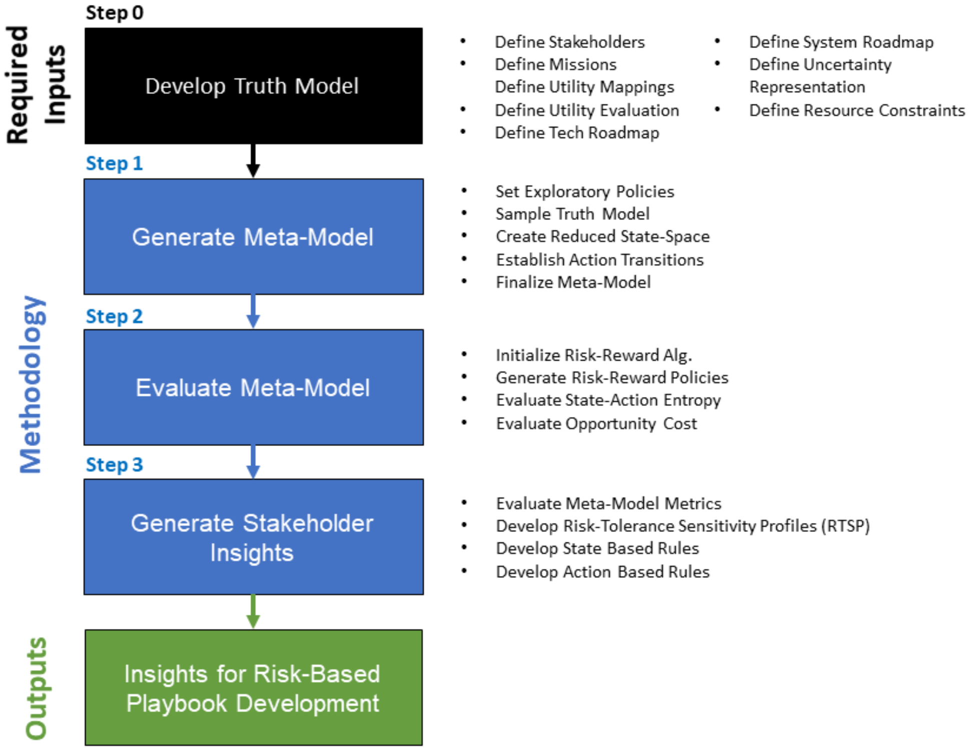

Required Inputs
Step 0
Develop Truth Model
Define Stakeholders
Define Missions
Define Utility Mappings
Define Utility Evaluation
Define Tech Roadmap
Define System Roadmap
Define Uncertainty Representation
Define Resource Constraints
Methodology
Step 1
Generate Meta-Model
Set Exploratory Policies
Sample Truth Model
Create Reduced State-Space
Establish Action Transitions
Finalize Meta-Model
Step 2
Evaluate Meta-Model
Initialize Risk-Reward Alg.
Generate Risk-Reward Policies
Evaluate State-Action Entropy
Evaluate Opportunity Cost
Step 3
Generate Stakeholder Insights
Evaluate Meta-Model Metrics
Develop Risk-Tolerance Sensitivity Profiles (RTSP)
Develop State Based Rules
Develop Action Based Rules
Outputs
Insights for Risk-Based Playbook Development

Figure 5.1: Methodology Overview

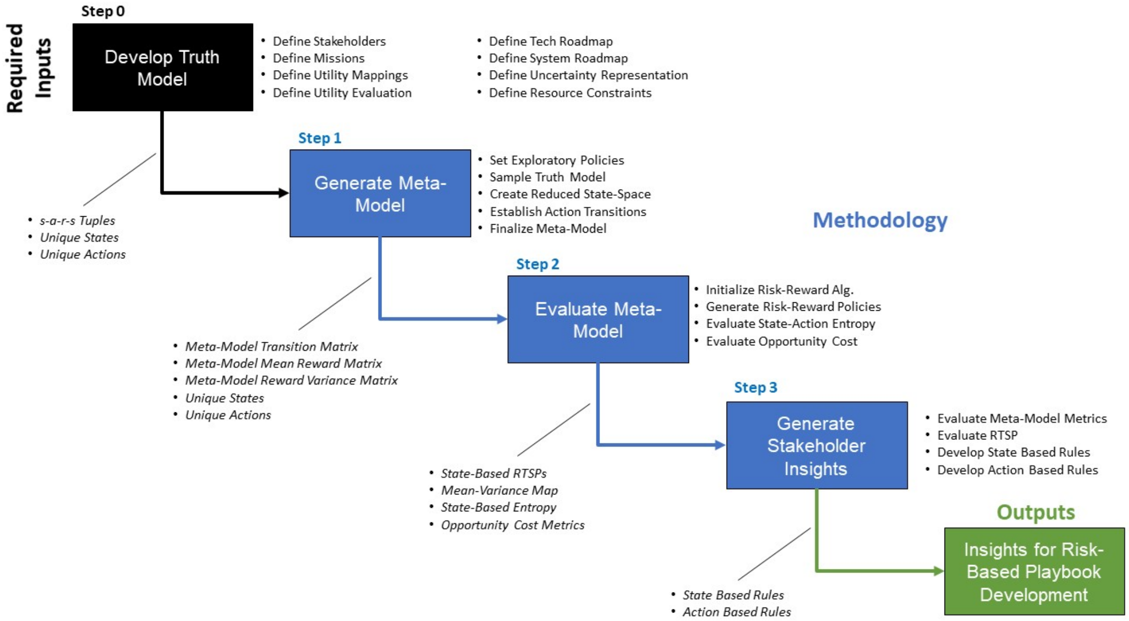

Required Inputs
Step 0
Develop Truth Model
Define Stakeholders
Define Missions
Define Utility Mappings
Define Utility Evaluation
Define Tech Roadmap
Define System Roadmap
Define Uncertainty Representation
Define Resource Constraints
s-a-r-s Tuples
Unique States
Unique Actions
Step 1
Generate Meta-Model
Set Exploratory Policies
Sample Truth Model
Create Reduced State-Space
Establish Action Transitions
Finalize Meta-Model
Methodology
Meta-Model Transition Matrix
Meta-Model Mean Reward Matrix
Meta-Model Reward Variance Matrix
Unique States
Unique Actions
Step 2
Evaluate Meta-Model
Initialize Risk-Reward Alg.
Generate Risk-Reward Policies
Evaluate State-Action Entropy
Evaluate Opportunity Cost
State-Based RTSPs
Mean-Variance Map
State-Based Entropy
Opportunity Cost Metrics
Step 3
Generate Stakeholder Insights
Evaluate Meta-Model Metrics
Evaluate RTSP
Develop State Based Rules
Develop Action Based Rules
Outputs
Insights for Risk-Based Playbook Development
State Based Rules
Action Based Rules

Figure 5.2: Methodology Data and Information Flow

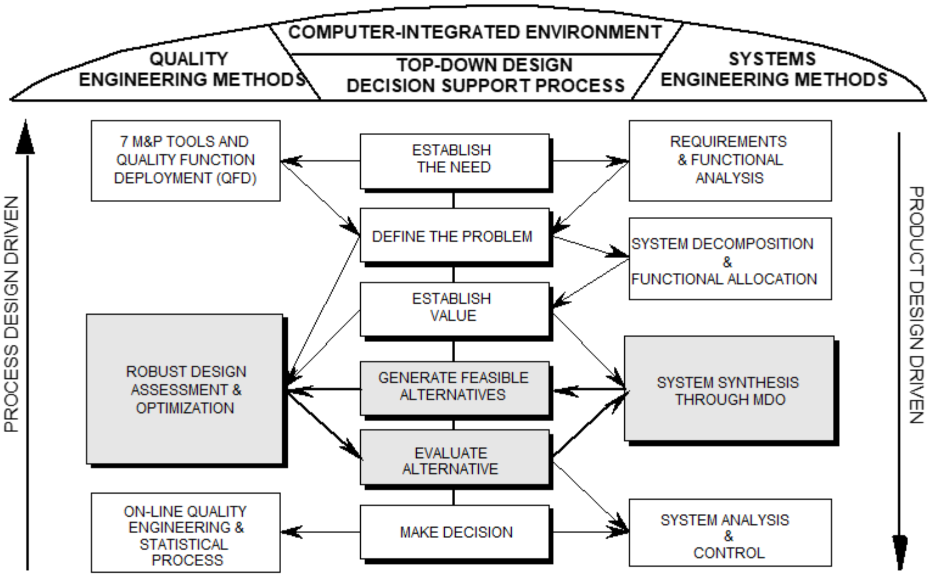

COMPUTER-INTEGRATED ENVIRONMENT
QUALITY ENGINEERING METHODS
TOP-DOWN DESIGN DECISION SUPPORT PROCESS
SYSTEMS ENGINEERING METHODS
PROCESS DESIGN DRIVEN
PRODUCT DESIGN DRIVEN
7 M&P TOOLS AND QUALITY FUNCTION DEPLOYMENT (QFD)
ESTABLISH THE NEED
REQUIREMENTS & FUNCTIONAL ANALYSIS
DEFINE THE PROBLEM
SYSTEM DECOMPOSITION & FUNCTIONAL ALLOCATION
ESTABLISH VALUE
ROBUST DESIGN ASSESSMENT & OPTIMIZATION
GENERATE FEASIBLE ALTERNATIVES
SYSTEM SYNTHESIS THROUGH MDO
EVALUATE ALTERNATIVE
ON-LINE QUALITY ENGINEERING & STATISTICAL PROCESS
MAKE DECISION
SYSTEM ANALYSIS & CONTROL

Figure 5.3: Georgia Tech Generic IPPD Methodology

tives, stakeholder priorities, stakeholder constraints, mission scenarios, available systems, and mission measurements. The next activity is to generate feasible alternatives. Step 1 creates a meta-model that captures the full decision space of an individual stakeholder constrained by resources and availability. Next, the feasible alternatives are evaluated. In step 2, the decision space is fully evaluated using Risk-Tolerance Sensitivity Profiles and other state/action metrics. Finally, a decision is made based on the evaluation. Step 3 supports the last activity by generating action and state based rules. The final output is a set of risk-based state and action based rule sets. These rules sets help provide decision makers and help cull the feasible alternatives based on a stakeholder's risk-tolerance.

## 5.2 Step 0: Defining the Truth Model

Step 0 defines the Truth Model to be used to sample future scenarios. The Truth Model allows future scenarios to be simulated starting from a fixed initialization point. Ultimately, the Truth Model has requirements to provide information used in Step 1. Specifically, it is required to provide sequential states, actions, and rewards.

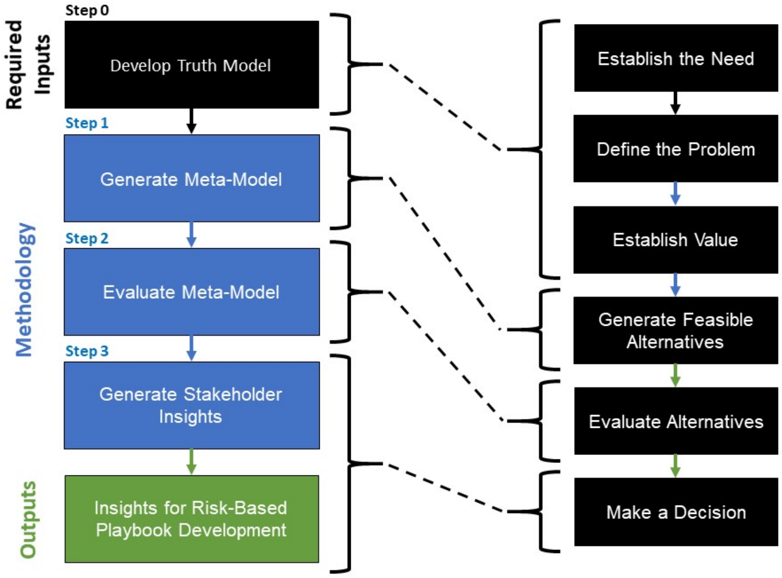


Figure 5.4: Methodology and Decision Making

The Truth Model can be described in the context of the conceptual system of system evolution cycle outlined in Chapter 2, Figure 2.6. The simulation must provide an open loop cycle of stakeholder decisions, decision impacts in SoS state, and the military utility provided to each stakeholder by each SoS (Figure 5.5)). The simulation should begin with a set of SoS states for each stakeholder. Each stakeholder should have a define set of decisions that can be made. Given resource constraints (e.g. budgets), stakeholders can decide to spend on asset creation, invest in technology, and allocate assets to specific SoS, or missions. The SoS will be defined by the state determined by current and previous stakeholder decisions. The current state of the SoS are evaluated against their respective missions.

Each described function does not need to be part of a monolithic simulation. The selection of asset and technology investment can be guided technology evaluation method (Section 3.2). The architecture selection and evolution could use Adaptive SoS Architecture Evolution (Section 3.8). Allocations don’t have to be set deterministically and can be selected at random. During sampling of the Truth Model it is essential that all decisions can

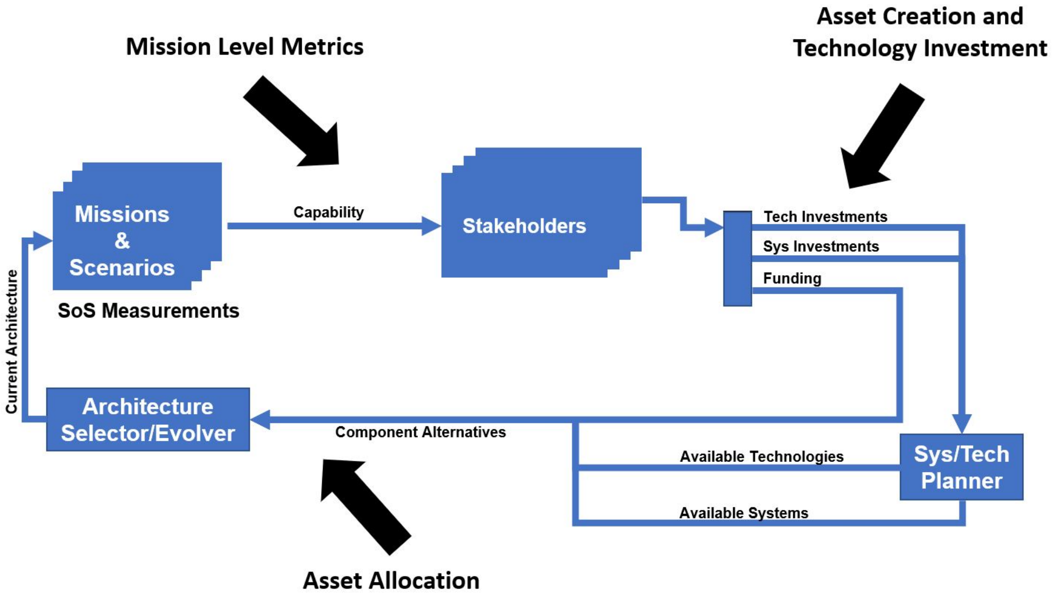


Figure 5.5: Conceptual Truth Model Open Loop Representation

be explored in a stochastic manor. The Truth Model is required to provide the capability to stochastically select decisions in addition to using stochastic models (Figure 4.2). This allows the exploration of decisions during sampling instead of using a pre-determined set of rules or optimization to make stakeholder decisions.

The Truth Model must provide three specfic outputs based on the decision exploration policies used during sampling (Figure 5.6). The truth model must provide discrete actions, discrete states, and stakeholder utility with a clearly defined chain of cause and effect. This data is used during the sampling process to create state and action based samples. The samples are then used to create a reduced meta-model.

A consideration can be given to both quantitative and qualitative metrics. The concept of Value-Driven Design should be applied to ensure the multi-level connections from subsystem to mission are made as well as the impacts of non-performance metrics on stakeholder utility (e.g. economic).

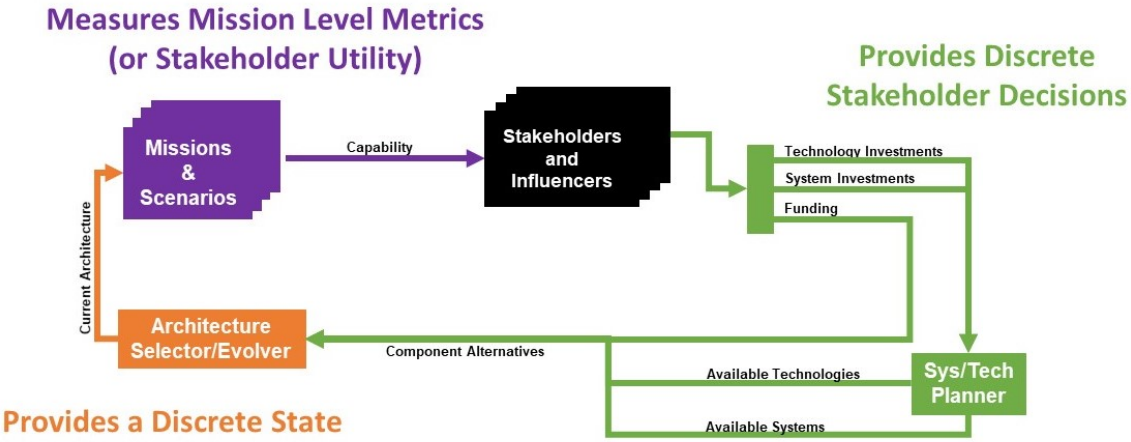


Figure 5.6: Truth Model Cycle Mapped to State, Action, and Reward Sampling

## 5.3 Step 1: Generating the Meta-Model

Step 1 of the methodology intakes the Truth Model developed in Step 0 and outputs a reduced state space Markov Decision Process (MDP), or meta-model. The Truth Model is first sampled using the Monte Carlo method to generate state-action-reward tuples, a unique state record, and a unique action record. The records are used to reduce the size of the state space creating a meta-model state space. The sampled tuples are used to produce a MDP using the reduced state space. The unique state records, unique action records, the full to reduced state space mapping, and the reduced MDP are then used in Step 2 for evaluation.

### 5.3.1 Truth Model Sampling

The sampling method used is a depth based Monte Carlo method (Figure 4.6) . The Truth Model is repeatedly sampled with the same initial conditions and a complete single simulation, or episode, is executed through a pre-defined simulation time. The constant initial conditions produce a fixed starting point for all episodes. Each episode produced a set of depended states, actions, and rewards. A single episode yields a chain of potential state-action pairs ($s_0, a_0 \rightarrow s_1, a_1 \rightarrow ... \rightarrow s_{T-1}, a_{T-1} \rightarrow s_T, a_T$) where $T$ is the number of time

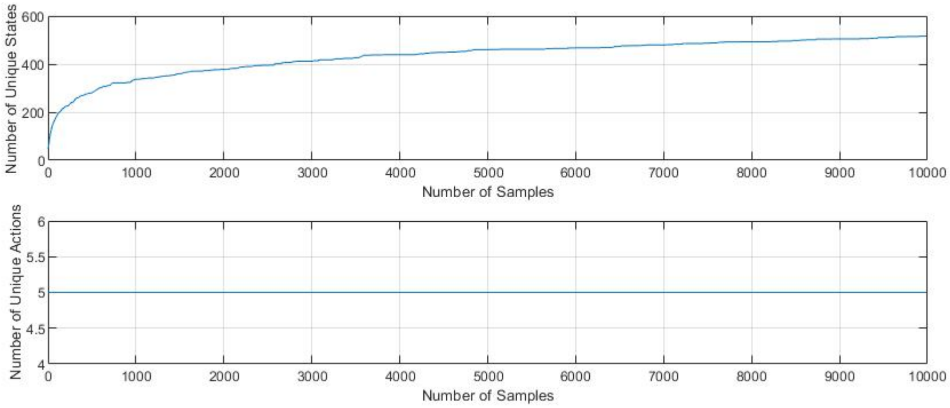


Figure 5.7: Number of States and Actions as a Function of Episode Samples

steps) all starting with a fixed $s_0$. There are three key pieces of information gathered during each episodes: states, actions, and rewards.

The intuitive problem introduced in Section 2.6 can be used as an example. The sampling metrics from the sampling of the Truth Model are shown in Figure 5.7. The actions, as expected, increase very quickly to five total actions in accordance with the breakdown of the decisions space (‘Wait’, ‘Acq Sys 1’, ‘Acq Sys 2’, ‘Dev Sys2’, ‘Dev Sys1’). The number of unique states begins to decrease as the re-sampling of states increases.

The reward as a function of state and action is recorded. It can be viewed as a function of time step (Figure 5.8). The simple reward versus time plot demonstrates the large amount of noise that uncertainty introduced. A view by time step provides another perspective (Figure 5.16.

Each state is defined by a quantitative characteristic vector. The state characteristic vector represents all information captured for future use. It can encompass temporal characteristics (e.g. time step, time in development for a system), asset quantities, and asset allocations. The decision can be made to incorporate or not incorporate portions of the recorded vectors. For example, if a stakeholder of interest may not have knowledge of an opponent stakeholder’s asset quantity then it can be disregarded during sampling. This allows the evaluation to be made on partial or full state knowledge. This allows either a

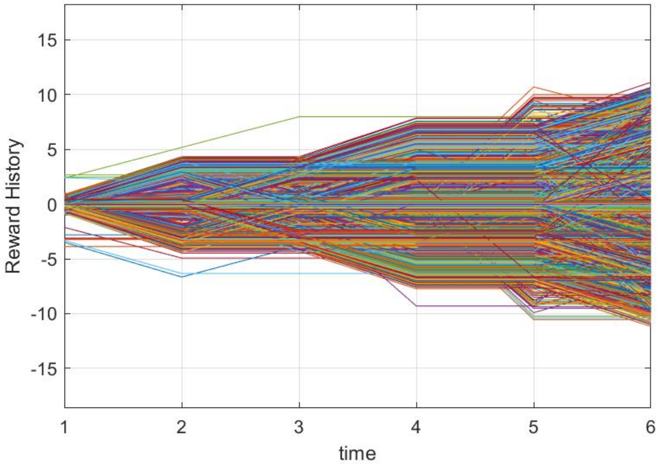


Figure 5.8: Reward versus Time Step

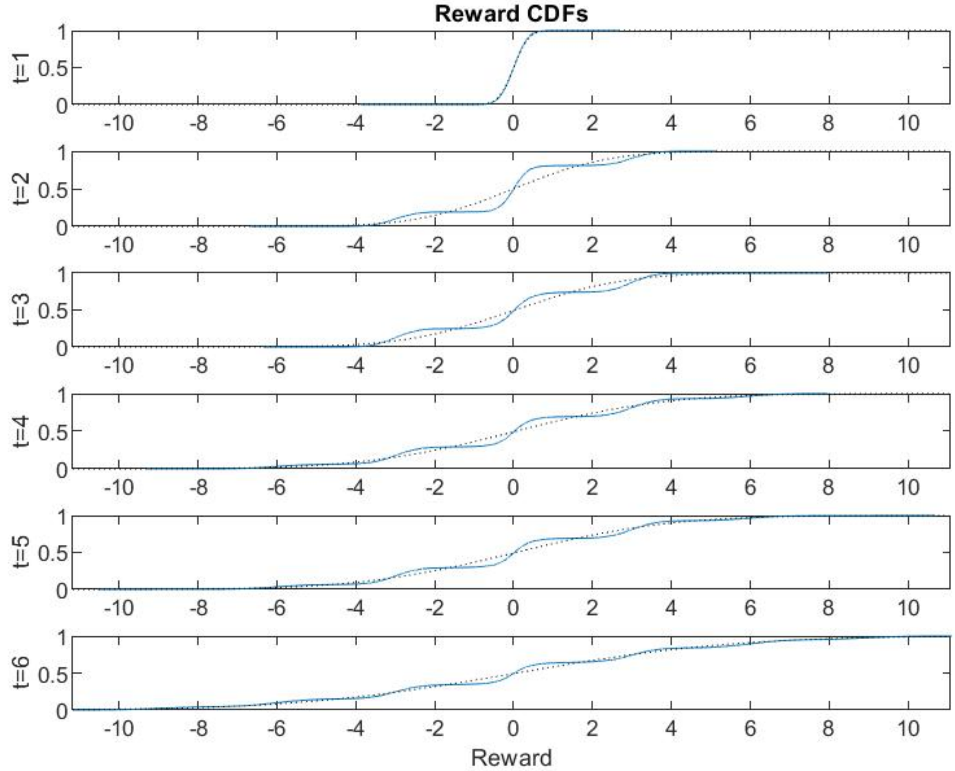


Figure 5.9: CDF of Reward by Time Step

single common state vector or stakeholder unique state vectors to be produced and used. Any simulation state reached resulting in the same state characteristic vector will be viewed as the same state through out usage in the methodology.

Each action is defined using quantitative and Boolean characteristics. The action characteristic vector has the capability to uniquely define decisions made by stakeholders in the Truth Model. A unique action vector is produced for each stakeholder individually. The action characteristic vector can included asset creation decisions records as a Boolean (e.g. decision to develop a specific system, decision to acquire a system) and as cardinal numbers (e.g. number of systems to be acquired, number of assets allocated to a specific mission). A specific action or set of actions may be available at any given state.

The development of the initial reward can be measured directly from the simulation for each stakeholder or can be developed during sampling as a composite of available simulation outputs. At each time step the Truth model will produce evaluations of specific mission level metrics. The mission level metrics (independent of a stakeholder) are the result of a given state (e.g. allocated systems by all stakeholders, available budgets). The mission level metrics at each state can be used to develop individual stakeholder utility metrics. Each state has a unique mission level metric profile and derived individual stakeholder utilities produced by the simulation. The stakeholder utility is mapped as the reward a stakeholder receives when reaching a specific state. The Reward is defined as a specific single step return.

A single episode path through the simulation yields a dependent string of states, actions, and rewards. The episode state-action-reward path is translated into a set of individual $(s, a, s', r)$ tuples. Each tuple acts as a single transition measurement. Each tuple sample is aggregated into the an going sample metrics generated after each episode is run. The aggregated sampling metrics are:

- Sum of Rewards Matrix ($\mathbf{r}_{s \times s \times a}$)

- Sum of Squares of Rewards Matrix ($\mathbf{ss}_{s \times s \times a}$)

- Number of Samples Matrix ($\mathbf{n}_{s \times s \times a}$)
- Unique State Vector Set ($\mathbf{s}_{sampled}$)
- Unique Action Vector Set ($\mathbf{a}_{sampled}$)

  where $s$ is the number of unique sampled states and $a$ is the number of unique sampled actions.

### 5.3.2 Reducing State Space

The state space reduction can use any of the sampled metrics created from sampling the Truth Model. The goal of the state space reduction is to reduce the number of overall states while maintaining a relevant model in evaluating the original Truth Model scenario. This is done by clustering states based on characteristics and maintaining consistency with the original sampled data.

The first step to reducing the state space is performing a clustering analysis on the unique state vectors generated during Truth Model sampling. Each individual application of the methodology is unique and can require tailoring of the clustering space. Specifically, it can be useful to add the available actions or additional temporal factors to the state space prior to clustering. Ensuring applicable variables are used to cluster the state space is essential as the complexity of the application increases.

The applications in this work use the unique state characteristics and their available actions as inputs to the clustering algorithm. An agglomerative hierarchical clustering is used to allow the impact of variable magnitude characteristics while ensuring the division of the state space is always at the most impactful point in the state space. The size of the reduction is specified in the total number of clusters. Each cluster will become a unique state in the meta-model. Each sampled state ($s$) has a many-to-one mapping to the new reduced state space ($s \rightarrow s'$). The number of clusters selected represents the total number of states to be present in the meta-model. The ratio of the number of clusters, or meta-model states, to the unique sampled states yields the state space compression ratio (Equation 5.1). A mapping between the uncompressed state space and the meta-model states is generated as an output of the clustering analysis.

$$\kappa = \frac{N_{states,meta-model}}{N_{states,sampled}} \tag{5.1}$$

The sample metrics are aggregated together based on the meta-model states identified via the clustering algorithm. The sample metric matrices ($\mathbf{r}_{s\times s\times a}$, $\mathbf{ss}_{s\times s\times a}$, and $\mathbf{n}_{s\times s\times a}$) are aggregated together based on the state space mapping to a new set of sample metrics ($\mathbf{r}'_{s'\times s'\times a}$, $\mathbf{ss}'_{s'\times s'\times a}$, and $\mathbf{n}'_{s'\times s'\times a}$).

The aggregation of the sample metrics results in the following sets of aggregated and sampled data which is used to construct the reduced state-space MDP:

- Aggregated Sum of Rewards ($\mathbf{r'}_{s\times s\times a}$)
- Aggregated Sum of Squares of Rewards ($\mathbf{ss'}_{s\times s\times a}$)
- Aggregated Number of Samples ($\mathbf{n'}_{s\times s\times a}$)
- Unique State Vector Set
- Unique Action Vector Set
- Sampled State Space to Reduced State Space Mapping ($\mathbf{s}_{reduced} \rightarrow \mathbf{s}_{sampled}$)

The aggregated and sample data is specific to an individual stakeholder point of view. The action selection and reward functions are stakeholder unique under all conditions. The state space may also be tailored to individual stakeholder's based on the anticipated knowledge expected.

### 5.3.3 Generating Reduced-Dimensional Meta-Model

The meta-model is a Markov Decision Process with added reward variance. The meta-model is generated using the reduced state-space sample metrics. The meta-model consists of the following attributes:

- Transition Probability Matrix ($\mathbf{T}_{s' \times x' timesa}$)
- Reward Mean Matrix ($\mathbf{R}_{\mu, s' \times x' \times a}$)
- Reward Variance Matrix ($\mathbf{R}_{\sigma^2, s' \times x' timesa}$)

The meta-model is generated using the $observations$ aggregated together into $\mathbf{r'}_{s \times s \times a}$, $\mathbf{ss'}_{s \times s \times a}$, and $\mathbf{n'}_{s \times s \times a}$. The meta-model is produced by identifying (1) the transition probabilities as a function of action and (2) the reward matrices based on the reduced state space sample metrics.

A transition probability matrix mapping a given $(s, a)$ pair to a next state $(s')$ is generated using Equation 5.2. The transition probability is based on the ratio of all observed $(s, a)$ pairs to those that explicitly lead to $s'$. The number of observations ($N - obs$) of a given starting state, action, or ending state are present in the $\mathbf{n'}_{s \times s \times a}$ sample metric. The actions at this stage represent a single stakeholder. The selection and result of other stakeholder actions are by definition absorbed into the probability of transition.

$$T(s, s', a) = P(s'|s, a) = \frac{N_{obs}(s, s', a)}{N_{obs}(s, a)}, s \in \mathbf{s}_{reduced} \tag{5.2}$$

The inclusion of the reward variance matrix is essential to the algorithms used to evaluate the meta-model and is a primary reason the state space reduction is necessary. Splitting the reward into a mean and variance allows for uncertainty to be captured and accounted but at the expense of additional evaluation complexity. The individual reward mean ($r_\mu(s, s', a)$) and reward variance ($r_\sigma(s, s', a)$) are calculated for each state-action-state tuple to develop both of the Reward matrices ($\mathbf{R}_\mu$ and $\mathbf{R}_{\sigma^2}$). The Reward mean matrix is

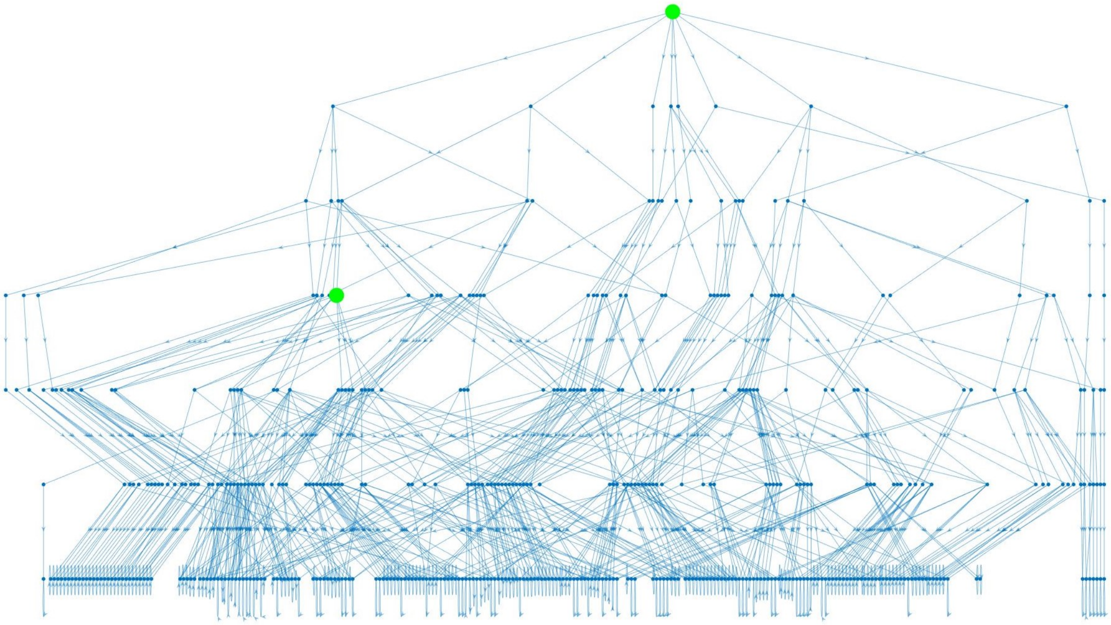


Figure 5.10: Example MDP Structure Graph with States 1 and 13 Highlighted

populated using Equation 5.3. The Reward variance matrix is populated using Equation 5.4.

$$\mathbf{R}_{\mu}(s, s', a) = \frac{\mathbf{r}'(s, s', a)}{\mathbf{n}'(s, s', a)}, s \in \mathbf{s}_{reduced} \tag{5.3}$$

$$\mathbf{R}_{\sigma^2}(s, s', a) = \frac{\mathbf{ss}'(s, s', a) - \frac{\mathbf{r}'^2(s,s',a)}{\mathbf{n}'(s,s',a)}}{\mathbf{n}'(s, a, s') - 1}, s \in \mathbf{s}_{reduced} \tag{5.4}$$

The final output of the meta-model generation is a MDP. The structure of the resulting graph can be viewed in two ways. The first is a structure graph that does not account for actions and only accounts for transitions (Figure 5.10). This can be considered ‘action collapsed’.

The second method is to break apart the MDP by action and display the transitions when a given action is available (Figure 5.11). These two view points give an understanding of the structure of the MDP. For example, at step 1, the primary action is to ‘Wait’. This is due to acquisitions and developments conducted in the initial state requiring all resources and multiple time steps. Additionally, more branches require the stakeholder to 'Wait' after 'Dev Sys 2' in the initial state. Lastly, 'Acq Sys 2' is only available in branches that have already 'Dev Sys 2'.

## 5.4 Step 2: Evaluating the Meta-Model

The meta-model generated during Step 1 is used to explore each individual meta-model state. This exploration which would not be feasible with a full MDP accounting for every possibility from the Truth Model. This step (1) develops risk-reward based policies for stakeholders, (2) identifies significant decision points, and (3) evaluates the opportunity cost of actions at each state. The output of Step 2 associates significance and opportunity cost to each state enabling identification of states of interest. The risk-based policies generated in Step 2 enable further evaluation in Step 3. The policies enable the generation of Risk-Tolerances Sensitivities Profiles (RTSP). Ultimately, these metrics and evaluations enable the generation of stakeholder playbooks.

### 5.4.1 Risk Based Policy Generation

In traditional model-based reinforcement learning, a policy ($\pi^h$) describes the probability that an agent (a stakeholder, $h$ in this problem description) will select an available action when in a specific state. An optimal policy can be seen as a 100% probability of selecting a single action at each state. A policy is defined by Equation 5.5 and has an entry for every state and action pair. The sum of all policy entries for a given state must equal 1.

$$\pi(s, a) = P(a|s) \tag{5.5}$$

where $\sum_{a \in A} \pi(s, a) = 1$ for any $(s)$ and available actions $(A)$

A risk-based policy is generated as a function of a risk tolerance metric ($\xi, [-1, 1]$) and a stakeholder of interest ($h$). The final policy is generated based on a set of policy iterations.

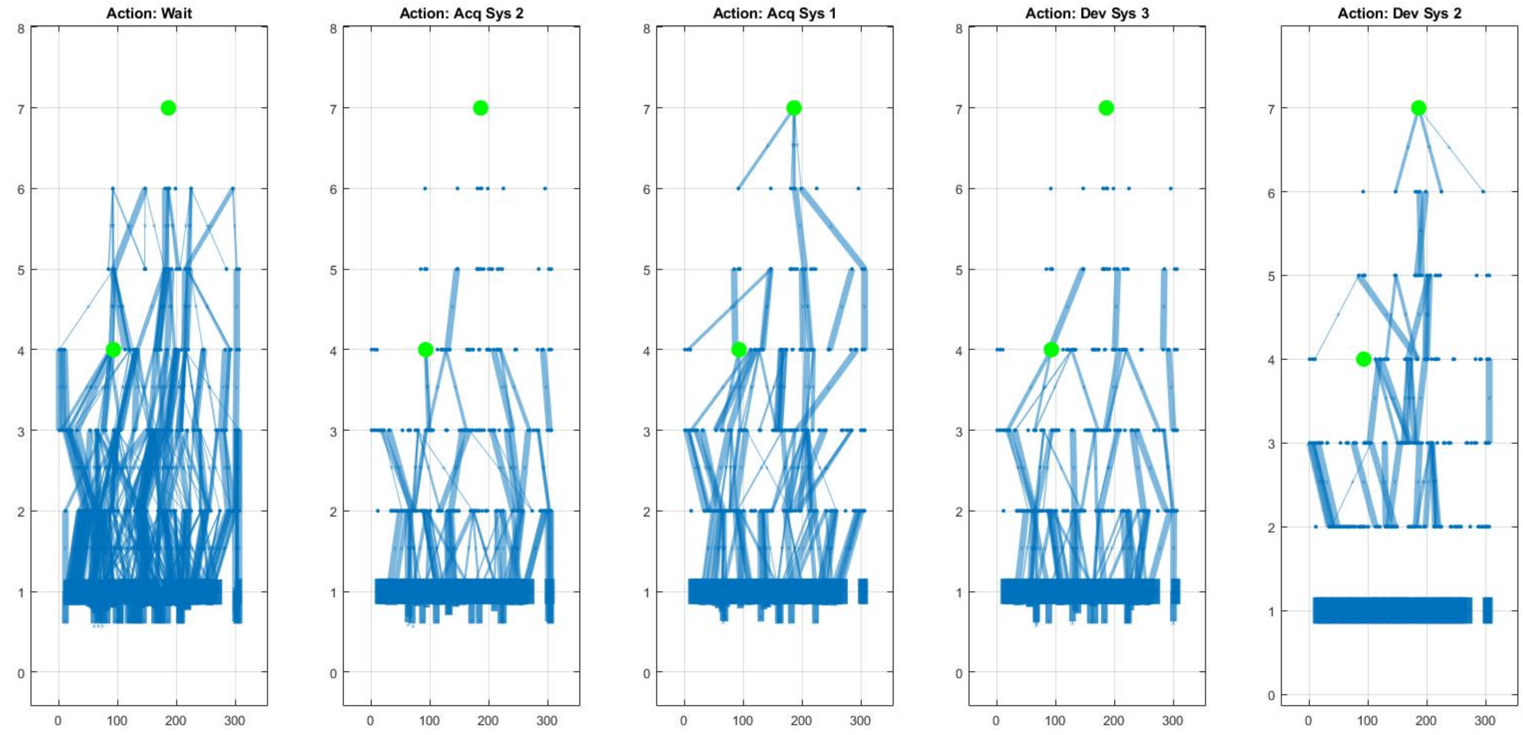


Figure 5.11: Example Action Graph with States 1 and 13 Highlighted

At each iteration, the Return mean and variance are calculated for each action based on the current policy. Pareto efficient portfolios defined by the relative weighting of individual actions are determined. A relative action weighting is determined based on the selected $\xi$. The relative weightings re used to update the current policy for use in the next iteration or for final output. The algorithm is captured in Algorithm 1 and described in more detail below.

**Algorithm 1** Risk Based Policy Algorithm

```
Set Risk-Tolerance ($\xi$)
Set sampling time horizon ($t$)
Set Return discount ($\alpha$)
Set learning rate as a function of iteration ($\gamma$)
$\pi_h^\xi \leftarrow \pi_0$                              ▷ Initialize the stakeholder policy
repeat
    for Each Stakeholder do
        Collapse action-space to Stakeholder
        for Each $s \in S$ do
            for Each $a \in A$ do
                Collapse the action space
                Run N samples $t$ steps in time
                Calculate Return sample mean (expected Return)
                Calculate Return sample variance (Return volatility)
            end for
            Establish mean-variance Pareto frontier
            Calculate relative action weightings ($w$) as a function of $\xi$
            $\pi_h^\xi(s) \leftarrow (1-\gamma) * \pi_h^\xi(s) + \gamma * w$
        end for
    end for
until policy convergence or max iterations reached
```

A single iteration of the algorithm is first walked through to provide a more in depth description. Then, the iteration-to-iteration behavior in the context of convergence of the algorithm is described.

Each iteration has an initial policy ($pi$), time horizon ($t$), policy learning rate ($\gamma$), Return discount ($\alpha$) and Risk-Tolerance level ($\xi$) provided as an input. This policy is updated through out the iteration. The output policy of the iteration acts as the input policy to the next iteration or as the final output policy if convergence (or a maximum iteration number) is reached. The other inputs are used through out the policy iteration process and are individually described in more detail below.

During an iteration, each state is addressed individually from leaf states to the initial state. Each action available in the current state is evaluated. The long term expected Return and Return volatility is measured for each action. A set of $N$ samples is MC episodes are run starting in the initial state ($s_0$) and selecting the action of interest ($a_0$). Each MC episode is run to the defined time horizon, $t$. Each episode yields a string of $sarsa$ samples, $s_0, a_0, r_1, s_1, ..., r_t, s_t$.

The samples are based on the current policy and will progressively change as the policy is updated. When the sampling is done, the policy (action selection) is combined with the transition matrix (transition probability given a starting state and action) to yield a direct state to state transition probability (Equation 5.6).

$$P(s|s') = \sum_{a \in A} T(s'|s,a)\pi_h^{\xi}(s,a) \tag{5.6}$$

A Return is calculated for each $sarsa$ sample string. Traditionally the sum of discounted Rewards (Equation 5.7) is used to calculate the Return. The discount ($\alpha$) variable sets the relative impact of future Rewards on the current Return. Traditionally, the discount variable is kept less than 1 ($\alpha \in (0,1]$. The impact of future rewards therefore decreases as time from the initial state increases.

$$R = \sum_{i=0}^{i=t} \alpha^i r_i \tag{5.7}$$

Alternative discount approaches exist. A non-traditional approach is to use an $\alpha > 1$. This allows future Rewards to more heavily impact the current Return but does introduce potential instability at long time horizons. Additionally, a customized weighting profile can be used to replace the $\alpha^i$ term in Equation 5.7. A relative weighting based on time from the current state enables the selections of a specific impact profile. An absolute weighting relative to the initial state of the MDP under evaluation provides targeted time based impacts of relative reward for the specific stakeholder under consideration.

The sampling of each action for a state of interest results in a distribution of future Returns as a function of the action selected. The expected Return for each action ($R_{\mu}^{\pi,s_0}(a)$) is computed using Equation 5.8 for each action. The Return volatility is computed computed using Equation 5.9 for each action.

$$R_{\mu}^{\pi,s}(a) = \mathbb{E}[R^{\pi,s_0}(a)] = \frac{\sum\limits_{o \in O(s,a)} R^{\pi,s}(o)}{N_{obs}(a)}, s \in \mathbf{s}_{reduced}, a \in \mathbf{A} \tag{5.8}$$

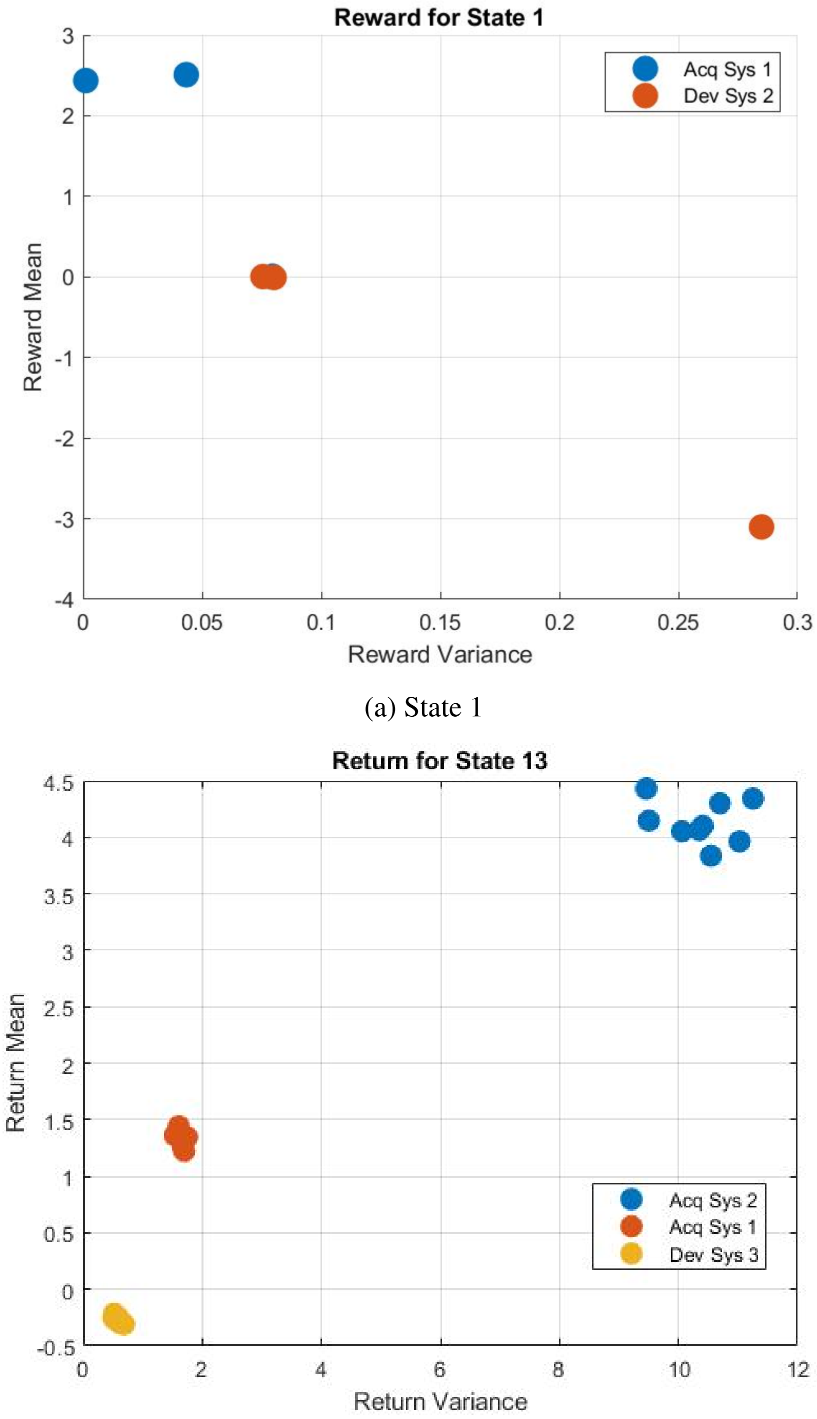
where the sample set, $O$, terminated at time $t$, seeded at state $s$ with action $a$

$$R_{\sigma^2}^{\pi,s}(a) = var[R^{\pi,s_0}(a)] = \frac{\sum\limits_{o \in O(s,a)} (R^{\pi,s}(o) - R_{\mu}^{\pi,s}(a))^2}{N_{obs}(a) - 1}, s \in \mathbf{s}_{reduced}, a \in \mathbf{A} \tag{5.9}$$

where the sample set, $O$, terminated at time $t$, seeded at state $s$ with action $a$

The expected Return and Return Volatility for each action yield a single mean-variance point for each action. The mean-variance plot (or mean-variance map) can visually depict the relative Return performance across the action space of the state of interest. An example mean-variance map for a State 1 and State 13 of the intuitive problem are shown in Figure 5.12.

A Pareto frontier is then developed based on the available mean-variance Return of each action. Portfolios of actions ($p_a$) are generated to form the Pareto efficient frontier. The process is similar to stock portfolio design using the expected mean and variance of future individual investments. A full efficiency frontier can be visualized in Figure 5.13. Note that in this application both the traditional higher-mean and lower-variance frontier and the lower-mean and lower-variance frontier are highlighted. The upper half of the



(a) State 1

(b) State 13

Figure 5.12: Example Return Mean-Variance Maps

efficiency frontier represents realistically feasible alternatives of non-dominated solutions. The lower half of the efficiency frontier represents fully dominated solutions or solutions that do not dominate other solutions.

The Pareto frontier highlighted in Figure 5.14 is constructed by combining available actions using a paramaterized weighting vector:

$$\mathbf{w} = \begin{pmatrix} w_2 \\ w_2 \\ \vdots \\ w_{m-1} \\ w_m \end{pmatrix}$$

where $m$ is the available number of actions and $\sum_{i=1}^{m} w_i = 1$.

Action portfolios are defined by the relative weighting vector ($w(p_a)$). The expected Return and Return volatility of an individual action portfolio are calculated using Equation 5.10 and Equation 5.11 respectively.

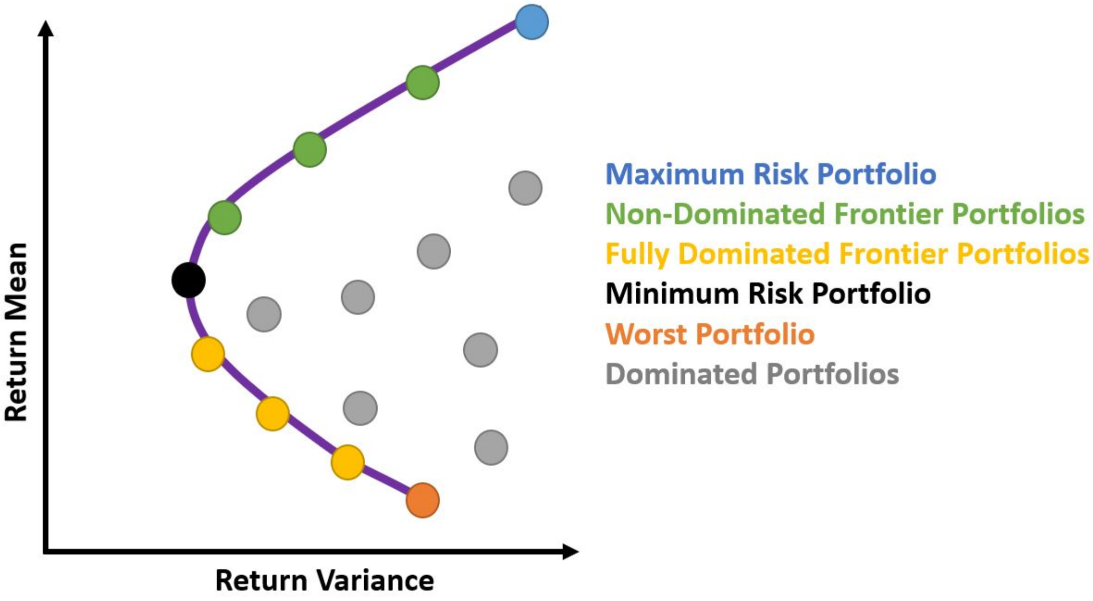
$$\mu(p_a) = \sum_{i=1}^{m} w_i(p_a) R_{\mu}^{\pi,s}(a), s \in \mathbf{s}_{reduced}, a \in \mathbf{A} \tag{5.10}$$

$$\sigma^2(p_a) = \sum_{i=1}^{m} w_i^2(p_a) R_{\sigma^2}^{\pi,s}(a), s \in \mathbf{s}_{reduced}, a \in \mathbf{A} \tag{5.11}$$

Each point on the frontier represents a single action portfolio constructed using the defined weighting vector. Selecting a point along the frontier is equivalent to selecting an action portfolio and a relative weighting. The frontier path can be paramaterized by a Risk-Tolerance ($\xi$). A visual depiction of the paramaterization is shown in Figure 5.13. The Risk-Tolerance parameter describes a stakeholder's desire to accept risk and is defined $[-1, 1]$.

There are three key positions along the full frontier:

Figure 5.13: Risk-Tolerance Paramaterization of the Pareto Frontier

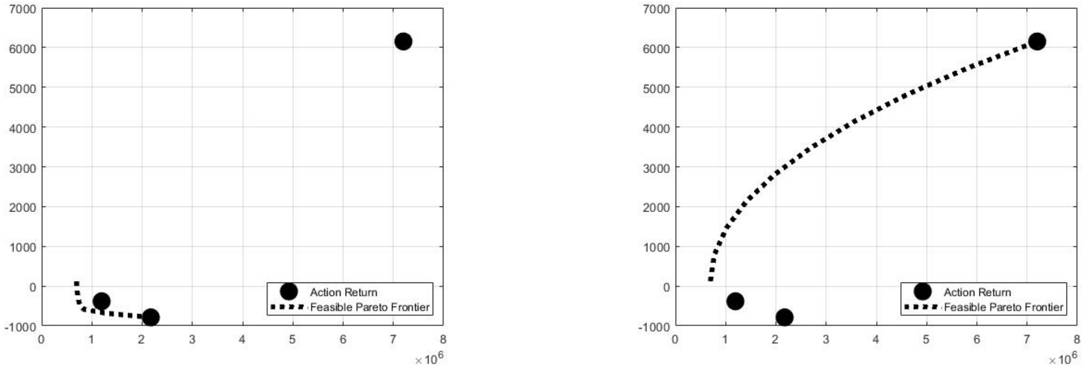


- Highest Risk Point ($\xi = 1$)

- Minimum Risk Point ($\xi = 0$)

- Worst Point ($\xi = -1$)

The point of highest risk is represented by a Risk-Tolerance of 1. This action portfolio has the highest-variance and highest-mean Return of all action portfolios. The point of minimum risk is represented by a Risk-Tolerance of 0. The minimum risk action portfolio has the least-variance of any other portfolio achievable. All action portfolios represented by $\xi \in [0, 1]$ are realistic options and feasible Pareto efficient alternatives. These action portfolios remain non-dominated by all other potential action portfolios. No other action portfolio will result in a higher mean and lower variance in Return. For these action portfolios, as the risk is increased (Return variance) the expected Return increases (mean Return). A feasible trade-off between risk and return exists. Examples from State 13 for the realistically feasible and unrealistic non-feasible Pareto frontiers are shown in Figure 5.14.



(a) $\xi \in [-1, 0)$

(b) $\xi \in [0, 1]$

Figure 5.14: State 13, Three Action, Pareto Frontiers

The worst action portfolio is represented by a Risk-Tolerance of -1. The worst action portfolio is characterized by the lowest-mean and highest variance. The portfolio has the most Return volatility with the lease expected Return. This represents the worst relative weighting of actions. All action portfolios represented by $\xi \in [-1, 1)$ represent full dominated options. Each portfolio has no other portfolio that has a lower mean and lower variance. This section of the frontier represents non-feasible alternatives. As the risk is increased (Return variance) the expected Return decreases (mean Return). A non-feasible negative feedback between risk and return exists for the fully dominated action portfolios.

A action profile is selected based on the input Risk-Tolerance which is held constant across all policy iterations. This yields a relative weighting vector selection as a function of Risk-Tolerance for the state of interest. The policy for the state of interest is updated using the Risk-Tolerance selected weighting vector (Equation 5.12).

$$\pi_h^{\xi, i+1}(s) = (1 - \gamma(i)) * \pi_h^{\xi, i}(s) + \gamma(i) * w_i \tag{5.12}$$

where $i$ is the current policy iteration number.

The policy update concludes the single state specific evaluation and is applied across all MDP states during each policy iteration. Note that $\gamma$ is a function of the policy iteration number and must be set as a decreasing function to guarantee convergence. Example functions include exponential (Equation 5.13) and geometric (Equation 5.14).

$$\gamma(i) = 1 - e^{\frac{i_{max}-i}{i_{max}}} \tag{5.13}$$

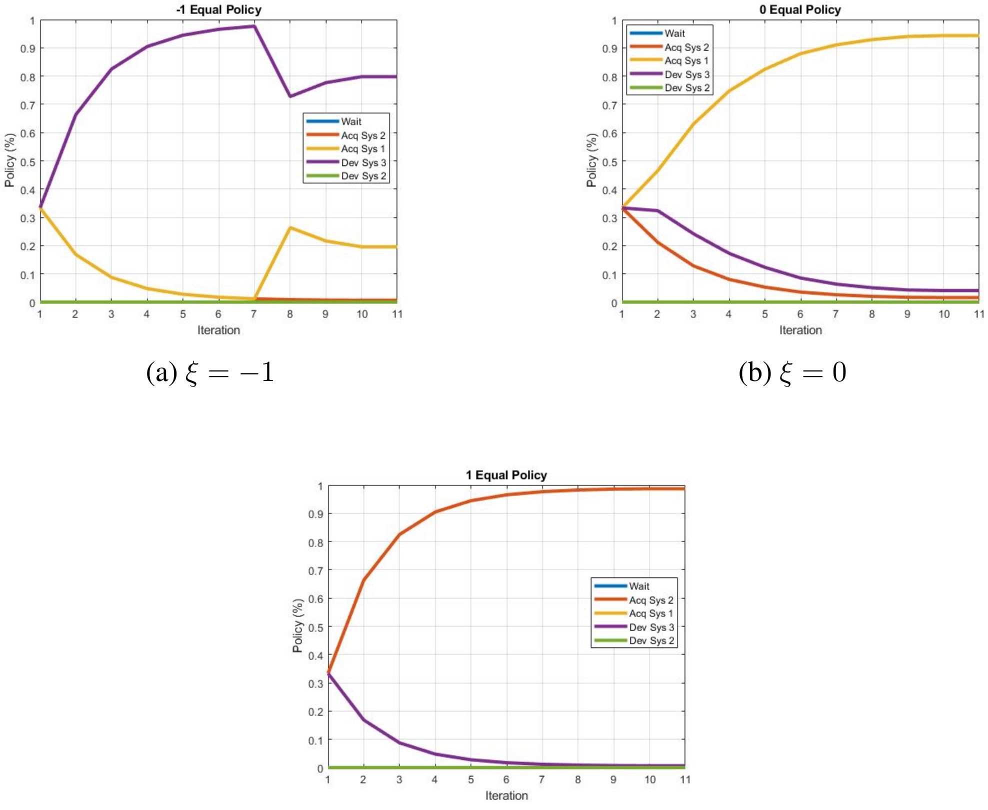
$$\gamma(i) = \frac{1 - (\frac{i}{i_{max}})^2}{2} \tag{5.14}$$

The decreasing influence of each iteration helps convergence of the policy. An example of policy convergence with an initial equal policy (all actions weighted equally) across varying Risk-Tolerances is shown in Figure 5.15. The gradual convergence on the final policies is clearly shown for available actions. Unavailable actions in State 13 equal zero for all iterations. The varying convergence points as a function of Risk-Tolerance gives an initial look at the sensitivity of risk-based policies to the Risk-Tolerance of a stakeholder.

The result of the risk-based policy algorithm is a single policy matrix that has an entry for each reduced state and action representing a probability of selection based on the selected Risk-Tolerance ($\pi^{\xi}_{s\times a}$). The risk-based policies can be paramaterized as a function of Risk-Tolerance. The paramaterized policies are used in Step 3 to assist in stakeholder decision making by providing more significant information than the selected optimal action at each state.

### 5.4.2 Evaluating Decision Significance

The significance of a single believe state, $s$, is determined by evaluating all $(s, a)$ pairs available under a specified polity, $\pi$. Each action is evaluated based on the entropy of future states for a set finite time horizon. Samples of the meta-model are made using the risk based policies starting with the state-action pair of interest as the initializing starting point. The samples are used to calculate an entropy, $E$, at a finite time horizon for each $(s, a)$ pair. The entropy is compared across all $a$ for a given $s$.



(a) $\xi = -1$

(b) $\xi = 0$

(c) $\xi = 1$

Figure 5.15: Policy Convergence Examples for State 13

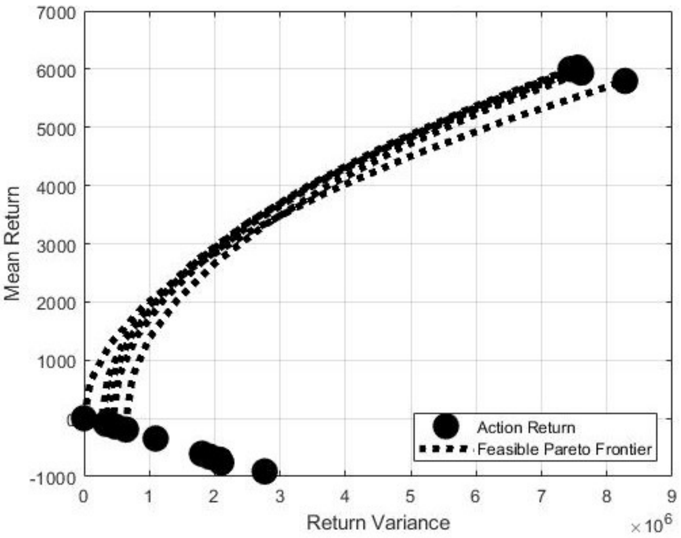
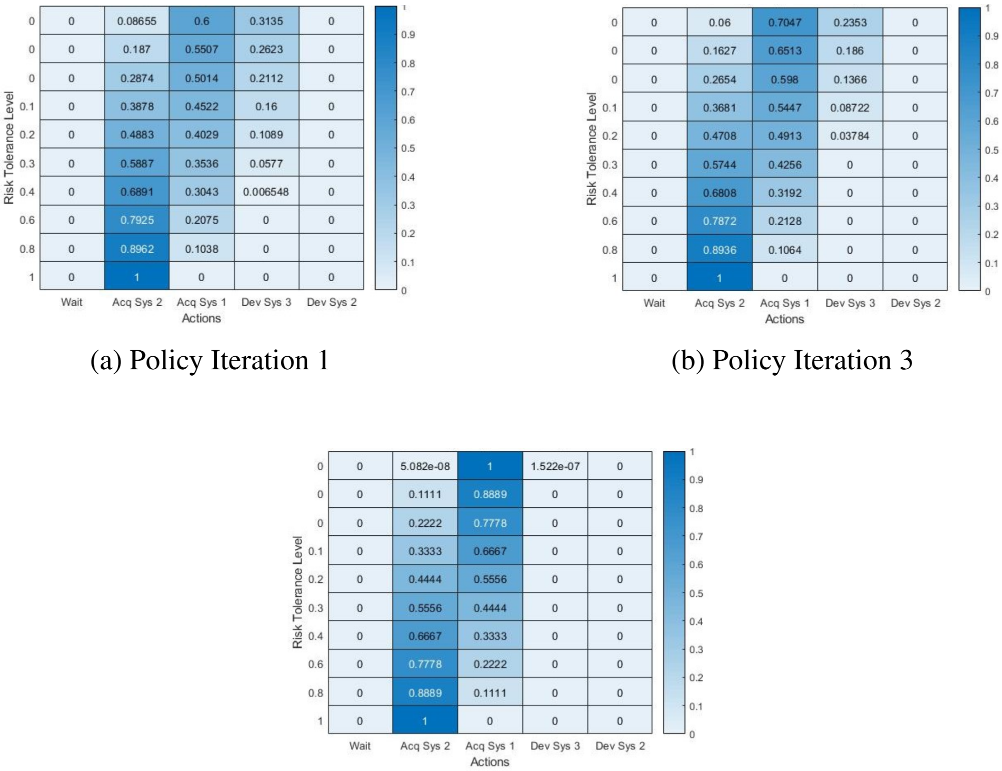


Figure 5.16: Pareto Frontier Convergence for $\xi = 1$



(a) Policy Iteration 1

(b) Policy Iteration 3

(c) Policy Iteration 5

Figure 5.17: Action Weighting Vectors as a Function of Iteration State 13

$$E(\pi, s, a) = \sum_{o \in O(s,a)} -P_{V_{bin,h}}(o) log_2 P_{V_{bin,h}} \tag{5.15}$$

where $O(s_b, a_h, t)$ are the observations at time horizon $t$ with state action predecessor $(s_b, a_h)$.

A similar entropy across all $a$ for a given $s$ shows little impact difference between actions. A significant difference in entropy between available actions identifies a key decision point due to the variation in the future. This can help identify key states, actions, and state-action pairs during Step 3 policy development.

### 5.4.3 Evaluating Opportunity Cost

Similar to the significance determination, the meta-model and risk-based policies are used to evaluate each state-action pair's cumulative reward at a finite time horizon. Instead of utilizing the overall utility of each stakeholder, $V_h(s)$, the utility for individual missions is used (Equation 5.16 and 5.17.

$$\mu_{V_h(s)} = \mathbb{E}[V_h(s)] = \frac{\sum\limits_{o \in O(s)} V_h(o)}{N_{obs}(s)}, s \in \mathbf{S}_b \tag{5.16}$$

$$\sigma^2_{V_h(s)} = var[V_h(s)] = \sqrt{\frac{\sum\limits_{o \in O(s)} (V_h(o) - \mu_{V_h(s)})^2}{N_{obs}(s) - 1}}, s \in \mathbf{S}_b \tag{5.17}$$

Significant differences in the mean or variance of individual mission metrics demonstrate that a specific action within a given state will yield an either-or choice to a stakeholder. Figure 5.18 demonstrates an example comparison of two actions in a single state-action set.

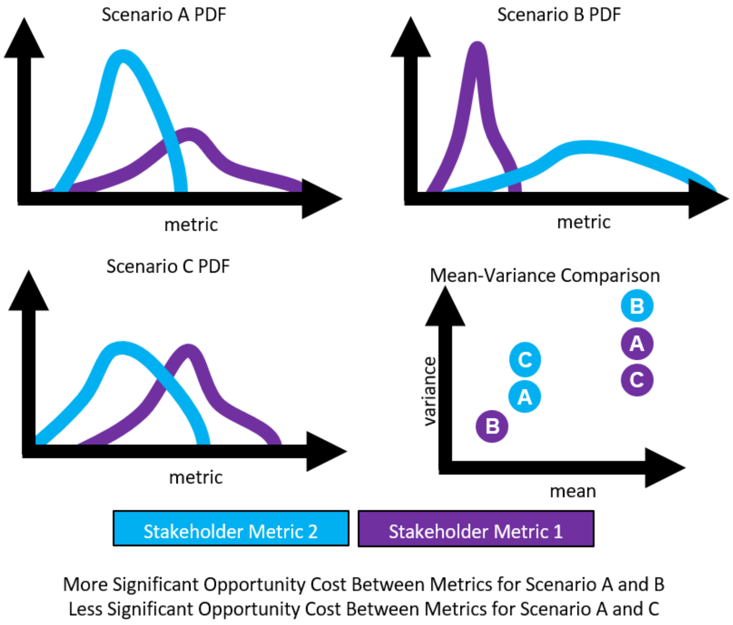


Figure 5.18: Opportunity Cost Indicator Example

## 5.5 Step 3: Generate Stakeholder Insights

The evaluation methods used in Step 2 to evaluate the meta-model generated in Step 1 produced the data necessary to provide evaluation of the stakeholder decision space. Using the metrics from the evaluation step allows the culling of decisions under specific situation. The metrics can be used to generate rules and paths forward that can guide a stakeholder in both present day and through future events. Four methods of generating stakeholder insights are presented and their potential impact on the generation of a stakeholder playbook.

The meta-model state based metrics are mapped from the reduced state space to the full state space using the $s_{sampled} \rightarrow s_{reduced}$ mapping and an action mask. The action mask allows only action metrics from the reduced space that were present in the full state space for each individual state. This includes state-action entropy, state-action return mean, state-action return variance, and state policies.

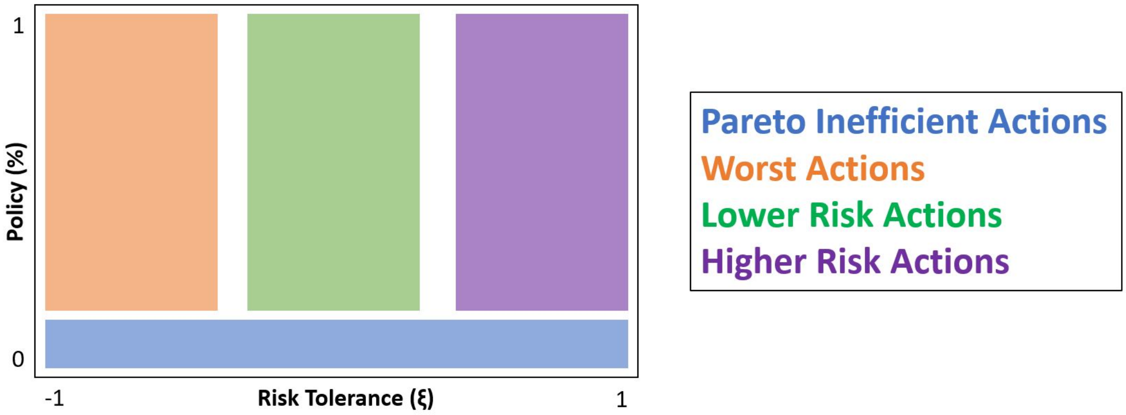
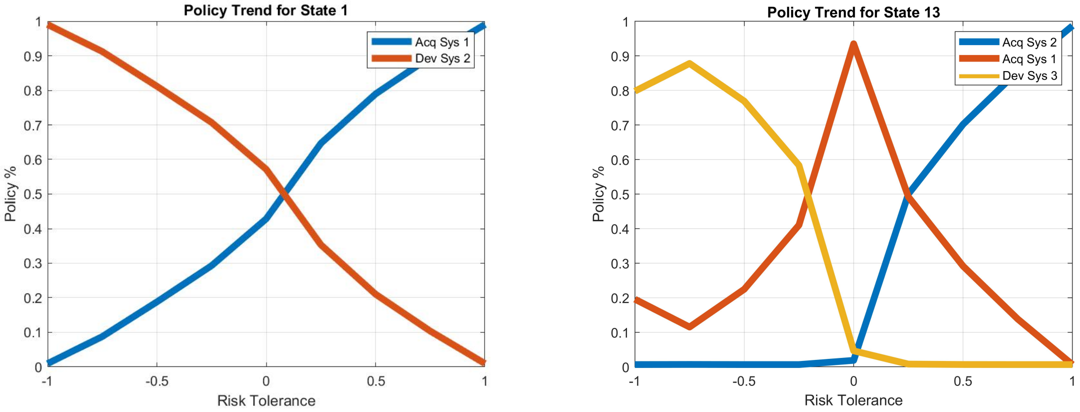


Figure 5.19: Interpreting Risk-Tolerance Sensitivity Profile

### 5.5.1 Risk-Tolerance Sensitivity Profile Analysis

Risk-Tolerance Sensitivity Profiles (RTSP) are generated for each state using the risk-based policy algorithm described in Section 5.4. The profiles represent the policy trends as the risk-tolerance level is varied. A set of risk-based policies are generated for risk-tolerances from $\xi = -1$ to $\xi = 1$. The policy trends are plotted against risk-tolerance. The profiles allow insight beyond what is gained from a single optimal policy (Figure 5.19).

The first piece of information that can be extracted are the Pareto inefficient actions. The Pareto inefficient actions will not peak as a the risk-tolerance is varied. Their contribution to the policy will remain low and without significant trends. The second piece of information is the identification of actions that make up the low-mean and high-variance returns, the worst actions. These actions represent those that should be avoided. This set of actions represents non-productive actions. The final piece of information is the identification of Pareto efficient and productive actions. These actions are characterized by higher policy contributions for a risk-tolerance of $0 < \xi < 1$. These action sets are actions sets representing low to high risk future outcomes. A selection of a risk-tolerance for a stakeholder will yield a productive policy. A stakeholder with a lower risk-tolerance will want to look more towards a $\xi = 0$ path where a stakeholder with a high risk-tolerance will want



(a) State 1 Risk-Tolerance Sensitivity Profile

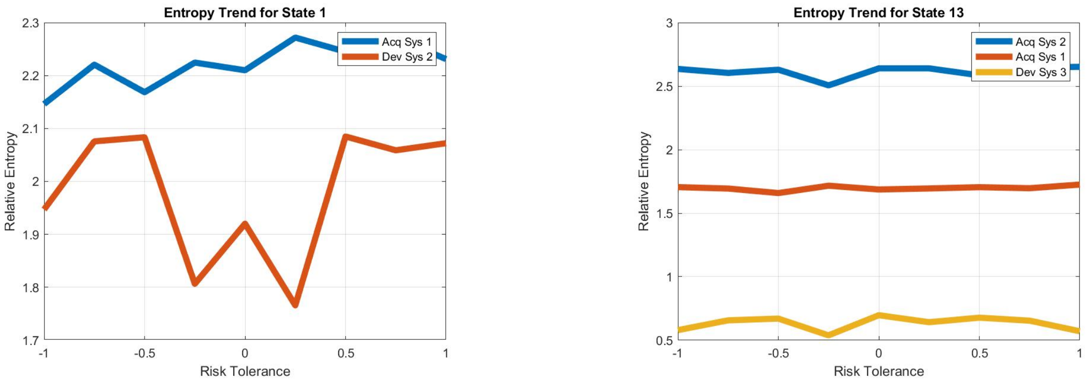
(b) State 13 Risk-Tolerance Sensitivity Profile

Figure 5.20: Example Risk-Tolerance Sensitivity Profiles

to look towards a $\xi = 1$ policy.

Two examples of RTSP are depicted in Figure 5.20. The first RTSP plot (Figure 5.20a) depicts State 1 with two actions available. The 'Acquisition of System 1' action peaks at $\xi = 1$ where 'Develop System 2' peaks at $\xi = -1$. The development action falls in the worst action category. The acquisition action falls in the productive action category. The second example RTSP (Figure 5.20b) is for a three-action state, State 13. A similar patter seen in State 1 can be seen in State 3 between 'Develop System 3' and 'Acquire System 2'. The new aspect is a third action which peaks near $\xi = 0$. Developing System 3 is a non-productive action and should not be selected. There is then a trade-off between which system to acquire, 1 or 2. System 1 will yield lower risk outcomes and System 2 will yield higher risk outcomes.

A more complete description of the interpretation of an RTSP is done in Appendix C. Selected inputs and RTSP outputs are used to deep dive into the calculation and interpretation of simple and complex RTSPs. Each of the examples ties back to selected experiment setups and subsequent results.



(a) State 1

(b) State 13

Figure 5.21: Example Entropy Measurements

### 5.5.2 State-Action Entropy Evaluation

In addition to the RTSP, the relative entropy of each action can be measured and compared. The entropy is a product of a policy and is therefore measured by state, action, and policy. Higher entropy means there is more variation in future states once that action is taken. It should be note that a higher variance in Return does not always directly manifest as a higher entropy across all risk-tolerance levels. Two examples of risk-tolerance are depicted in Figure 5.21. The entropy trend for the two-action State 1 shows a lower entropy for the non-productive action and a higher entropy for the productive action (Figure 5.21a). It is possible that the development action could possess a lower mean and a higher variance that acquisition action. This would correlate to a higher entropy in the productive state than the non-productive state. State 13 entropy (Figure 5.21) depicts the highest risk action as the highest entropy action. The development action has a much lower entropy. This identifies a significant decision point. The low entropy action shows little changes in the future if that action is selected. A high Return and low entropy state will tend to guarantee positive results. A non-productive state with low entropy should be avoided as it locks in a stakeholder in poor track.

### 5.5.3 State-Action Return Mean-Variance Map

For a given policy, the Return mean and variance is measured for each state-action. This allows the long term Return to be analyzed as a function of risk-tolerance and action for a given state. The resulting map will yield a reference for the generation of the RTSP graphs and help identify trends, differences, and anomalies. Figure 5.12 is an example of a Return map for a specific state. The black markers with a centered white dot represent the $\xi = -1$ points. The black markers with a centered white 'x' represent the $\xi = 1$ points. Each dot in between represents discrete $\xi$ steps in between. Each line corresponds to a separate action available at the given state. A more specific interpretation relative to the RTSPs is depicted in Appendix C.

### 5.5.4 Decision Space Analysis

Thus far, decision evaluations have been state-centric. Grouping states together based on available actions can yield a more action based evaluation method. A decision space is defined as a set of states with the same, or similar, available actions. Evaluating an action set can establish action set based rules (e.g. general preferences or non-preferences regardless of state). Variations in action preference yield additional information. Variations in action preference can be correlated to state differences and state based rules can be developed for action selection. Figure 5.22 represents a look at states that have the same three actions as State 13. State 13 is only one of many states that make up this three action decision space which is characterized by the 'Acquire System 1 or Acquire System 2 or Develop System 3' action set. In this example, there is a clear trend aligning to the observations previously made for the RTSP for State 13.

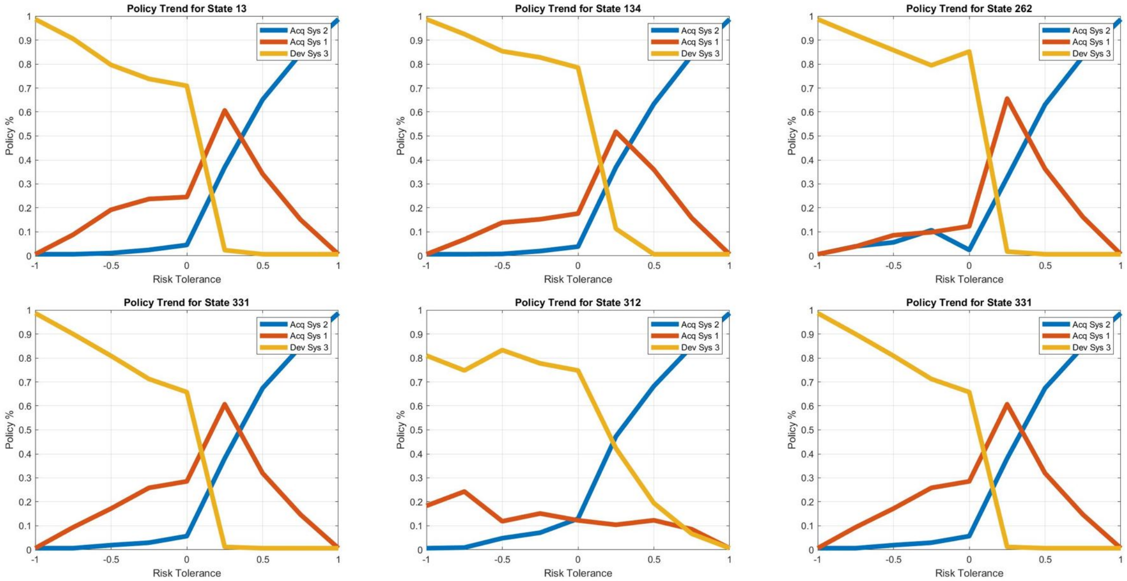


Figure 5.22: Example Decision Space Perspective

## 5.6 Output: Insights for Playbook Development

The final output of the methodology is are increased insights for stakeholders to help develop a stakeholder playbook or rule set. This rule set acts as a guide to sequential strategic decision made in real time. There are two types of rules that can be created: state based rules and action based rules. Individual state RTSP, entropy, and return mean-variance based analysis yield rule sets for specific states a stakeholder may find themselves. The decision space analysis yields action based rules that provide guidance despite the specific state or as a function of specific state variables.

Two states and one action set were analyzed for the example problem outlined in Section 2.3.1 throughout the methodology overview in Chapter 5. The following derived rules set can be prescribed from the analysis presented:

- In State 1, the stakeholder should select to Acquire System 1 over Developing System 2.

- In State 13, the stakeholder should never choose to Develop System 3.

- In State 13, the stakeholder should select Acquiring System 1 if they are risk adverse.

- In State 13, the stakeholder should select Acquiring System 2 if they are more risk tolerant.

- When selecting between Acquiring System 1 and Developing System 2, the stakeholder should always choose to Acquire System 1.

- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should never choose to Develop System 3.

- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should select Acquiring System 1 if they are risk adverse.

- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should select Acquiring System 2 if they are more risk tolerant.

# CHAPTER 6
# EXPERIMENTS

Each Experiment Set is designed to test a single hypothesis. A mapping between the proposed methodology, identified research questions, hypotheses, and experiments are shown in Figure 6.1. Experiment 1 demonstrates the risk-based policy generation methods ability to produced risk varying policies against varying complexity input scenarios. Experiment 2 compares the evaluation results from the reduced meta-model against the full MDP derived from the Truth Model to demonstrate the usability of the reduced model. Experiment 3 compares the information generated by the risk varied policies and the optimal policies to generate an increase in information. Additionally, Experiment 3 benchmarks the full methodology against the current solution method (optimal policy based analysis). Experiment 3 results in an example rules set that can be used to generate a stakeholder playbook.

Each Experiment Set consists of a subset of experiments denoted by an alphabetic sequenced character (e.g. Experiment Set 1a, Experiment Set 2b). Each subset is selected based on a significant contribution to testing a hypothesis and results in a varied experimental setup. Each subset may be decomposed into Cases. A Case represents a discrete change in a significant experimental parameter or setup definition. Each Case contributes stand alone knowledge toward proving or disproving a hypothesis. Cases are composed of Scenarios which varying lower level experimental settings and are the lowest discritization of an Experiment. Each Experiment and it's decomposition are defined by the experimental setup and the selected variables at each level. Table 6.1 provides a summary of all experiments described in Chapter 6.

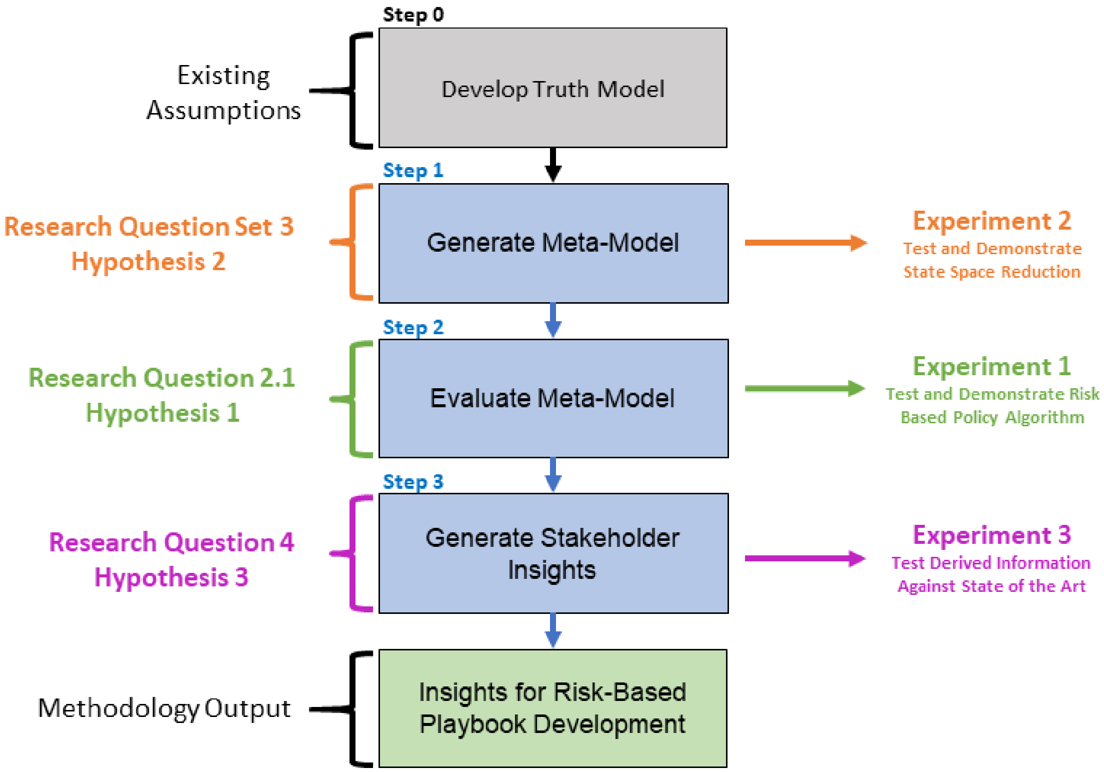


Figure 6.1: Research Questions, Methodology, Hypothesis, and Experiments

Table 6.1: Experiment Description Table

| **Exp.Set** | **Name** | **Description** | **Section** |
|---|---|---|---|
| 1a | Explicit MDPs | Test the risk-based policy algorithm against explicitly defined MDPs. | Section 6.1.1 |
| 1b | Sequential Decision Making | Test the risk-based policy algorithm against increasingly complex Truth Model cases. | Section 6.1.2 |
| 2a | Repeated Pareto Efficient Actions | Test the impact of state-space compression on risk-based policies using a Truth Model set-up resulting in simple decision spaces. | Section 6.2 |
| 2b | Acquire vs. Develop Scenario | Test the impact of state-space compression on risk-based policies using Truth Model set-up with more complex asset creation decisions space. | Section 6.2 |
| 2c | Multi-Mission Acquire vs. Develop Scenario | Test the impact of state-space compression on risk-based policies using a Truth Model set-up resulting in asset creation and allocation decision spaces. | Section 6.2 |
| 3a | Lower Complexity Problems | Compare the derived information of increasing complex test cases used in Experiment Set 1b and Experiment Set 2 to that of an optimal policy solution. | Section 6.3.1 |
| 3b | Full Complexity Problem | Demonstrate the methodology application to a realistic scenario and compare the derived information to an optimal policy solution. | Section 6.3.2 |

## 6.1 Experiment Set 1: Risk-Based Policy Development

The first set of experiments (Experiment Set 1) is designed to evaluate Hypothesis 1. Hypothesis 1 theorizes a relationship between the variation of the risk-tolerance ($\xi$) and the policies generated using the risk-based policy algorithm used in Step 2 of the methodology. There are two specific measure tied to the variation in risk-tolerance. First, the resulting mean and variance of stakeholder return will produce a return with a relative

- higher mean and higher variance for a high risk-tolerance (e.g. $\xi = 1$)
- lower mean and lowest variance for a lowest risk-tolerance (e.g. $\xi = 0$)
- lowest mean and high variance for a worst risk-tolerance (e.g. $\xi = -1$)

Second, through varying the risk-tolerance of a stakeholder, the Pareto optimal (and non-optimal) Action frontier can be determined.

The independent variable in Experiment Set 1 is the risk tolerance level of the stakeholder of interest. A different explicit MDP or basic Truth Model set up are used for each scenario. The measurements for a given scenario are the resulting policies produced for each risk tolerance level. The policies are used to evaluate their use in the full MDP or Truth Model to demonstrate the relative mean and variance. The policies are also used to identify the Pareto optimal and non-optimal actions for a given state. The first independent variable is the Return vs. Time resulting from policy implementation. The second independent variable is the policy risk-tolerance sensitivity profile.

Two steps of complexity are used to demonstrate the dependency of the Pareto efficient actions on the risk-tolerance level. Experiment Set 1a explicitly defines MDPs with varying actions and reward profiles allowing direct comparison of the defined action-rewards and the resulting policies. Experiment Set 1b uses a Truth Model with varying degrees of scenario complexity to investigate the Pareto efficiency of actions and the risk-tolerance level.

Table 6.2: Experiment Set 1 Overview

| | |
|---|---|
| **Independent Variable** | Risk-Tolerance Level ($\xi$) |
| **Dependent Variable** | Initial-State Risk-Tolerance Policy Sensitivities |
| **Case Variables** | MDP Source (Explicit or Truth Model Derived) |
| **Scenario Variables** | MDP and Scenario Complexity |

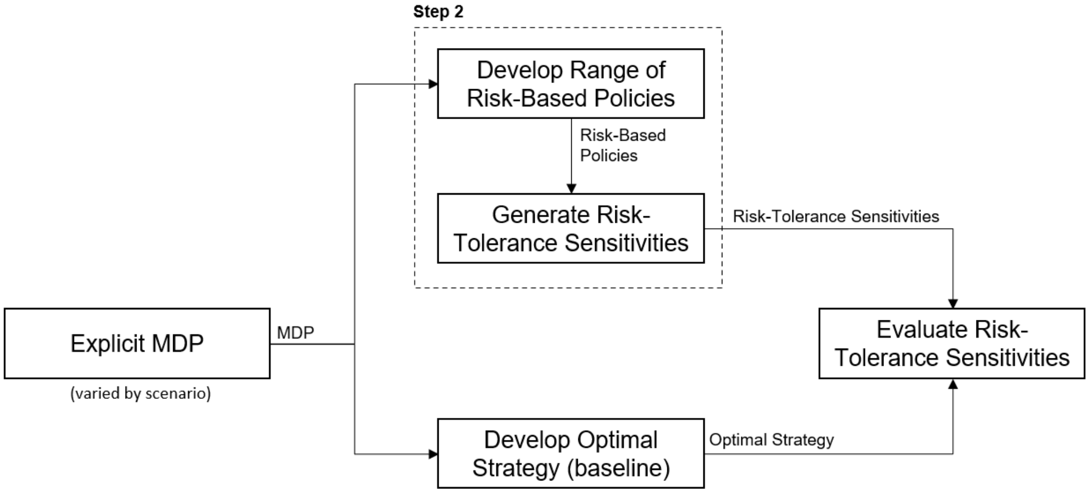

Step 2
Develop Range of Risk-Based Policies
Risk-Based Policies
Generate Risk-Tolerance Sensitivities
Risk-Tolerance Sensitivities
Explicit MDP
(varied by scenario)
MDP
Evaluate Risk-Tolerance Sensitivities
Develop Optimal Strategy (baseline)
Optimal Strategy

Figure 6.2: Experiment Set 1a Setup

### 6.1.1 Experiment Set 1a: Solving Explicit MDPs

Experiment Set 1a is defined by the use of explicit MDPs to evaluate the policy generation algorithm used in Step 2 of the methodology. The design of the explicit MDP is varied across the Scenarios while other Experiment Set 1 variables are held constant (Table 6.3). The experimental setup is outlined in Figure 6.30. Each explicit MDP is solved using traditional Q-learning via TD-$\lambda$ to generate an optimal solution and the risk-based policies used in Step 2 of the methodology. The risk-based policies are compared against the expected outcome and the optimal policies. The comparison between the calculated and expected outcome demonstrates the algorithm produces appropriate risk sensitive policies and identifies Pareto efficient (and inefficient) actions. The comparison against an optimal solution demonstrates the information lost when uncertainty is not taken into account.

Table 6.3: Experiment Set 1a Overview

| | |
|---|---|
| **Independent Variable** | Risk-Tolerance Level ($\xi$) |
| **Dependent Variable** | Initial-State Risk-Tolerance Policy Sensitivities |
| **Case Variables** | Explicit MDP |
| **Scenario Variables** | MDP Complexity |

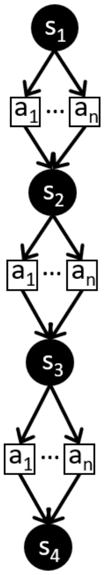


Figure 6.3: Experiment Set 1a, Case 1 MDP Set Up

*Experiment Set 1a, Case 1: Simple Staged MDPs*

Experiment 1a, Case 1 is characterized by the use of a constant action set with specified action rewards. The state space is independent of action selection and therefore solely time dependent. Figure 6.3 illustrates the action collapsed structure of the MDP (three steps, four states).

The constant action set results in sequential decision making at each time step. The number of actions and the associated reward are varied with each Scenario. Scenario 1 begins with a simple two-action equal-variance setup with more complexity added for each additional scenario. Scenario 9 culminates in a full set of Pareto efficient and inefficient sequential actions.

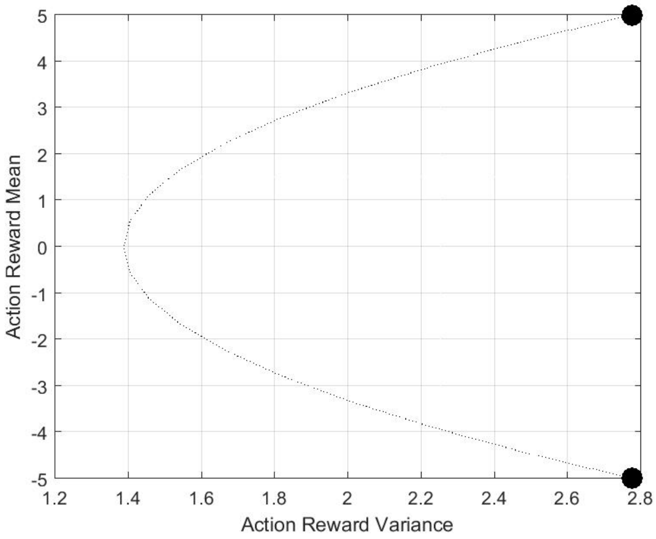


Figure 6.4: Two Actions with Equal Variance Action-Reward Profile

**Scenario 1: Two Actions** The simplest MDP scenario is a sequential game with two actions (Figure 6.5). The action reward is set to an equal reward variance and symmetric reward mean ($\mu_r(a_1) = -\mu_r(a_1)$). The selected action rewards and the Pareto frontier are shown in Figure 6.4.

**Scenario 2: Three Actions, Equal Variance** Scenario 2 adds a third action available to stakeholder at each state (Figure 6.6). The new action maintains an equivalent reward variance with a mean of zero (Figure 6.7).

**Scenario 3: Three Actions, Equal Mean** Scenario 3 is again defined by four states and three actions (Figure 6.6) but with a modified action-reward profile. The action-reward profile is modified to have a constant mean and varied variance across the three available actions (Figure 6.8).

**Scenario 4: Three Actions, Linear** Scenario 4 is again defined by four states and three actions (Figure 6.6) but with a modified action-reward profile. The action-reward profile is modified to have a linear change in variance with respect to mean across the three available

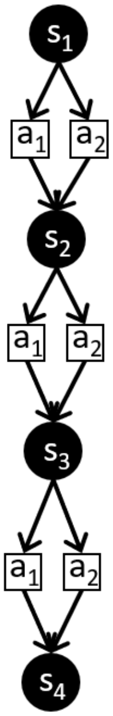


Figure 6.5: Four States, Two Action MDP

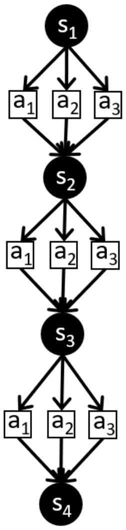


Figure 6.6: Four States, Three Action MDP

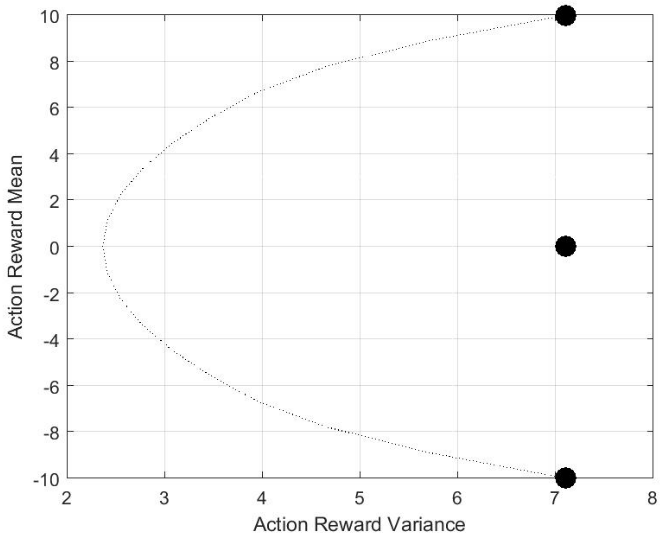


Figure 6.7: Three Actions with Equal Variance Action-Reward Profile

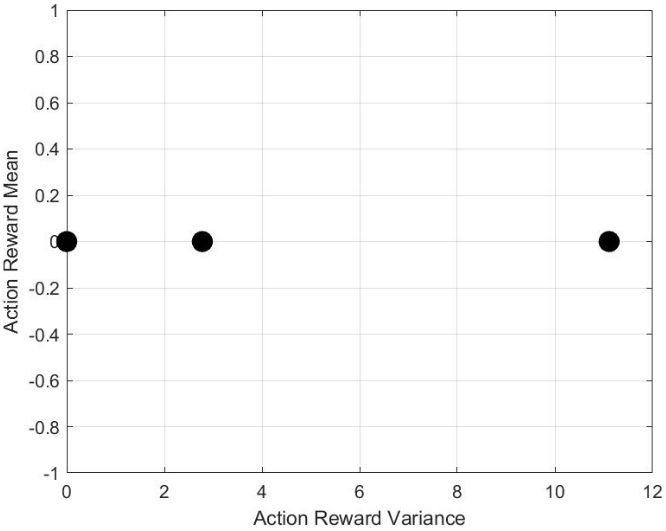


Figure 6.8: Three Actions with Equal Mean Action-Reward Profile

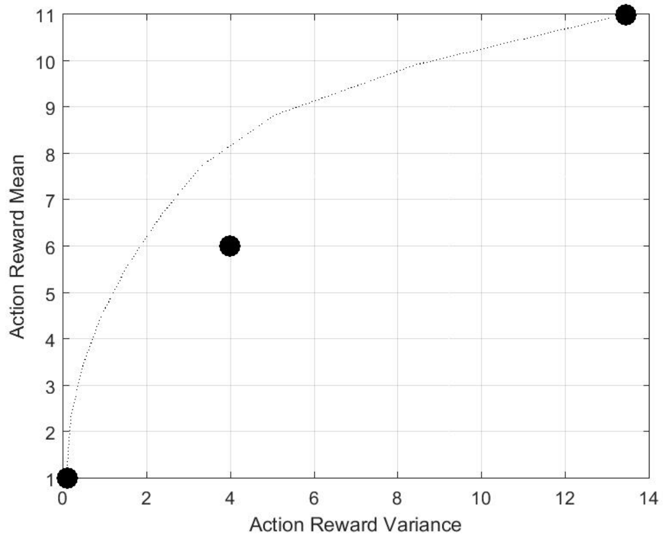


Figure 6.9: Three Actions with Linear Mean-Variance Relationship Action-Reward Profile

actions (Figure 6.9).

**Scenario 5: Four Actions, One Mild Pareto Inefficient** Scenario 3 expands to four states and four actions (Figure 6.3 with $n = 4$). The action-reward profile builds on Scenario 2 and is depicted in Figure 6.10.

Three actions represent Pareto efficient options with two extreme options with equal variance ($\sigma^2(a_1) = \sigma^2(a_3)$) and symmetric mean rewards ($\mu_r(a_1) = -\mu_r(a_3)$). A near minimal variance action is characterized by $\mu_r(a_3) = 0$ and a $\sigma^2(a_2) < \sigma^2(a_1) = \sigma^2(a_3)$. A fourth action-reward is defined such that it will be Pareto inefficient. This is done by setting $\mu_r(a_4) = 0$ ($\mu_r(a_4) < |\mu_r(a_1)| = |\mu_r(a_3)|$) and $\sigma^2(a_1) = \sigma^2(a_3) < \sigma^2(a_4)$.

**Scenario 6: Four Actions, One Significant Pareto Inefficient** Scenario 6 builds on Scenario 5 by increasing the extent of the Pareto inefficiency of $a_4$ (Figure 6.11). This is done by increasing the variance in the action-reward of $a_4$ such that $\sigma^2(a_1) = \sigma^2(a_3) << \sigma^2(a_4)$.

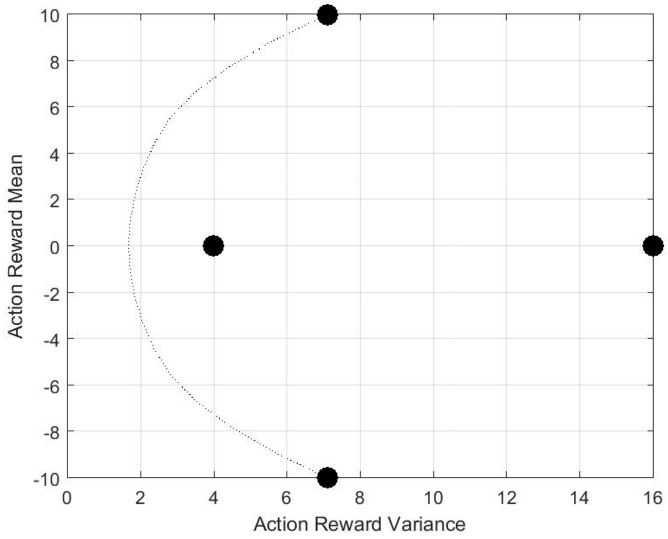


Figure 6.10: Three Pareto Actions with One Mild Pareto Inefficient Action Action-Reward Profile

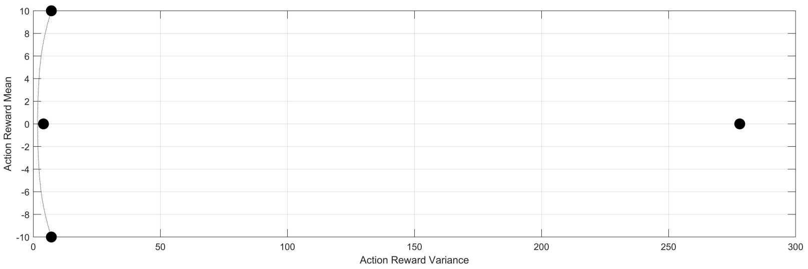


Figure 6.11: Three Pareto Actions with One Significant Pareto Inefficient Action Action-Reward Profile

Table 6.4: Seven-Action Pareto Efficient Action-Reward Profile

| Action | $\mu_r$ | $\sigma_r^2$ |
|---|---|---|
| $a_1$ | -10 | 3.33 |
| $a_2$ | -8 | 2.13 |
| $a_3$ | -2 | 0.13 |
| $a_4$ | 0 | 0 |
| $a_5$ | 2 | 0.13 |
| $a_6$ | 8 | 2.13 |
| $a_7$ | 10 | 3.33 |

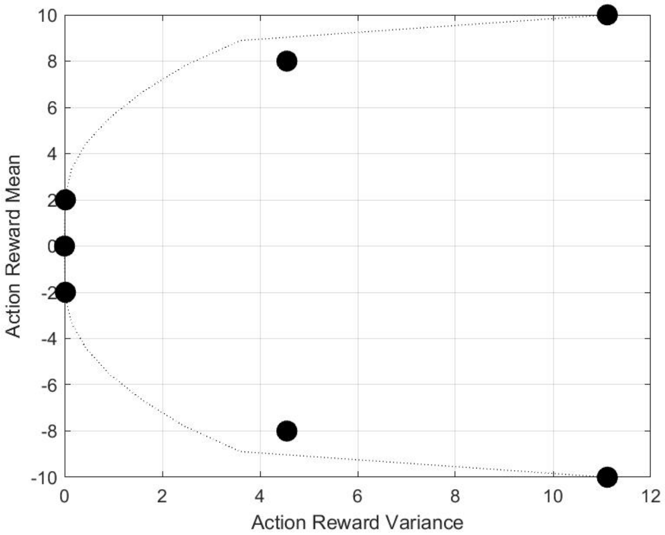


Figure 6.12: Seven-Action Pareto Efficient Action-Reward Profile

**Scenario 7: Explicit Pareto Frontier Action Space** Scenario 7 further extends the characterization of a Pareto frontier beyond the more simple setup used in Scenario 5. The action-reward profile is expanded to a four-state seven-action MDP depicted in Figure 6.12. The explicit values for the action-reward profile are documented in Table 6.4.

**Scenario 8: Explicit Pareto Frontier Action Space, Mild Pareto Inefficient Actions** Scenario 8 builds on Scenario 7 similar to how Scenario 5 builds on Scenario 2. Eight additional actions were added to the MDP making $n = 15$ as defined in Figure 6.3. The additional actions $\mu_r$ and $\sigma_r^2$ were selected to be Pareto inefficient with a bounding rule of $-2 < mu_r(a_{inefficient}) < 2$ and $4 < \sigma_r^2(a_{inefficient}) < 8$. Note the Pareto inefficient

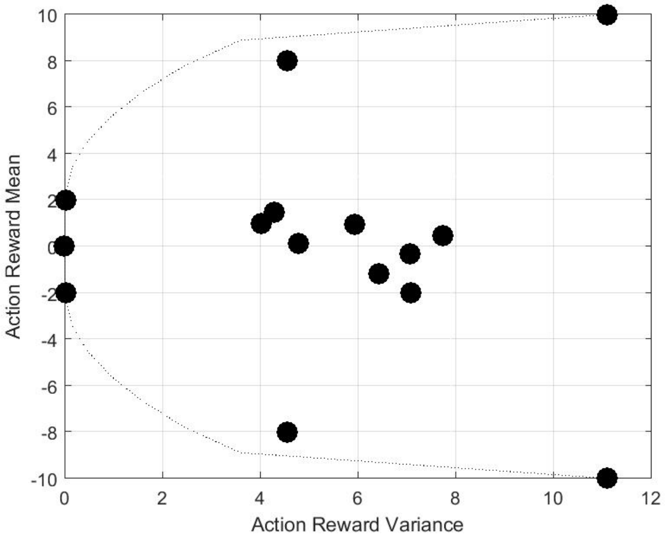


Figure 6.13: Seven-Action Pareto Efficient with Eight Mild Pareto Inefficient Actions Action-Reward Profile

action-rewards are bounded within the Pareto efficient action-rewards (Figure 6.13).

**Scenario 9: Explicit Pareto Frontier, Significant Pareto Inefficient Actions** Scenario 9 begins with the setup used for Scenario 8 and increases the reward variance for the Pareto inefficient actions (Figure 6.14). The shift to $30 < \sigma_r^2(a_{inefficient}) < 60$ further increases the Pareto inefficiency of the actions. The Pareto optimal action-rewards remain the same.

*Experiment Set 1a, Case 2: Short and Long Term Stakeholder Preferences*

Moving from Case 1 to Case 2 for Experiment Set 1a add complexity to the action space. In Case 1, the full MDP was characterized by a single state at each time step. In Case 2, each action selected results in both a unique reward and a resulting next state. Additionally, each state-action reward profile is uniquely selected to test information extracted via risk-tolerance sensitivity analysis. Each selected MDP is unique to a given scenario. The general structure for each MDP is outlined in Figure 6.15. Scenario 1 demonstrates expected outcomes using the most simplistic state-action set up. Scenario 3 increases the complexity of the MDP and introduced the concept of short versus long term preference.

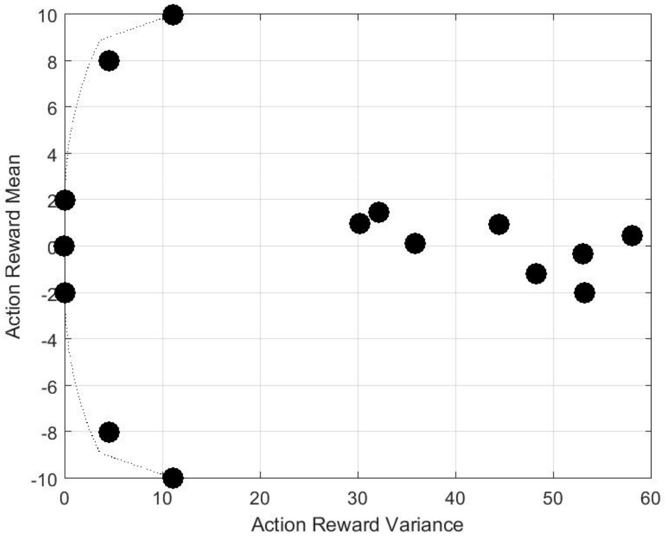


Figure 6.14: Seven-Action Pareto Efficient with Eight Significant Pareto Inefficient Actions Action-Reward Profile

Scenario 3 and 4 test the concept of Pareto efficient action identification similar to Case 1 Scenarios 5 through 9.

**Scenario 1: Baseline Multi-Action Multi-State** Scenario 1 begins with the simplest of scenarios and examines using the algorithm over multi-stage (3 step) two-action MDP (Figure 6.16). For each state, there are the same two actions available with the same return mean and variance. The mean-variance is similar to that used in Case 1 Scenario 1 for the repeat cases. Each action yields a unique state and is not shown for simplicity. The blue represents a state reached via the higher mean reward action and the green represents a state reached via the lower mean reward.

**Scenario 2: Short Term Reward Versus Long Term Return** The second scenario adds another action to each state and modifies the relative action mean-variance to enable short term versus long term trades. The action space at each state represents a simple Pareto frontier, similar to Case 1 Scenario 2. The relative mean and variance rewards for state-actions along with the MDP structure are depicted in Figure 6.17. Each state has a max-

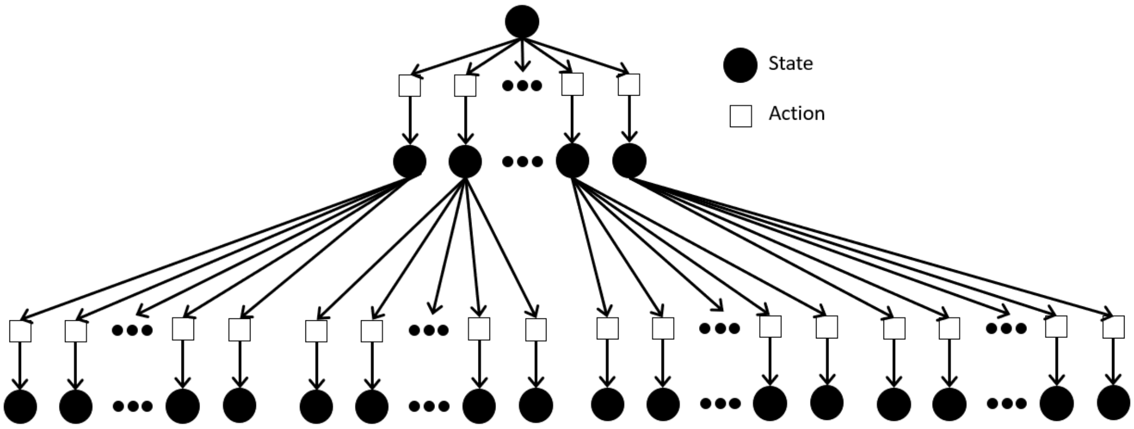


Figure 6.15: Experiment Set 1a, Case 2 MDP Set Up

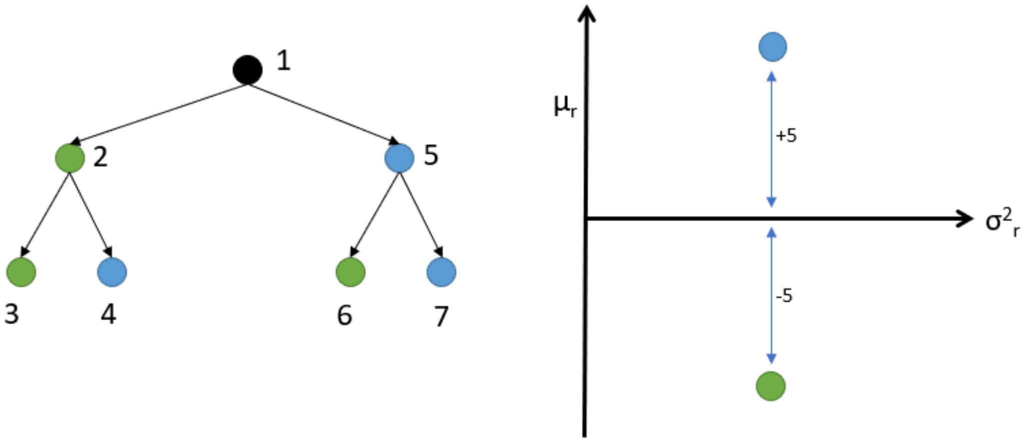


Figure 6.16: Experiment Set 1a, Case 2, Scenario 1 Setup Description

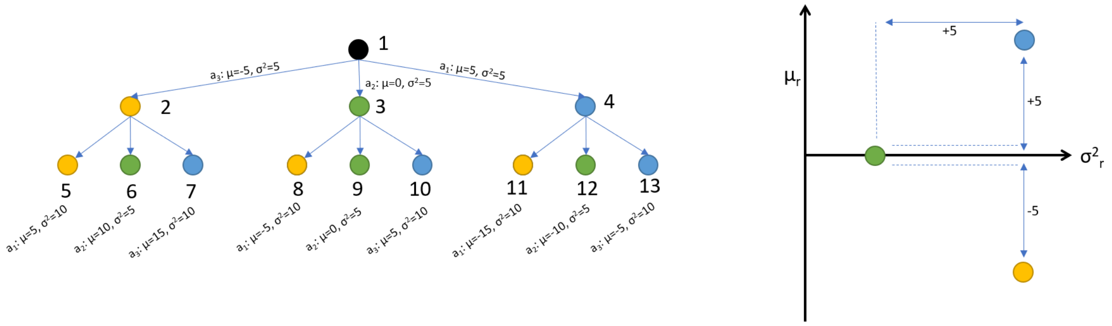


Figure 6.17: Experiment Set 1a, Case 2, Scenario 2 Setup Description

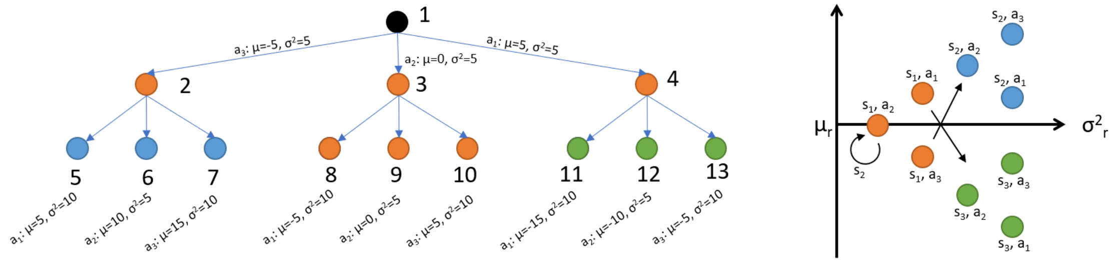


Figure 6.18: Experiment Set 1a, Case 2, Scenario 2 State-Action Reward Clusters

risk, min-risk, and worst action. The relative mean and variance of each state's actions set is varied as shown in Figure 6.18. The second step action-set varies across states 2, 3, and 4 relative to the first time-step action-set seen in state 1. State 1 to State 2 is the lowest of the first state action rewards but the subsequent action-set yields the highest reward relative to all other action-sets across all states. Similarly, State 1 to State 4 is the highest reward initial action with the State 4 actions-set the lowest return of all actions sets. It is anticipated, if the time horizon is long enough, that the preferred action in the first state will be contributed to by the actions available in the second step states. In this case, it is anticipated that despite a lower initial reward, Action 1 in State 1 will be preference for a higher-return higher-risk scenario. Similarly, Action 3 in State 1 will be preferred at a low risk-tolerance level.

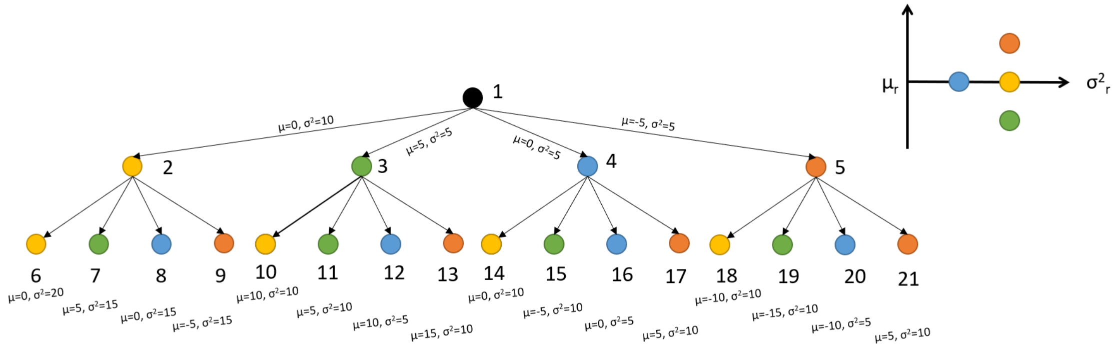


Figure 6.19: Experiment Set 1a, Case 2, Scenario 3 Setup Description

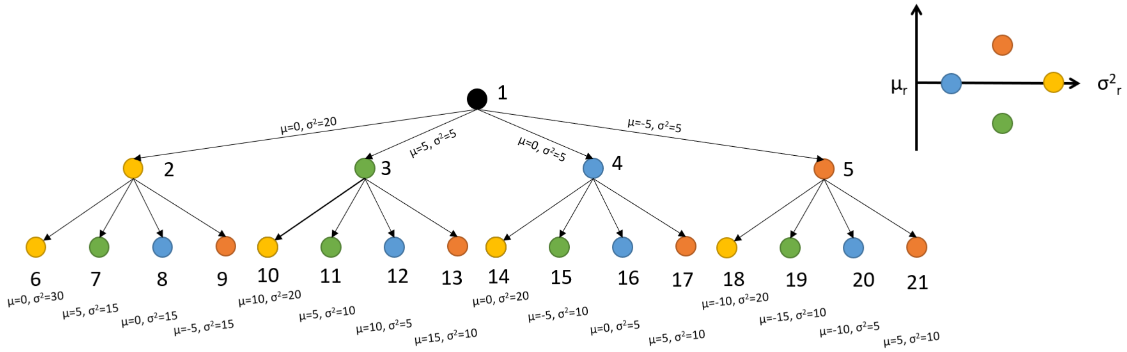


Figure 6.20: Experiment Set 1a, Case 2, Scenario 4 Setup Description

**Scenario 3: Multi-Action Multi-State Pareto Frontier Actions and Mild Pareto Inefficient Actions** Scenario 3 setup establishes a baseline for consecutive Pareto efficient actions with a single mildly inefficient action building on Case 1 Scenario 5. Each state has four actions as depicted in Figure 6.19.

**Scenario 4: Multi-Action Multi-State with Pareto Efficient and Significant Pareto Inefficient Actions** Scenario 4 builds on Scenario 3 by increasing the level of inefficiency of the Pareto inefficient action (Figure 6.20).

*Experiment Set 1a, Case 3: Fully Random MDP*

A third case for Experiment Set 1a is used to demonstrate the policy generation using a randomly developed graph. Two levels of comlexity are used. The first random graph is generated using a three action algorithm and the second is generated using a five action algorithm.

For the first, at each node, three actions max are available and can result in single or multiple new states. The tree is randomly grown with these rules in mind. The mean and variance of the reward is randomly seeded for each state-action-state tuple. The developed MDP structure is depicted in Figure 6.21 with only state-to-state transitions. The MDP action space is depicted in Figure 6.22 displaying the specific state-action-state transitions. The width of the line is the resulting probability of transition given a specific state-action selection (independent of selected policy).

Five actions max are available and can result in single or multiple new states for the second random graph. Similar to the first scenario, the five action graph is grown using the same rules and random state-action-state Reward means and variances. The developed MDP structure is depicted in Figure 6.23a with only state-to-state transitions. The MDP action space is depicted in Figure 6.23b displaying the specific state-action-state transitions.

### 6.1.2 Experiment Set 1b: Sequential Decision Making

Experiment Set 1b adds the use of the Truth Model with prescriptive scenarios to test the policy algorithm (Figure 6.24 and Table 6.5). The prescriptive scenario defines the setup of the Truth Model. The Truth Model is sampled via MC episodes and a full MDP is generated (note: The generation of the meta-model and state space compression is the focus of Experiment Set 2). Similar The resulting uncompressed MDPs are evaluated using the policy algorithm and state-action metrics used in Step 2 of the methodology. Additionally, optimal strategies are developed against the Truth-Model-generated uncompressed MDP. The resulting risk-tolerance sensitivities are evaluated against anticipated results. A description

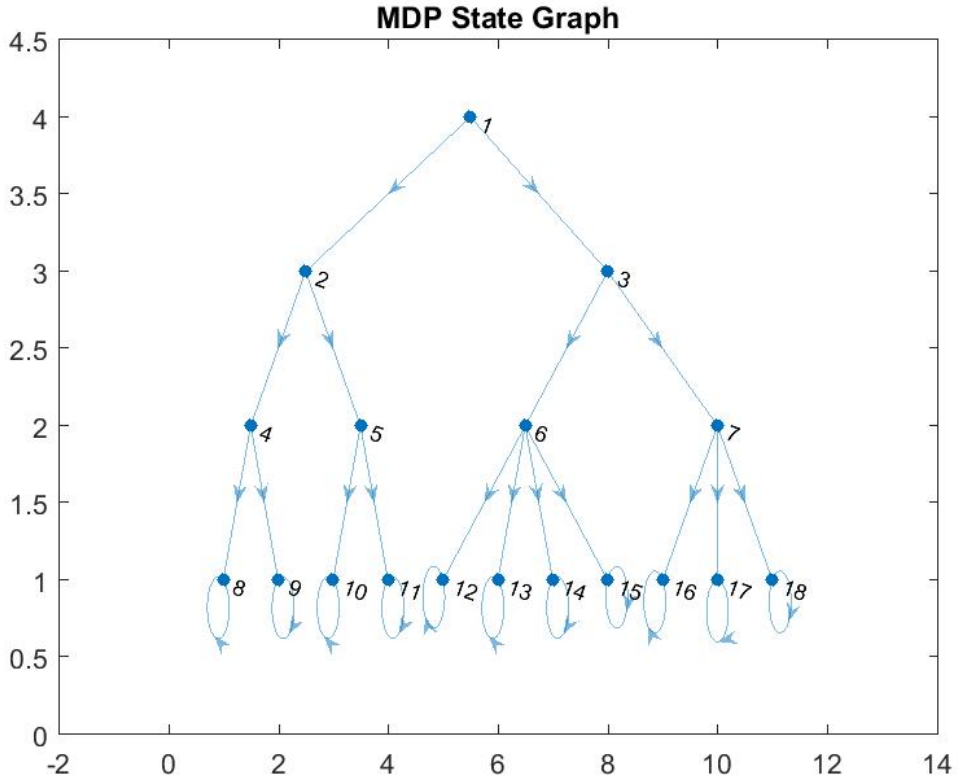


Figure 6.21: Experiment Set 1a, Case 3, Scenario 1 MDP Graph

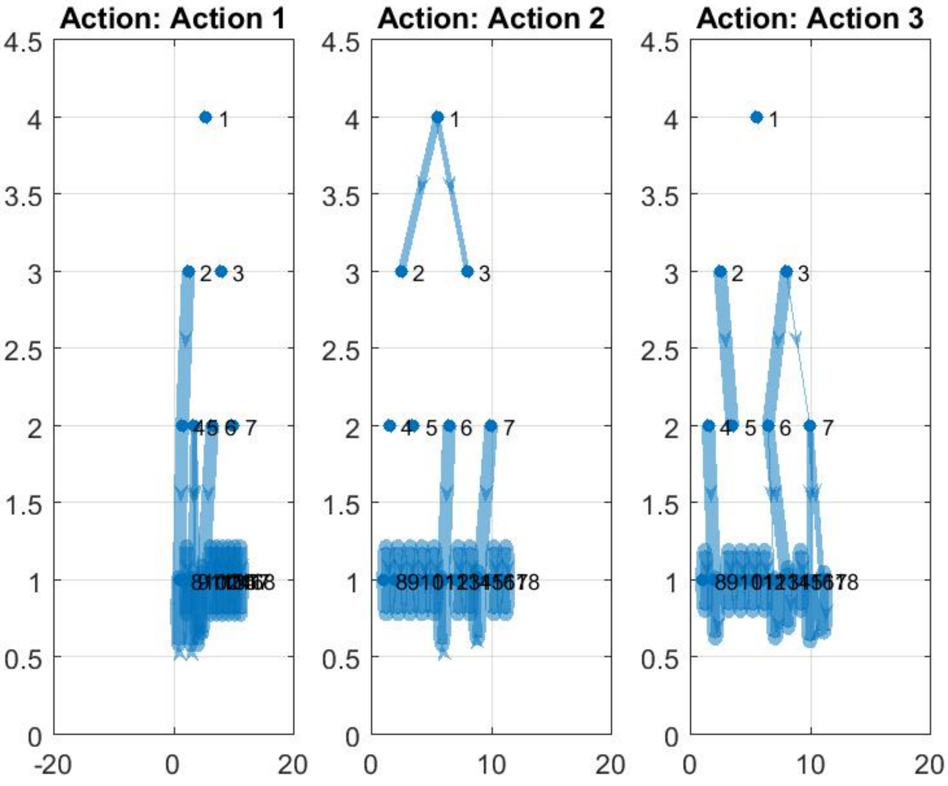
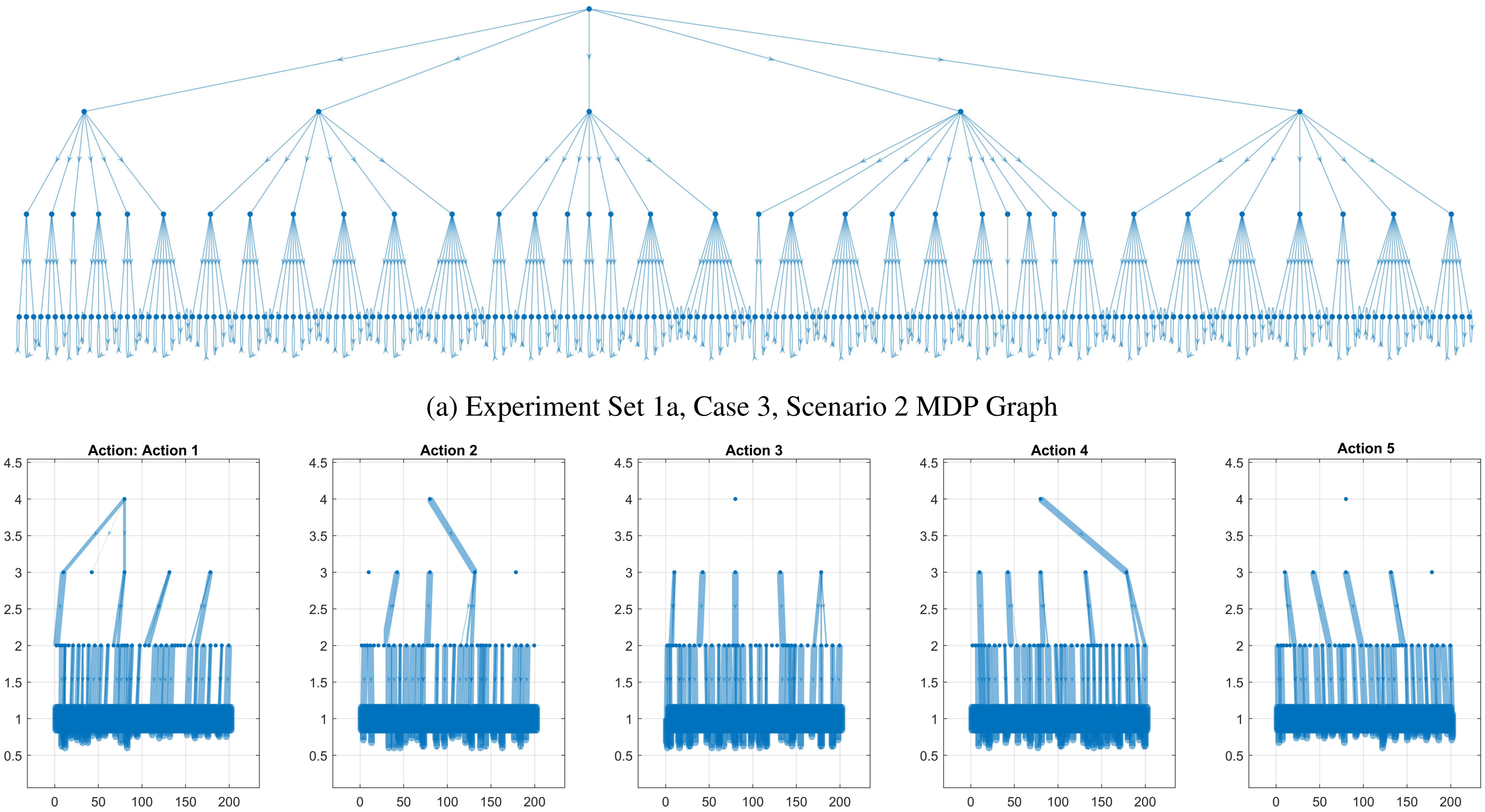


Figure 6.22: Experiment Set 1a, Case 3, Scenario 1 MDP Graph



(a) Experiment Set 1a, Case 3, Scenario 2 MDP Graph

(b) Experiment Set 1a, Case 3, Scenario 2 MDP Graph

Figure 6.23: Experiment Set 1a, Case 3, Scenario 2 Graph Structure

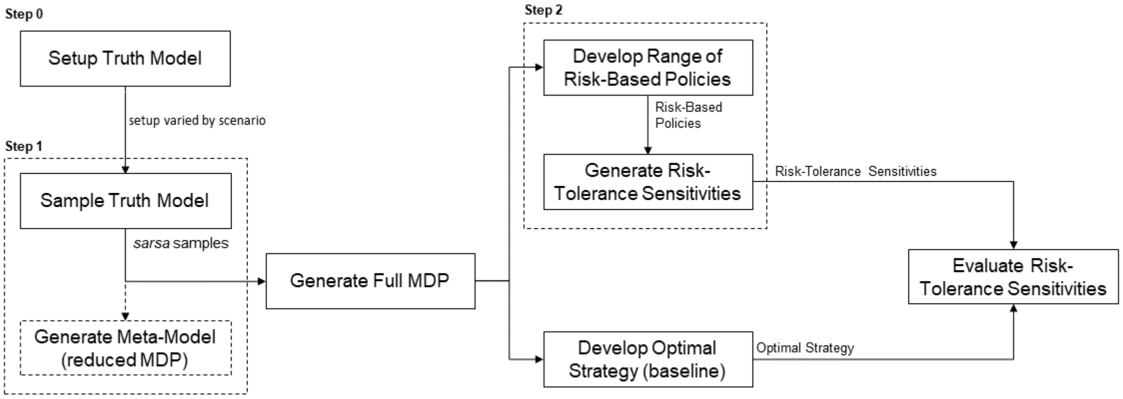

Step 0
Setup Truth Model
setup varied by scenario
Step 1
Sample Truth Model
sarsa samples
Generate Meta-Model (reduced MDP)
Generate Full MDP
Step 2
Develop Range of Risk-Based Policies
Risk-Based Policies
Generate Risk-Tolerance Sensitivities
Risk-Tolerance Sensitivities
Evaluate Risk-Tolerance Sensitivities
Develop Optimal Strategy (baseline)
Optimal Strategy

Figure 6.24: Experiment 1b Setup

Table 6.5: Experiment Set 1b Overview

| | |
|---|---|
| **Independent Variable** | Risk-Tolerance Level ($\xi$) |
| **Dependent Variable** | Risk-Tolerance Policy Sensitivities |
| **Case Variables** | Truth Model Setup |
| **Scenario Variables** | Truth Model Definition Variables (selected cases only) |

of the Truth Model can be found in Appendix B.

*Experiment 1b, Case 1: Repeated Pareto Efficient Actions*

The identification of the Pareto frontier under increasing complex scenarios is necessary to demonstrate the capability of the risk-based policy algorithm. At teach time step, the stakeholder of interest can select between a set actions to acquire a system from a defined set of systems. The resulting system is added to the available deployed systems for the next time step (acquisition time is set equal to a single time step). The difference between Scenario 1 and Scenario 2 is defined mean and variance of the system performance. Varying the presence of the inefficient acquisition options demonstrates the ability to discern the Pareto efficient actions from the Pareto inefficient actions. Both temporal and performance uncertainty are included in both scenarios.

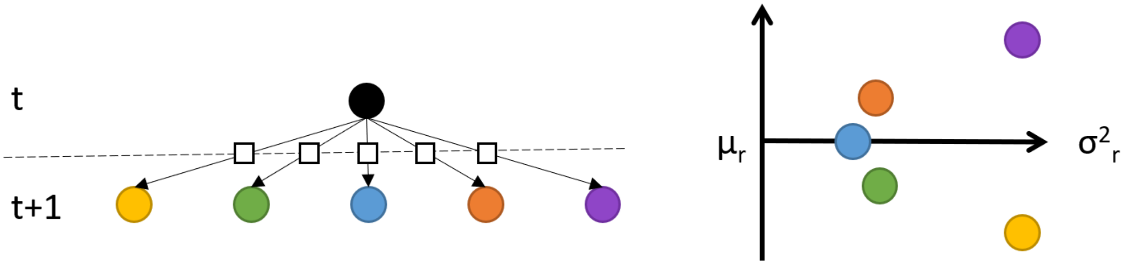


Figure 6.25: Experiment Set 1b, Case 1, Scenario 1 Setup Description

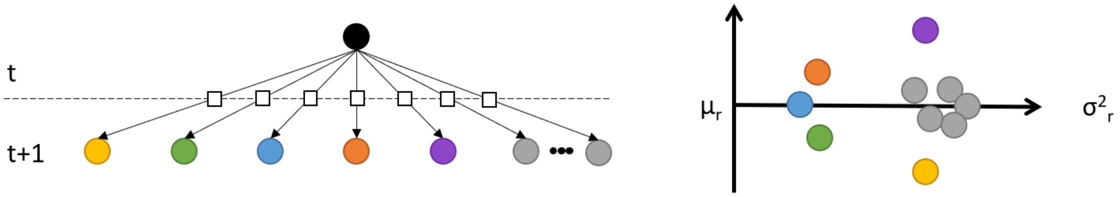


Figure 6.26: Experiment Set 1b, Case 1, Scenario 2 Setup Description

**Scenario 1: Baseline Repeated Pareto Efficient Actions** Scenario 1 represents a set of acquisition decisions that can result in a defined Pareto frontier (Figure 6.25).

**Scenario 2: Repeated Pareto Efficient Actions With Pareto Inefficient Actions** Scenario 2 adds Pareto inefficient actions to the decisions space of the stakeholder (Figure 6.26). At each state, the stakeholder has the additoinal option to select the acquisition of various Pareto inefficient systems.

*Experiment 1b, Case 2: Acquire vs. Develop Scenario*

The general Truth Model setup for Experiment 1b is outlined in Chapter 2 with the introduction of the example problem. At each step, the stakeholder-of-interest can select to develop a new system (higher mean and higher variance in performance) or to acquire a previously developed system. The development time and and performance are both subject to uncertainty. Case two also adds sequential development of systems. A predeceasing sys-

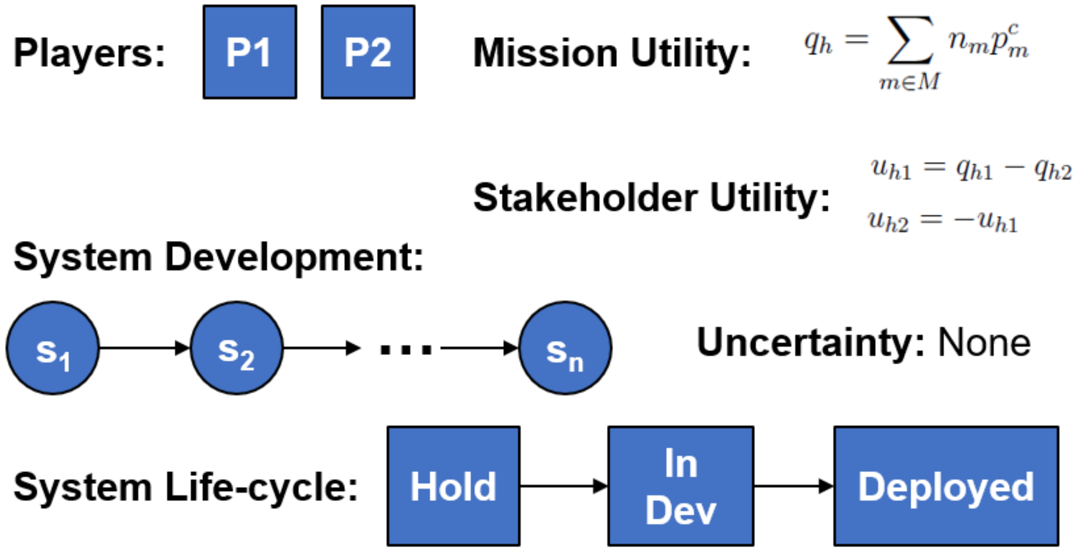


Figure 6.27: Experiment Set 1b, Case 2 Setup Description

tem must be developed before a stakeholder has the option to develop the next higher-mean higher-variance performing system. There is a single mission of interest that each system applies their capabilities, or performance, against. The stakeholder utility is a zero-sum game between the two stakeholders, or players. An overview of the Case setup is depicted in Figure 6.27.

**Scenario 1: Baseline Acquire Versus Develop** The baseline scenario uses a trend of increasing mean performance and performance variation as Stakeholder one develops new systems. Both the mean and variance in system definition are described in Table 6.9.

**Scenario 2: Long-Term Acquire Versus Develop** The variation for Case 2 is the mean performance of System 2. The mean performance of System 2 is increased significantly. This allows the impact of short time frames versus longer time frames on the policy algorithm.

Table 6.6: Experiment Set 2c: Stakeholder System Ownership

| | **Stakeholder 1** | **Stakeholder 2** | **Stakeholder 3** |
|---|---|---|---|
| System 1 | 1 | 0 | 0 |
| System 2 | 1 | 0 | 0 |
| System 3 | 1 | 0 | 0 |
| System 4 | 1 | 0 | 0 |
| System 5 | 0 | 1 | 0 |
| System 6 | 0 | 1 | 0 |
| System 7 | 0 | 1 | 0 |
| System 8 | 0 | 1 | 0 |
| System 9 | 0 | 0 | 1 |
| System 10 | 0 | 0 | 1 |
| System 11 | 0 | 0 | 1 |
| System 12 | 0 | 0 | 1 |

*Case 3: Multi-Mission Acquire Versus Develop*

Case 3 adds additional complexity to case four providing additional testing of the policy algorithm. A third stakeholder is added with Stakeholder 1 and Stakeholder 2 cooperating against Stakeholder 3. A second mission is added which allows each stakeholder to allocate current resources to specific missions as part of their decision space at at each time step. The development sequence is also modified to incorporate an acquisition, refresh, or develop trade. Stakeholder utility is now based on a weighting vector describing the importance they give each individual mission. Each of the three stakeholders has a different budget and similar choices to make. Each stakeholder is examined as a separate scenario. The detailed description is outlined below.

**Stakeholders:** Three stakeholders, two are cooperative (Stakeholder 1 and Stakeholder 2).

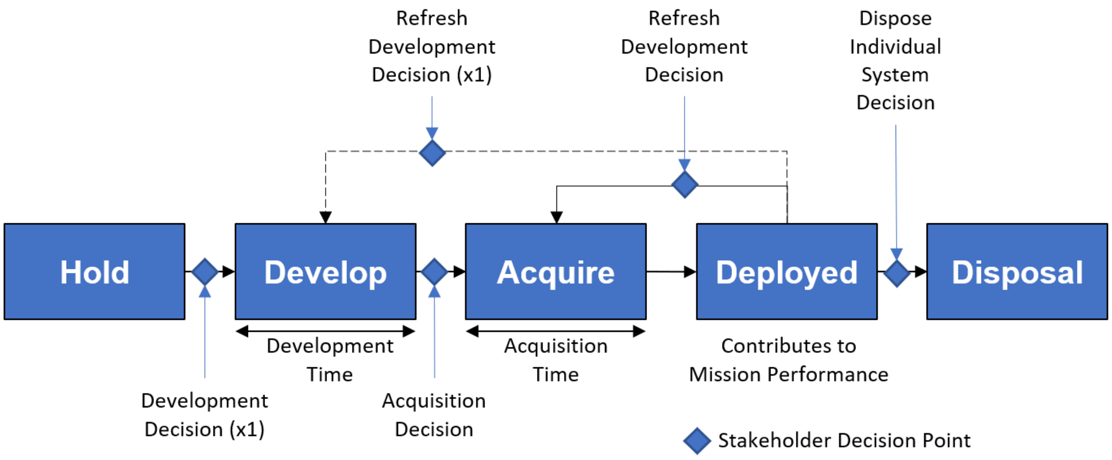

Refresh Development Decision (x1)
Refresh Development Decision
Dispose Individual System Decision
Hold
Develop
Acquire
Deployed
Disposal
Development Time
Acquisition Time
Contributes to Mission Performance
Development Decision (x1)
Acquisition Decision
Stakeholder Decision Point

Figure 6.28: Experiment Set 1b Case 3: System Life-Cycle

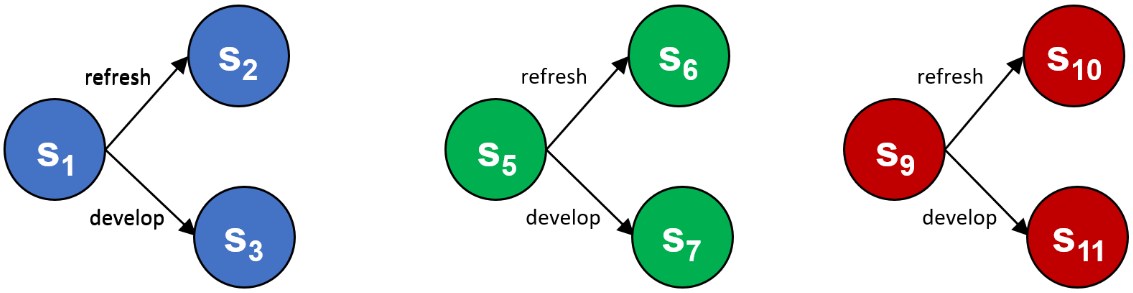


Figure 6.29: Experiment Set 1b Case 3: System Progression

**Systems:** Twelve systems with four attributed to each stakeholder (Table 6.6).

**System Life-Cycle:** The modeled system life-cycle has grown in complexity. The ability to refresh a system once it has reached the end of it's like is now and option. It is modeled as a new development with the ability to upgrade systems that would otherwise be disposed of (technology refresh). The system life-cycle is depicted in Figure 6.28 and the system progression is depicted in Figure 6.29.

**System Performance:** The system performance of a given system is defined by the a mean and variance capability which is attributed to a given mission. The system performance mean (Figure 6.8) and variance (Figure 6.7) are defined to demonstrate the result in

Table 6.7: Experiment Set 1b Case 3: System Mean Performance

| | **Mission 1** | **Mission 2** |
|---|---|---|
| System 1 | 3 | 3 |
| System 2 | 9 | 27 |
| System 3 | 27 | 9 |
| System 4 | 0 | 0 |
| System 5 | 3 | 3 |
| System 6 | 9 | 27 |
| System 7 | 27 | 9 |
| System 8 | 0 | 0 |
| System 9 | -3 | -3 |
| System 10 | -9 | -27 |
| System 11 | -27 | -9 |
| System 12 | 0 | 0 |

decisions resulting from mission preference (defined below).

**System Definition:** The system timelines are defined in Table 6.9. The asymmetry in the development and acquisition time allows the impact of temporal variations on the risk-sensitive profiles to be realized. The uncertainty associated with timelines was set to zero to allow the sensitivity to timeline to be clearly measured.

**Stakeholder Decisions (Actions):** Two classes of actions are now present in the defined scenario: asset creation and asset allocation. The asset creation encompasses the development, refresh, and acquisition of systems. The new component to Experiment Set 2c is the addition of allocating systems to specific missions. Each stakeholder also has a vary budget (Table 6.10). The budget is expressed in terms of the number of asset creation actions that can be taken. Each acquisition, refresh, or development in progress takes one action.

Table 6.8: Experiment Set 1b Case 3: System Performance Variance

| | **Mission 1** | **Mission 2** |
|---|---|---|
| System 1 | 0.6 | 0.6 |
| System 2 | 1.8 | 5.4 |
| System 3 | 5.4 | 1.8 |
| System 4 | 0 | 0 |
| System 5 | 0.6 | 0.6 |
| System 6 | 1.8 | 5.4 |
| System 7 | 5.4 | 1.8 |
| System 8 | 0 | 0 |
| System 9 | 0.6 | 0.6 |
| System 10 | 1.8 | 5.4 |
| System 11 | 5.4 | 1.8 |
| System 12 | 0 | 0 |

**Stakeholder Utility:** Stakeholder utility is no longer a direct result of the systems deployed. Each stakeholder has the decision to allocate available assets to different missions. Two missions are defined for Case 3 as described above in system performance. Individual mission level outcome (positive or negative) is determined as previously described in Chapters 2 and Chapter 6. Each stakeholder has a preference vector that defines the contribution of each mission metric to their stakeholder utility. The matrix for this case is defined in Table 6.11.

## 6.2 Experiment Set 2: State Space Compression

The second set of experiments (Experiment Set 2) is designed to evaluate Hypothesis 2. Hypothesis 2 asserts that the policies generated from compressed MDPs (meta-models) will remain usable as the compression ratio is increased. The compression of the state space and action space is used in Step 1 of the methodology to ensue tractability of the MDP to be evaluated (Figure 6.1).

Table 6.9: Experiment Set 1b Case 3: System Timeline Definition with Uncertainty

| | **Mean Development Time (time steps)** | **Development Time Uncertainty (time steps, $3\sigma$)** | **Mean Acquisition Time (time steps)** | **Acquisition Time Uncertainty (time steps, $3\sigma$)** |
|---|---|---|---|---|
| **System 1** | 4 | 0 | 2 | 0 |
| **System 2** | 2 | 0 | 1 | 0 |
| **System 3** | 4 | 0 | 2 | 0 |
| **System 4** | - | - | - | - |
| **System 5** | 4 | 0 | 2 | 0 |
| **System 6** | 2 | 0 | 1 | 0 |
| **System 7** | 4 | 0 | 2 | 0 |
| **System 8** | - | - | - | - |
| **System 9** | 4 | 0 | 2 | 0 |
| **System 10** | 2 | 0 | 1 | 0 |
| **System 11** | 4 | 0 | 2 | 0 |
| **System 12** | - | - | - | - |

Table 6.10: Experiment Set 1b Case 3: Stakeholder Budgets

| Time Step | Stakeholder 1 | Stakeholder 2 | Stakeholder 3 |
|---|---|---|---|
| 1 | 1 | 2 | 4 |
| 2 | 1 | 2 | 4 |
| 3 | 1 | 2 | 4 |
| 4 | 1 | 2 | 4 |
| 5 | 1 | 2 | 4 |

Table 6.11: Experiment Set 1b Case 3: Stakeholder Mission Preference

| Stakeholder | Mission 1 | Mission 2 |
|---|---|---|
| Stakeholder 1 | 1 | 0.25 |
| Stakeholder 2 | 0.25 | 1 |
| Stakeholder 3 | -1 | -1 |

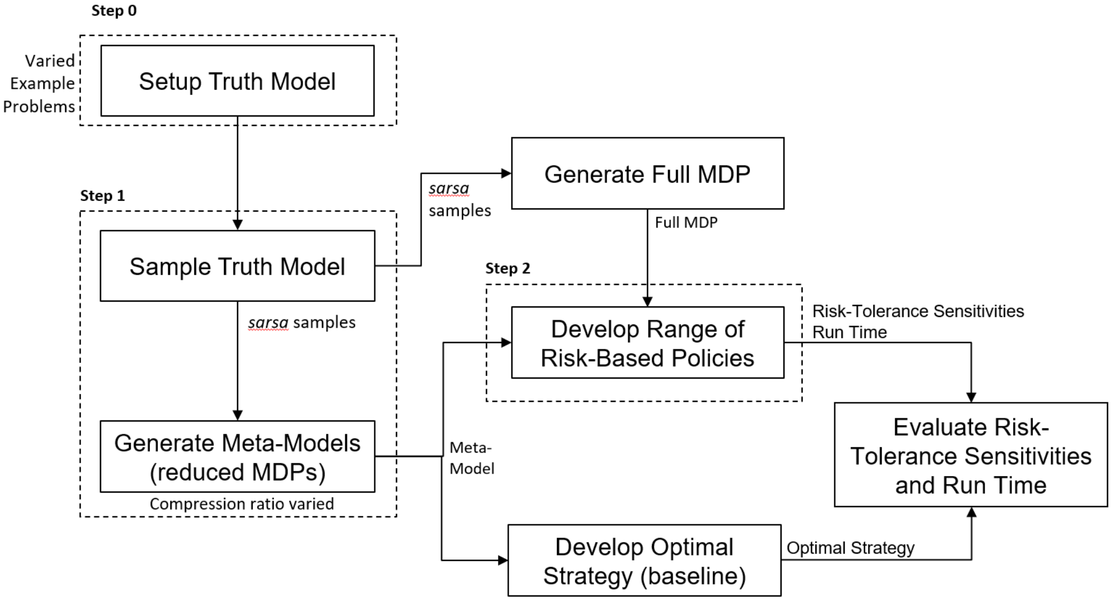


Figure 6.30: Experiment 2 Setup

Table 6.12: Experiment Set 2 Overview

| | |
|---|---|
| **Independent Variables** | Risk-Tolerance Level ($\xi$)<br>State Compression Ratio |
| **Dependent Variables** | Risk-Tolerance Policy Sensitivities<br>Policy Generation Computation Time |
| **Case Variable** | Truth Model Setup |
| **Scenario Variables** | Truth Model Definition Variables (selected Cases only) |

The independent variable in Experiment Set 1 is the compression ratio of the state and action space relative to the full MDP (Table 6.30). The scenarios defined by specific Truth Model inputs used in Experiment 1b are used in Experiment Set 2. There are two measurements taken. The first is the risk-tolerance sensitivity profiles generated by state for each compression ratio. The second is the mean, across risk-sensitivity inputs, policy computation time. The resulting risk-tolerant policy sensitivities are compared for key states across compression ratios to measure consistency and where consistency breaks down. The relative computation time allows the measurement of increased tractability as the compression ratio is decreased. The experimental setup is outlined in Figure 6.12.

Three steps of complexity are used to evaluate the consistency of policy generation and decreased computation time. Each step corresponds to increased complexity described in Experiment 1b. Case 1 uses the repeat acquisition-only decision of Pareto efficient system capabilities. Case 2 adds the complexity of system development and non-ideal system performance profiles. Case 3 adds varied resource constraints, asset allocation to multiple objectives, and additional stakeholders.

## 6.3 Experiment Set 3: Generating Insights from Derived Information

Experiment Set 3 tests the generation of informative insights based on the data generated in methodology Step 2. The provided information is analyzed and insights into the decision making process are generated. The generated insights are compared to those generated using an optimal policy to benchmark the methodology against current best techniques. A

Table 6.13: Experiment Set 3 Overview

| | |
|---|---|
| **Independent Variable** | Scenario Complexity (across Experiment 3a and Experiment 3b) |
| **Dependent Variables** | State-Based Rule Sets<br>Action-Based Rule Sets<br>Additional Insights |
| **Case Variables** | Experiment 3a: Truth Model Setup<br>Experiment 3b: n/a |

summary overview of the Experiment 3 set-up can be found in Table 6.13.

### 6.3.1 Experiment Set 3a: Lower Complexity Problems

Experiment Set 3a builds upon Experiment 1b and Experiment 2 cases. The evaluation of the meta-model conducted in Experiment Set 1b are used to generate insights and are compared against the insight gained from optimal policies. The following three scenarios are addressed:

- Acquire Only Case (Extension of Experiment 1b Case 1)
- Acquire and Develop Case (Extension of Experiment 1b Case 2)
- Acquire, Develop, and Allocate Case (Extension of Experiment 1b Case 3)

### 6.3.2 Experiment Set 3b: Full Complexity Problem

Experiment Set 3b is the culmination of the complexity increases applied in Experiment Set 1b and Experiment Set 2. Experiment Set 3b fully exercises the developed methodology using a full complexity Truth Model setup (Figure 6.31). The experiment is the primary test for Hypothesis 3. To test the hypothesis, both the risk-based methodology presented in Chapter 5 and a traditional optimal strategy method are used to evaluate the full complexity problem. The insights garnered from both methods are then evaluated against each other.

The full complexity problem is described by the Truth Model set-up used to exercise the methodology. The description of the Truth Model set-up depicts the multi-stakeholder,

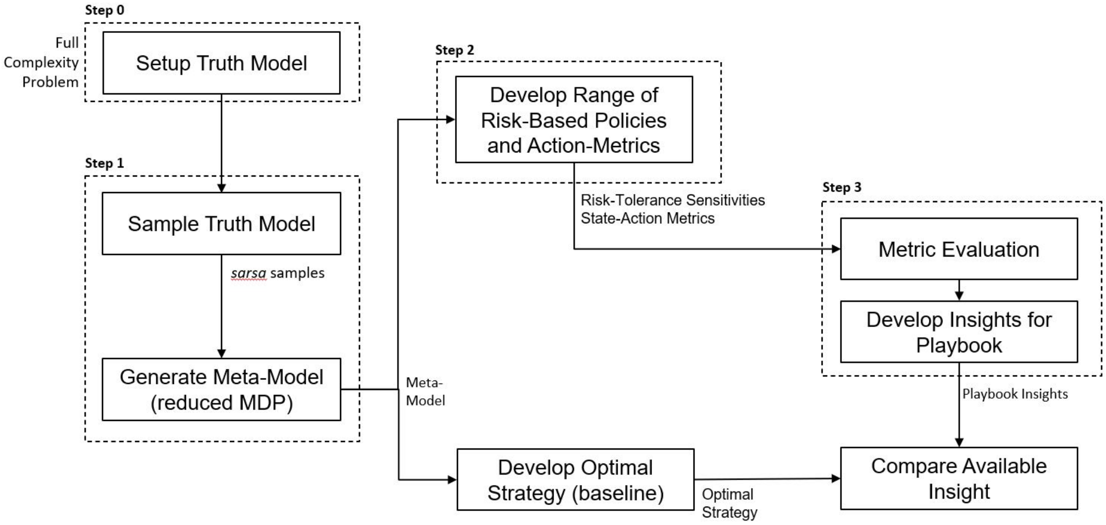

Step 0
Full Complexity Problem
Setup Truth Model
Step 1
Sample Truth Model
sarsa samples
Generate Meta-Model (reduced MDP)
Meta-Model
Step 2
Develop Range of Risk-Based Policies and Action-Metrics
Risk-Tolerance Sensitivities
State-Action Metrics
Step 3
Metric Evaluation
Develop Insights for Playbook
Playbook Insights
Develop Optimal Strategy (baseline)
Optimal Strategy
Compare Available Insight

Figure 6.31: Experiment 3 Setup

multi-objective, and uncertainty aspects used in constructing the model. The problem includes five stakeholders (3 blue and 2 red) and three mission level objectives. Both performance and temporal uncertainty are used in the definition of system creation and system performance.

*Experimentation Equivalency*

Specific requirements for the methodology were outlined in Section based on the unique aspects of the problem addressed. These aspects are also requirements on the experimental test bed which is the Truth Model for the full complexity test case, Experiment 3b. Section 5.2 describes an example Truth Model set-up and the required outputs. A full set-up entails, at a minimum, the development of multiple mission level evaluation simulations, stakeholder decision space generation, technology life-cycle modeling, system life-cycle modeling, and an architecture selection mechanism. Experiment 1 and 2 operated on less than full complexity test cases. Experiment 3b looks to fully address the complete problem. The requirements specified above apply to the Truth Model test bed used in Experiment 3b.

The Truth Model used as an input for experiments is described in Appendix B. The specific Truth Model set-up used to fully exercise and benchmark the methodology is described later in this section. The combination of the Truth Model and the Truth Model set-up must provide the necessary multi-stakeholder, multi-objective, and uncertain environment with which to test the methodology. The model and set-up fully addresses the prescribed requirements outlined in Section :

**Multi-Stakeholder Decision Making:** Variable stakeholders can be defined, each with their own independent decision cycle at each time step. Each stakeholder is treated equally as an independent entity. The stakeholder state, actions, and utilities are independently tracked and recorded. The impacts of individual stakeholders impact the overall mission level metrics that feed into individual stakeholder utility metrics.

**Evolutionary Feedback Loop:** The impact of decisions at an initial time step, the later time step impacts, and stakeholder utility feedback are captured as a central component of the Truth Model.

**Technology and System Development:** Technology and system development cycles are modeled through a state machine process for system life-cycle and represent the temporal aspects of developing systems and technology.

**Capturing of Uncertainty:** Uncertainty is modeled in two primary factors: temporal and performance. This capture the life-cycle uncertainty and time of received impact of decisions as well as the uncertainty in the utility feedback due to performance.

**Architecture Representation and Evaluation:** Architecture representation is depicted by the allocation to a mission and the evaluation is based on a system to mission utility transfer function. Each system allocated by a stakeholder has a specified impact on the mission outcome. There are varying profiles that modify the impact of adding another system up to a maximum impact value.

**Environment and Scenario Representation:** The environment and scenarios are represented by adding in independent adversary and cooperative stakeholder. Each of these stakeholders is modeled similarly to the stakeholder of interest. Additionally the resources and mission preferences of each stakeholder allow the representation of future scenarios outside of stakeholder actions.

**Multiple Mission Objectives:** Each stakeholder can allocate systems to a specific capabilities or missions. An allocation represents adding that system to a specific SoS. The resulting stakeholder utility is an Overall Evaluation Criteria (OEC) of the feedback from all SoS performances (mission level metrics).

**Defined SoS Engineering Reference Process:** The truth model follows the Wave model as a framework to define the progression of a SoS over time.

### *Motivation and Conceptual Description*

The full complexity problem is derived from the current development of the Future Combat Air System (FCAS) by Germany, France, and Spain (Figure 6.32). The FCAS represents a multi-national System of Systems at the early stages of development. The corner stone asset is the stealthy Next Generation Fighter (NGF) focused on near peer threats and penetrating strike. Autonomous Remote Carriers (RC) will act as teammates to the NGF. An accompanying communications layer, the Combat Cloud, is another planned acquisition that will tie together not just the new air platforms but legacy air platforms, naval assets, and space-based assets. The two new air platforms and communication infrastructure compose the core of the FCAS SoS.

There are additional assets that will integrate using the Combat Cloud infrastructure. The joint European Mutli-Role Tanker Transport (MTRR) will be integrated as a refueling asset. Legacy fighters (Tornado, Typhoon, Eurofighter, Rafale, Mirage, etc.) from all contributing countries are candidates for integration as well.

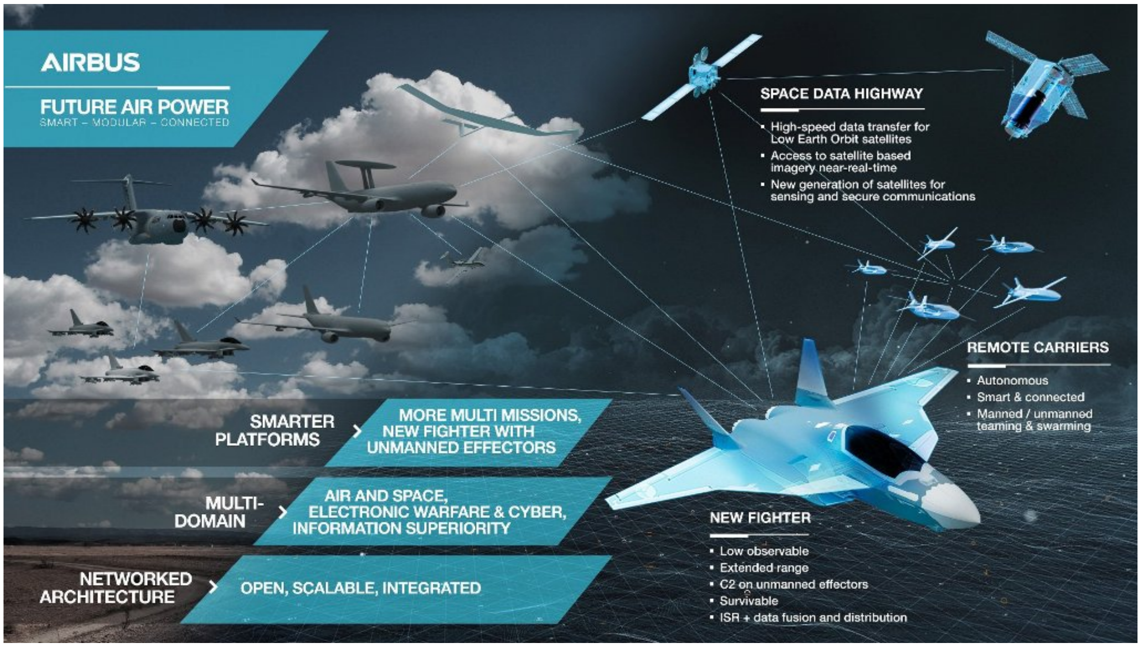

AIRBUS
FUTURE AIR POWER
SMART – MODULAR – CONNECTED
SPACE DATA HIGHWAY
▪ High-speed data transfer for Low Earth Orbit satellites
▪ Access to satellite based imagery near-real-time
▪ New generation of satellites for sensing and secure communications
REMOTE CARRIERS
▪ Autonomous
▪ Smart & connected
▪ Manned / unmanned teaming & swarming
SMARTER PLATFORMS
MORE MULTI MISSIONS, NEW FIGHTER WITH UNMANNED EFFECTORS
MULTI-DOMAIN
AIR AND SPACE, ELECTRONIC WARFARE & CYBER, INFORMATION SUPERIORITY
NETWORKED ARCHITECTURE
OPEN, SCALABLE, INTEGRATED
NEW FIGHTER
▪ Low observable
▪ Extended range
▪ C2 on unmanned effectors
▪ Survivable
▪ ISR + data fusion and distribution

Figure 6.32: FCAS OV-1 [187]

For the purposes of this work, the nominal mission set for the FCAS will be scoped to two categories: addressing a conventional threat and addressing a near-pear adversary threat. For a conventional threat, the objective is to dismantle a less-than-peer adversary's Integrated Air Defenses System (IADS). Traditionally this is done using a mixture of non-stealthy fighters, bombers, and EW aircraft. The final goal is to have air superiority over the conventional adversary controlled region. The near-peer threat consists of more advanced IADS system and requires a different approach than full air superiority. A penetrating strike scenario is used to evaluate the near-peer threat. This requires a varied set of capabilities from an air platform based SoS.

Each stakeholder, or contributing nation, of the FCAS family of programs has alternative approaches to a future architecture that could be fielded to achieve some portion of the capability desired. Germany currently deploys both Panavia Tornados interdiction/strike (IDS), Panavia Tornados electronic combat/reconnaissance (ECR), and Eurofighter Typhoons. Germany is currently phasing out the Eurofighter Typhoons while also looking at the long term replacements the Panavia Tornado (IDS and ECR). The NGF and RC to-

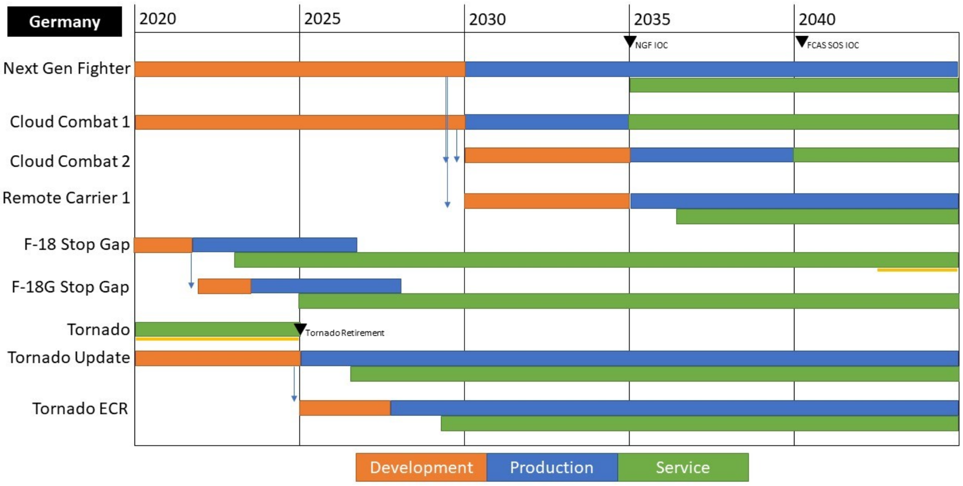


Figure 6.33: Conceptual Timeline for Germany

gether could be considered a replacement capability wise for both with the added capability of performing the penetrating strike mission. Additionally, Germany could refresh existing Tornado aircraft with new technology (avionics, sensors, etc.) for both the ECR and the IDS variant. Alternatively, Germany could purchase F/A-18 aircraft to replace the Tornado IDS and F-18G aircraft to replace the Tornado ECR.

Germany has an additional mission that is not levied on the other two stakeholders in the FCAS SoS. The additional mission is a nuclear delivery mission derived from NATO agreements. The only aircraft currently certified for nuclear delivery is the Eurofighter Typhoon. Germany will need to find a replacement for it once it is retired. The Tornado aircraft is not nuclear capable and the NGF is planned to be nuclear capable. There is no immediate stop gap from the retirement of the Typhoon and NGF. The F/A-18 is nuclear capable of nuclear weapons delivery. The need to satisfy the nuclear delivery mission and the available assets creates another dimension of complexity.

Both France and Spain are at very similar decision points regarding the path forward for their fighter aircraft. France has the aging Mirage 2000 introduced in 1995 and the newer Rafale introduced in 2006. Spain has an aging F/A-18 fleet and a newer Eurofighter Typhoon fleet. Both are looking toward NGF as a next generation replacement and both

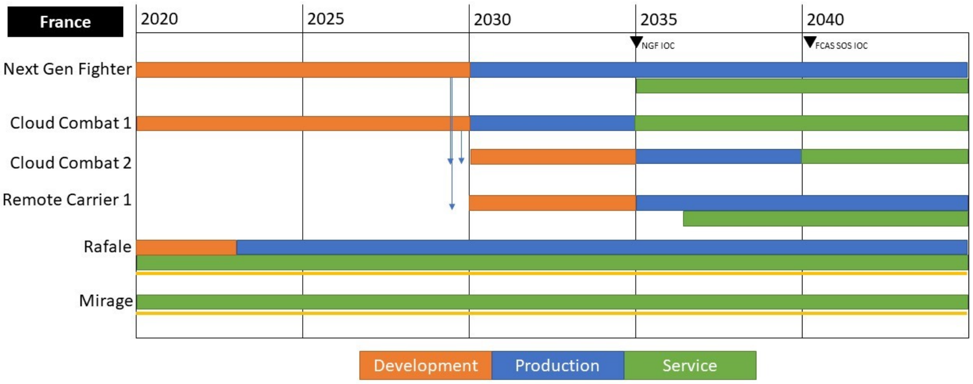


Figure 6.34: Conceptual Timeline for France

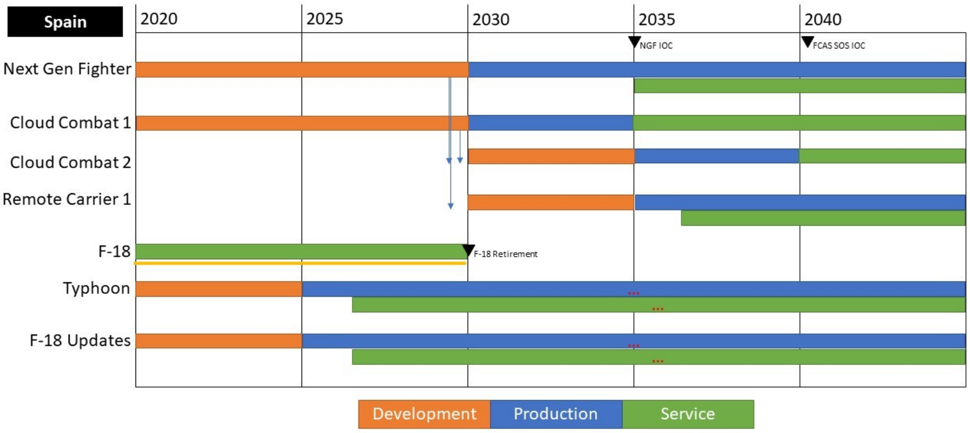


Figure 6.35: Conceptual Timeline for Spain

have alternatives to consider. France can continue to acquire Rafale aircraft. Spain can select additional F/A-18 or Eurofighter Typhoons as a replacement.

The above scenario was used as motivation for the full complexity problem. The scenario was translated into system definitions, stakeholder definitions, and mission descriptions. The concentration is placed on Germany as the stakeholder of interest. Spain and France are treated as cooperative stakeholders. The conventional and near-peer adversary are treated as non-cooperative stakeholders.

*Stakeholder Inquiries*

Each stakeholder involved will have their own evaluations conducted to answer questions of what decision should be made today and what actions should be taken. Germany is the stakeholder of interest for Experiment 3. At the initial state there are a number of example question that Germany may be seeking to answer given the set-up scenario:

- Should Germany invest in (NGF, Cloud Combat, Remote Carries) new novel systems to address emerging near peer capabilities?
- Is it worth investing in a short term solution (F/A-18 platforms) or long term solutions to keep continuity of the nuclear carry mission?
- Will new and novel solutions (NGF, Cloud Combat, Remote Carries) enable enough mission utility across all missions or should there be additional investments in EW platforms (Tornado ECR, F-18G).
- Should Germany refresh current Tornado platforms or seek an alternate route such as new development or acquisition of alternative aircraft?

*Evaluation Metrics*

Systems can be allocated to three difference scenarios that generate mission level metrics for all stakeholders based on all stakeholder allocations:

1. Conventional SEAD
2. Penetrating Strike
3. Nuclear Delivery

Each stakeholder can allocate available assets to the scenarios outlined above. Mission level metrics that represent the outcome of the scenarios are generated based on the allocations of all stakeholders. The allocation-to mission-level-metric transfer function represents the results of a engagement or mission level simulation evaluation (e.g. FLAMES, AFSIM, SEAS etc.). The utility of each stakeholder is derived from a composite of the mission level metrics based on an individual mission-level-metric-to-utility mapping.

### *Stakeholder Definition*

Each stakeholder is defined by their attributed assets, time based budget, and utility mapping. The time phased relative budgets represented in normalized currency ($\hat{\$}$) are captured in Table 6.14. Both simulation time and the corresponding conceptual year are represented. France, Germany, and the Near Peer Adversary have twice the available budget for new acquisitions and development as do Spain and the Conventional Adversary.

The budget allows the constraint of the decision space to feasible alternatives. Additionally, the time based change in budget and the uncertainty in budget can be addressed. For Experiment 3b, the relative budgets were held constant.

The Reward and Return used in the methodology is based on an overall stakeholder utility. The stakeholder utility is based on the preferences of each stakeholder. The preferences are used to create a mapping from mission metrics to stakeholder utility (Figure 6.15).

Each stakeholder has ownership and influence on the development, acquisition, and allocation of specific assets. The stakeholder ownership and influence is captured in Figure 6.16 in the system-to-stakeholder mapping. A $1$ represents a connection and a $0$ represents no connection. This matrix is used to determine the decision space of and cost incurred by each stakeholder stakeholder.

Germany, France, and Spain each have individual assets. As part of the future FCAS SoS family, there are multi-stakeholder dependent systems that rely on all three blue stakeholders. The adversary stakeholders each have individual assets assigned to them (systems 15 through 23).

Table 6.14: Experiment Set 3b: Stakeholder Budgets

| **Time Step** | **Year** | **Germany Budget ($\hat{\$}$)** | **France Budget ($\hat{\$}$)** | **Spain Budget ($\hat{\$}$)** | **Conventional Adversary Budget ($\hat{\$}$)** | **Near Peer Adversary Budget ($\hat{\$}$)** |
|---|---|---|---|---|---|---|
| 0 | 2020 | 4 | 4 | 2 | 2 | 4 |
| 1 | 2021 | 4 | 4 | 2 | 2 | 4 |
| 2 | 2022 | 4 | 4 | 2 | 2 | 4 |
| 3 | 2023 | 4 | 4 | 2 | 2 | 4 |
| 4 | 2024 | 4 | 4 | 2 | 2 | 4 |
| 5 | 2025 | 4 | 4 | 2 | 2 | 4 |
| 6 | 2026 | 4 | 4 | 2 | 2 | 4 |
| 7 | 2027 | 4 | 4 | 2 | 2 | 4 |
| 8 | 2028 | 4 | 4 | 2 | 2 | 4 |
| 9 | 2029 | 4 | 4 | 2 | 2 | 4 |
| 10 | 2030 | 4 | 4 | 2 | 2 | 4 |
| 11 | 2031 | 4 | 4 | 2 | 2 | 4 |
| 12 | 2032 | 4 | 4 | 2 | 2 | 4 |
| 13 | 2033 | 4 | 4 | 2 | 2 | 4 |
| 14 | 2034 | 4 | 4 | 2 | 2 | 4 |
| 15 | 2035 | 4 | 4 | 2 | 2 | 4 |
| 16 | 2036 | 4 | 4 | 2 | 2 | 4 |
| 17 | 2037 | 4 | 4 | 2 | 2 | 4 |
| 18 | 2038 | 4 | 4 | 2 | 2 | 4 |
| 19 | 2039 | 4 | 4 | 2 | 2 | 4 |
| 20 | 2040 | 4 | 4 | 2 | 2 | 4 |

Table 6.15: Experiment Set 3b: Stakeholder Mission Preference

| **Stakeholder** | **Conventional SEAD** | **Penetrating Strike** | **Nuclear Delivery** |
|---|---|---|---|
| Germany | 0.4 | 0.4 | 0.2 |
| France | 0.7 | 0.3 | 0 |
| Spain | 0.9 | 0.1 | 0 |
| Conventional Adversary | -1 | 0 | 0 |
| Near Peer Adversary | 0 | -1 | 0 |

*System Definition*

The system definition is defined by the system development mapping, the system performance, system cost, and the system timelines. The system development mapping allows sequential develop, refresh, and acquire decision space to be derived. The system cost for development, acquisition, and refresh enables the impact of budget and budget uncertainty to be captured. The system development, refresh, and acquisition timelines enable the impact of long term decision feedback and uncertainty to be captured.

The system development mapping (Figure 6.36). The system dependency mapping captures system predecessors and successors. Once a system is developed the dependent systems are then available for development. For example, the NGF is a predecessor to the RC system development. The Cloud Combat for legacy systems has not predecessor itself but is a predecessor to the future system Cloud Combat. The Tornado is a predecessor to the Tornado refresh, Tornado ECR, and the F/A-18 developments. The system development and refresh dependencies defined by the mapping allow the development of sequential acquisition decisions.

Each system is represented by four inputs per mission: a mean performance, perfor-

Table 6.16: Experiment Set 3b: Stakeholder System Ownership and Influence

| **System Name** | **System Number** | **Germany** | **France** | **Spain** | **Conventional Adversary** | **Near Peer Adversary** |
|---|---|---|---|---|---|---|
| Next Gen Fighter | 1 | 1 | 1 | 1 | 0 | 0 |
| Combat Cloud (legacy) | 2 | 1 | 1 | 1 | 0 | 0 |
| Combat Clout (next gen) | 3 | 1 | 1 | 1 | 0 | 0 |
| Remote Carrier | 4 | 1 | 1 | 1 | 0 | 0 |
| Tornado | 5 | 1 | 0 | 0 | 0 | 0 |
| Tornado Refresh | 6 | 1 | 0 | 0 | 0 | 0 |
| Tornado ECR | 7 | 1 | 0 | 0 | 0 | 0 |
| F/A-18 (German Variant) | 8 | 1 | 0 | 0 | 0 | 0 |
| F-18G (German Variant) | 9 | 1 | 0 | 0 | 0 | 0 |
| Rafale | 10 | 0 | 1 | 0 | 0 | 0 |
| Mirage | 11 | 0 | 1 | 0 | 0 | 0 |
| F/A-18 (Spanish Variant) | 12 | 0 | 0 | 1 | 0 | 0 |
| Eurofighter | 13 | 0 | 0 | 1 | 0 | 0 |
| F/A-18 Update (Spanish Variant) | 14 | 0 | 0 | 1 | 0 | 0 |
| Conventional EW Radar | 15 | 0 | 0 | 0 | 1 | 0 |
| Conventional TTR Radar | 16 | 0 | 0 | 0 | 1 | 0 |
| Conventional SAM | 17 | 0 | 0 | 0 | 1 | 0 |
| Near Peer EW Radar | 18 | 0 | 0 | 0 | 0 | 1 |
| Near Peer TTR Radar | 19 | 0 | 0 | 0 | 0 | 1 |
| Near Peer SAM | 20 | 0 | 0 | 0 | 0 | 1 |
| Near Peer Next Gen EW Radar | 21 | 0 | 0 | 0 | 0 | 1 |
| Near Peer Next Gen TTR Radar | 22 | 0 | 0 | 0 | 0 | 1 |
| Near Peer Next Gen SAM | 23 | 0 | 0 | 0 | 0 | 1 |

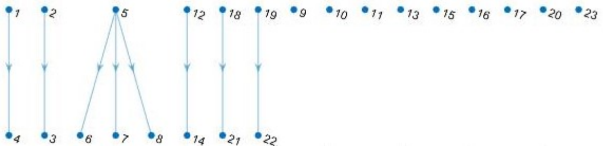


Figure 6.36: System Development Mapping

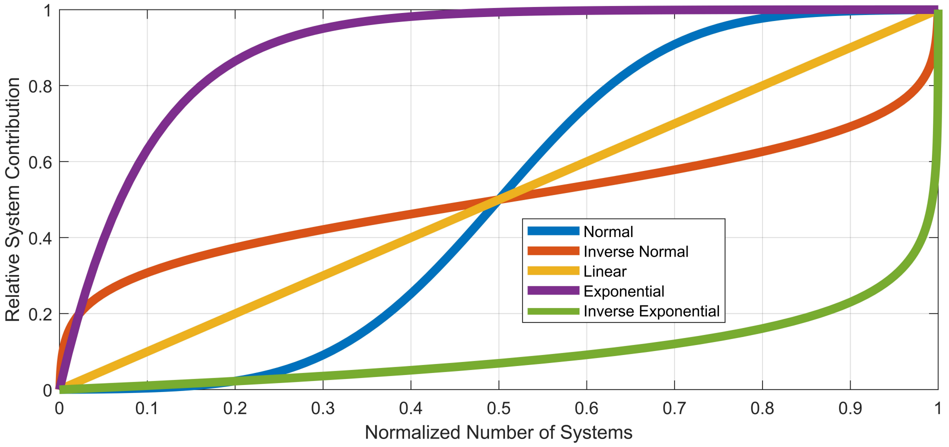


Figure 6.37: Alloaction to Utility Functions Examples

mance variance, maximum asset impact, and utility as a function of number of assets allocated. The system contribution to the mission level metric is captured in Figure 6.17. The table represents the raw performance for each individual system allocated to the specified mission. Each system type has a specified maximum impact allocation.

If more systems are allocated beyond the maximum impact allocation amount there is no incremental benefit. Additionally, the contribution of each incremental system is not necessarily linear. Varying utility functions are used to modify the incremental impact of additional systems (Figure 6.37. The customization of incremental impact provides a more nuanced result than a pure linear uncapped mapping. The non-linear function allows a more sophisticated mission level simulation to be represented in the test bed.

The normalized system development and acquisition cost are captured via yearly mean and variance costs (Figure 6.18). Additionally, the development and acquisition time is represented by a mean and variance. The timeline and cost uncertainty are combined for a final total cost impact. The last timeline is the deployment time which is left deterministic for the purposes of this experiment.

Table 6.17: Experiment Set 3b: System Performance

| **System Name** | **Conventional SEAD** | | **Penetrating Strike** | | **Nuclear Delivery** | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\boldsymbol{3\sigma}$ | $\mu$ | $\boldsymbol{3\sigma}$ | $\mu$ | $\boldsymbol{3\sigma}$ |
| Next Gen Fighter | 20 | 10 | 30 | 8 | 20 | 5 |
| Combat Cloud (legacy) | 15 | 8 | 0 | 0 | 0 | 0 |
| Combat Cloud (next gen) | 0 | 0 | 20 | 4 | 0 | 0 |
| Remote Carrier | 0 | 0 | 10 | 2 | 0 | 0 |
| Tornado | 5 | 1 | 0 | 0 | 0 | 0 |
| Tornado Refresh | 10 | 1 | 0 | 0 | 0 | 0 |
| Tornado ECR | 15 | 2 | 5 | 1 | 0 | 0 |
| F/A-18 (German variant) | 5 | 1 | 0 | 0 | 15 | 2 |
| F-18G (German variant) | 15 | 5 | 5 | 1 | 0 | 0 |
| Rafale | 5 | 1 | 0 | 0 | 0 | 0 |
| Mirage | 5 | 1 | 0 | 0 | 0 | 0 |
| F/A-18 (Spanish variant) | 5 | 2 | 0 | 0 | 0 | 0 |
| Eurofighter | 10 | 1 | 0 | 0 | 0 | 0 |
| F/A-18 Update (Spanish variant) | 10 | 2 | 0 | 0 | 0 | 0 |
| Conventional EW Radar | -25 | 5 | 0 | 0 | 0 | 0 |
| Conventional TTR Radar | -20 | 4 | 0 | 0 | 0 | 0 |
| Conventional SAM | -15 | 3 | 0 | 0 | 0 | 0 |
| Near Peer EW Radar | 0 | 0 | -10 | 2 | 0 | 0 |
| Near Peer TTR Radar | 0 | 0 | -5 | 1 | 0 | 0 |
| Near Peer SAM | 0 | 0 | -5 | 1 | 0 | 0 |
| Near Peer Next Gen EW Radar | 0 | 0 | -40 | 8 | 0 | 0 |
| Near Peer Next Gen TTR Radar | 0 | 0 | -40 | 8 | 0 | 0 |
| Near Peer Next Gen SAM | 0 | 0 | -40 | 8 | 0 | 0 |

Table 6.18: Experiment Set 3b: System Cost and Timeline Definition

| **System Name** | **Mean Acquisition Cost ($\hat{\$}/yr$)** | **Mean Development Cost ($\hat{\$}$/yr)** | **Mean Development Time (years)** | **Development Time Uncertainty (years, $3\sigma$)** | **Mean Acquisition Time (years)** | **Acquisition Time Uncertainty (years, $3\sigma$)** | **Deployment Time ($yr$)** |
|---|---|---|---|---|---|---|---|
| Next Gen Fighter | 2 | 1.5 | 8 | 5 | 5 | 2 | 20 |
| Combat Cloud (legacy) | 2 | 1 | 8 | 5 | 5 | 2 | $\infty$ |
| Combat Clout (next gen) | 2 | 1 | 5 | 5 | 5 | 2 | $\infty$ |
| Remote Carrier | 2 | 1 | 5 | 5 | 5 | 2 | 20 |
| Tornado | - | - | - | - | - | - | 20 |
| Tornado Refresh | 2 | 1 | 3 | 1 | 3 | 1 | 20 |
| Tornado ECR | 2 | 1 | 3 | 1 | 5 | 2 | 20 |
| F/A-18 (German variant) | 2 | 1 | 2 | 0 | 3 | 0 | 20 |
| F-18G (German variant) | 2 | 1 | 2 | 1 | 5 | 1 | 20 |
| Rafale | - | 1 | - | - | 5 | 0 | 20 |
| Mirage | - | - | - | - | - | - | 20 |
| F/A-18 (Spanish variant) | - | - | - | - | - | - | 20 |
| Eurofighter | - | 1 | - | - | 2 | 0 | 20 |
| F/A-18 Update (Spanish variant) | 2 | 1 | 2 | 1 | 3 | 0 | 20 |
| Conventional EW Radar | - | 1 | - | - | 1 | 0 | 15 |
| Conventional TTR Radar | - | 1 | - | - | 1 | 0 | 15 |
| Conventional SAM | - | 1 | - | - | 1 | 0 | 15 |
| Near Peer EW Radar | - | 1 | - | - | 1 | 0 | 15 |
| Near Peer TTR Radar | - | 1 | - | - | 1 | 0 | 15 |
| Near Peer SAM | - | 1 | - | - | 1 | 0 | 15 |
| Near Peer Next Gen EW Radar | 2 | 1 | 7 | 0 | 1 | 0 | 15 |
| Near Peer Next Gen TTR Radar | 2 | 1 | 7 | 0 | 1 | 0 | 15 |
| Near Peer Next Gen SAM | 2 | 1 | 7 | 0 | 1 | 0 | 15 |

*Benchmark: Optimal Strategy*

The goal of RL and ADP is to identify the optimal policy that should be followed through sequential states and actions. An optimal policy will identify a single action that should be taken in each state. The optimal policy is deterministic and not stochastic. An optimal policy represents a 'optimal' stakeholder strategy. Traditional methods, as discussed in Chapter 4, do not account for the variance in outcomes. Roughly, only the mean return is used to determine an optimal strategy. it should be noted that 'optimal' refers to the traditional name given to the solution and optimality will differ depending on the desired outcome. If uncertainty is necessary to consider, traditional optimal policy methods will not address the need.

For Experiment 3, action-value and policy iterations are used to calculate the approximate solution to the meta-model MDP (Figure 6.38). The meta-model MDP solution is then mapped back to the full state space using the same mapping used for the risk-based policy methods. The meta-model MDP is sampled using Monte-Carlo samples using first an initializing policy ($\pi_0$). The state-action value matrix is updated using an on-policy n-TD SARSA method. For each MC sample, a state Return is calculated using Equation 6.1. The Return is used to update the action-value function using Equation 6.2.

$$G_e(s) = \sum_{i=0}^{i=t} \gamma^i r_i \tag{6.1}$$

where $e$ is the MC episode, $s$ is an episode state, $r$ is a state-action reward, $t$ is the time horizon, and $\gamma$ is the discount.

$$Q_\pi(s,a) \leftarrow Q_\pi(s,a) + \alpha(G_e(s) - Q_\pi(s,a)) \tag{6.2}$$

where $\pi$ is the current policy, $Q$ is the action-value function, $s$ is a episode state, $a$ is an episode action.

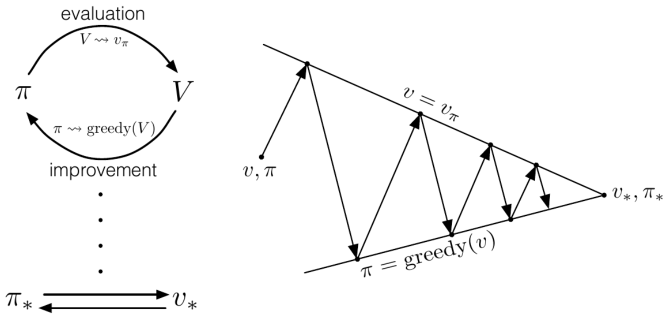

evaluation
$V \rightsquigarrow v_\pi$
$\pi$
$V$
$\pi \rightsquigarrow \mathrm{greedy}(V)$
improvement
$\pi_*$
$v_*$
$v, \pi$
$v = v_\pi$
$v_*, \pi_*$
$\pi = \mathrm{greedy}(v)$

Figure 6.38: Policy and Value Iterations Diagram [158]

The policy is then updated once the policy dependent action-value function ($Q_\pi(s, a)$) is calculated. A new complete policy is created for each state using the softmax function (Equation 6.3) which normalizes relative arbitrary scored values. The policy is updated using a monotonically decreasing $\alpha$ value based on the maximum number of policy iterations (Equation 6.4).

$$\pi(s, a_o) = \frac{e^{Q(s,a_o)}}{\sum_{a \in A} e^{Q(s,a)}} \tag{6.3}$$

where $A$ are all available actions in a given state and $a_o$ is the specfic action of interest in a given state.

$$\pi \leftarrow (1 - \lambda) * \pi + \lambda * \pi' \tag{6.4}$$

where $\lambda$ is the update factor, $\pi$ is the policy in use, $\pi'$ is the new policy based on the action-value iteration.

# CHAPTER 7
# RESULTS AND ANALYSIS

The results and analysis of each of the executed experiments described in Chapter 6 are captured in Chapter 7. Experiment Set 1 results demonstrate the identification of Pareto efficient actions and the capabilities provided by applying the risk-based policy algorithm to scenarios with varying degrees of complexity. Experiment Set 2 demonstrates the ability to maintain usable risk-based metrics when evaluating a meta-model generated from a compressed state-space. Lastly, Experiment Set 3 results demonstrate the ability to derive nuanced information from the risk-based evaluation of the meta-model and benchmarks the methodology against optimal policy solutions.

## 7.1 Experiment Set 1: Risk-Based Policy Development

Experiment Set 1 results first explore the application of the risk-based policy method to explicitly defined MDPs. Second, the algorithm is applied to full MDPs derived from Truth Model scenarios of increasing complexity. The analysis demonstrates that Hypothesis 1 to be true.

### 7.1.1 Experiment Set 1a: Explicit MDPs

Experiment Set 1a defines explicit MDPs and uses the constructed MDP for direct evaluation. The constructed MDP is directly used for evaluation using the risk-based policy algorithm.

*Experiment Set 1a: Simple Stated MDPs*

The purpose of Experiment Set 1a Case 1 is to demonstrate the identification of Pareto frontier actions by comparing the change in risk-based policies as the risk-tolerance level

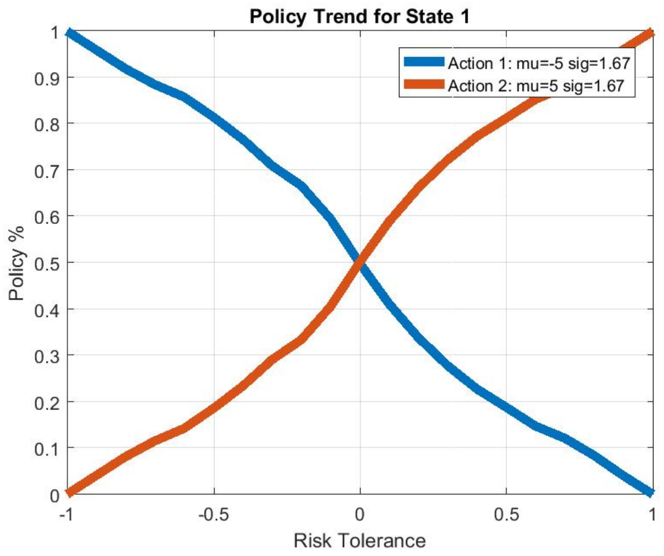


Figure 7.1: Experiment Set 1a Case 1: State One, Two Actions, Equal Reward Mean RTSP

of a stakeholder in varied explicitly defined scenarios. The anticipated results are compared with the policy risk-tolerance sensitivity profiles for each scenario to demonstrate the identification of the Pareto frontier. The risk-tolerance sensitivity profiles plot the policy as a function of risk-tolerance.

Scenario 1 examines the simplest of cases. A repeating state with two available decisions for the stakeholder. As the risk-tolerance level is moved from minimum to maximum, the expected result is to see the lower mean action selected near the minimum and the higher mean action at the maximum. A gradual symmetrical swapping of preference is anticipated in between. The results for the risk-tolerance sensitivity plot for the initial state policies are shown in Figure 7.1.

The results for Scenario 1 show the anticipated pattern of gradual change in action preference as well as the symmetry about the minimum risk point ($\xi = 0$).

Scenario 2 adds an addition available action with an intermediate mean and equal variance. The anticipated result is an additional preference near the minimum risk point for the new intermediate action. This pattern can be observed in Figure 7.2.

The antithesis of Scenario 2 is Scenario 3, where only the variance is modified across

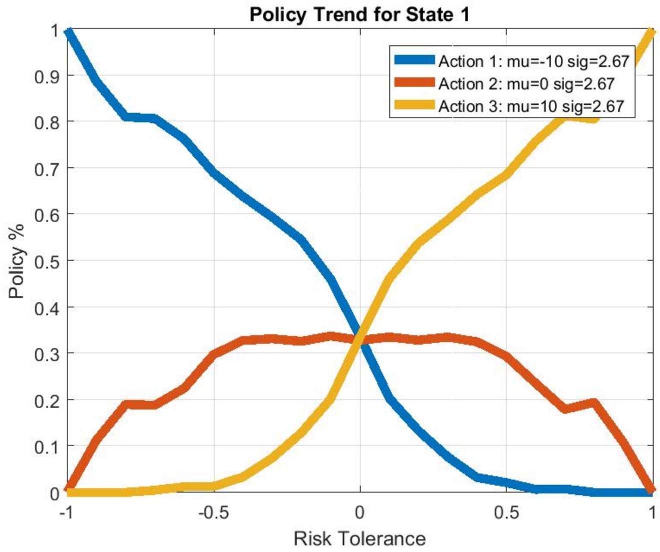


Figure 7.2: Experiment Set 1a Case 1: State One, Three Actions, Equal Reward Mean RTSP

actions. Here, there is no identified Pareto frontier and the results should display that. It is anticipated that there should be no distinction in preference as the risk-tolerance is varied but for an increase in the minimum variance action near the minimum risk point. Figure 7.3 shows the lack of policy sensitivity everywhere along the risk-tolerance axis sans near the minimum risk point.

Scenario 4 demonstrates the ability to find the pareto frontier formed by a linear mean-variance action set. Figure 7.4 shows the linear trend in mean-variance leads to a near constant relative alignment for the lower half of the Pareto frontier with the expected trend for a near Pareto frontier profile formed by the linear mean-variance trend. The Pareto optimal profile for $\xi < 0$ is near a point and represents a near constant policy case. A representation of this phenomenon can be seen in the problem construction (Figure 6.6).

Scenario 5 and 6 demonstrate the ability for the method to identify the Pareto frontier, both efficient and anti-efficient, while identifying the Pareto inefficient actions as well. The preference of an action is anticipated to near zero for all risk-tolerance levels as the action is moved from near Pareto efficiency to significantly out of the Pareto efficient region. Action

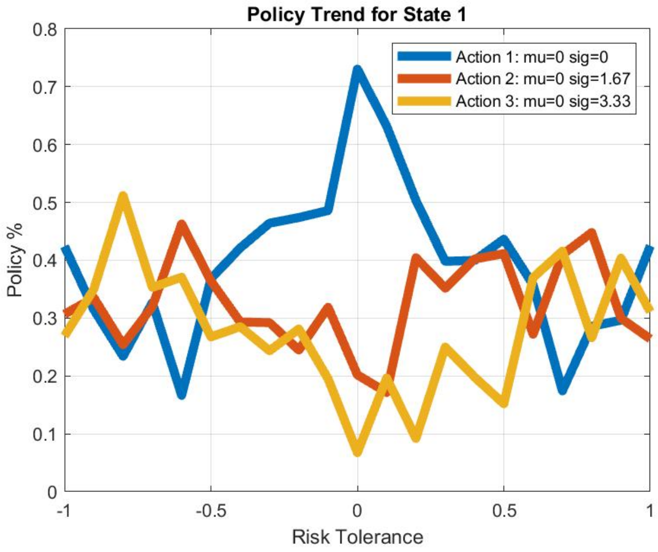


Figure 7.3: Experiment Set 1a Case 1: State One, Three Actions, Equal Reward Variance RTSP

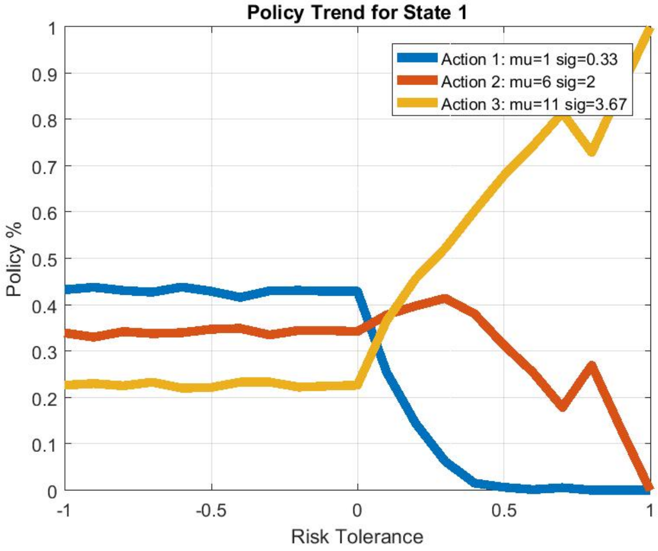


Figure 7.4: Experiment Set 1a Case 1: State One, Three Actions, Linear Reward RTSP

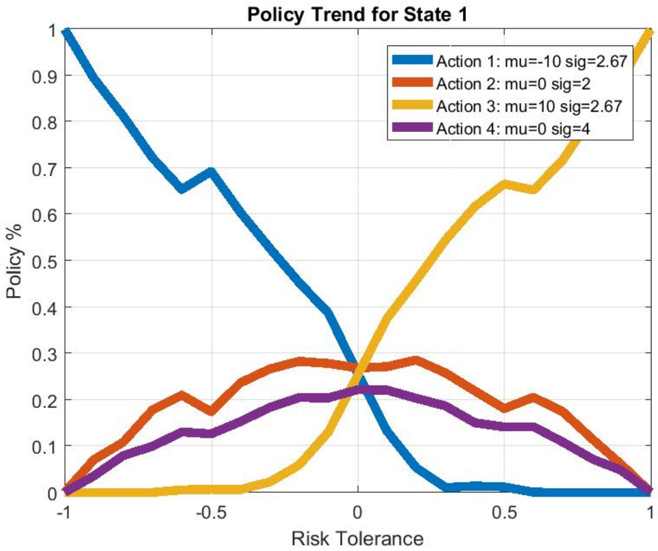


Figure 7.5: Experiment Set 1a Case 1: State One, Four Actions, One Mild Pareto Inefficient RTSP

4 in Figure 7.5 and Figure 7.6 represents the inefficient action. The action is made more inefficient by increasing the variance from Scenario 5 (Figure 7.5) to Scenario 6 (7.6). A decrease in overall preference across risk-tolerance can be observed in Action 4. The same Pareto frontier identified in Scenario 2 can be identified in Scenario 5 and 6.

Scenario 7 shows the results of an expanded Pareto frontier action space with no inefficiencies (Figure 7.7). Similar to Scenario 2, the anticipated results are for individual actions to peak near where a similar single step-action Pareto frontier would peak. The risk-tolerance sensitivity of each action demonstrates the appropriate peak.

Three of the actions have direct corollaries to the three actions seen in Scenario 2. Action 1 is the highest-risk lowest-mean action with Action 7 being the highest-risk highest-mean. They are symmetrically biased at the extremes of risk-tolerance as anticipated. Action 4 is the minimum risk action and peaks as anticipated near the minimum risk risk-tolerance level. Actions 2 and 3 lie on the Pareto inefficient frontier and peak in order as $\xi$ varies from $-1$ to $0$. Similarly and in a symmetric manor about $\xi = 0$ axis Action 5 and 6 peak in the anticipated trend.

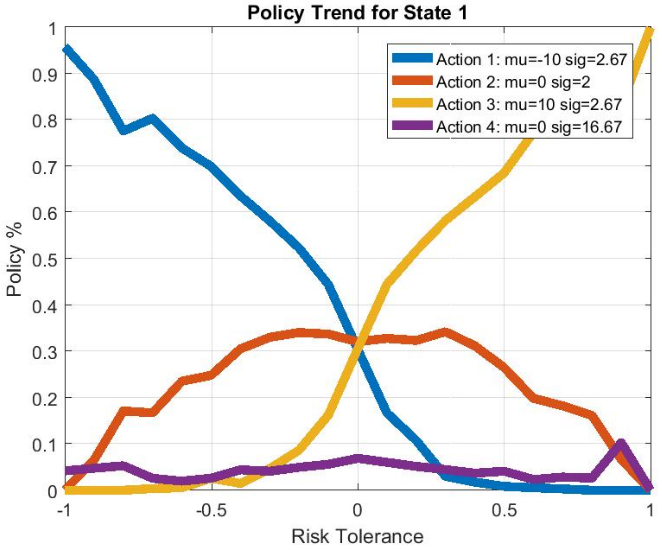


Figure 7.6: Experiment Set 1a Case 1: State One, Four Actions, One Significant Pareto Inefficient RTSP

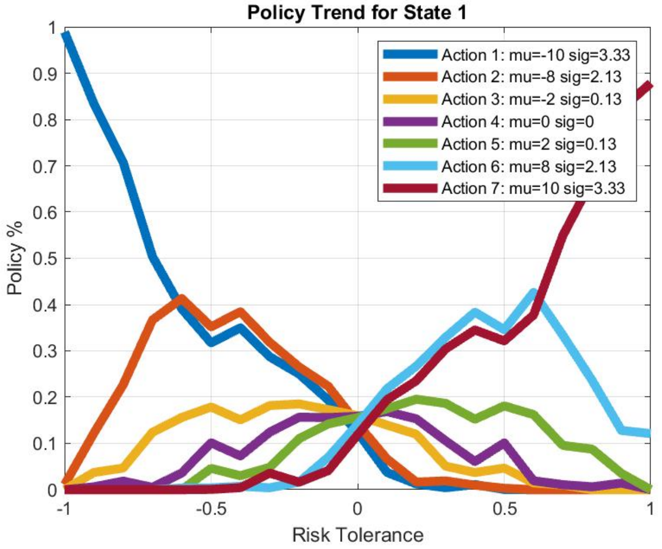


Figure 7.7: Experiment Set 1a Case 1: State One, Seven Actions, Pareto Inefficient RTSP

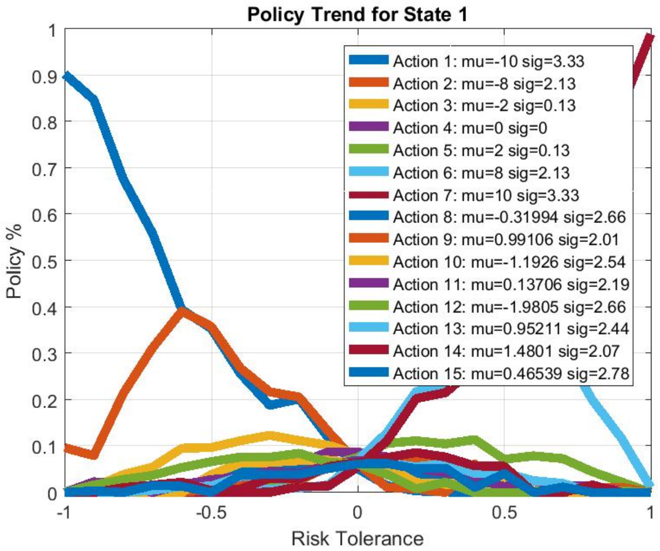


Figure 7.8: Experiment Set 1a Case 1: State One, Seven Pareto Efficient Actions, Eight Mild Pareto Inefficient RTSP

Scenario 8 and 9 once again add Pareto inefficient actions of varying magnitude. Scenario 8 adds mild Pareto inefficient actions (Action 8 through 15) and Scenario 9 adds significantly Pareto inefficient actions (Actions 8 through 16) by increasing the action reward variance. The risk-tolerance sensitivity profile for Scenario 8 is shown in Figure 7.8 and for Scenario 9 in Figure 7.9. The relative preference for inefficient actions decreases as the inefficiency of the actions are increased.

*Experiment Set 1a Case 2: Short and Long Term Stakeholder Preferences*

Scenario 1 examines an explicit set-up with two actions that lead not to a similar state with similar action but two separate states with varying action results. This adds adds a mutli-tiered state to with near repeated actions on top of Case 1. The resulting State 1 RTSP is shown in Figure 7.10. This demonstrates the same expected RTSP patter as scene in Case 1, Scenario 1. Case 1 had repeating decisions and Case 2 introduced non-repeated decisions resulting in varied states.

Scenario 2 examines the impact of multiple actions resulting in varied states. The con-

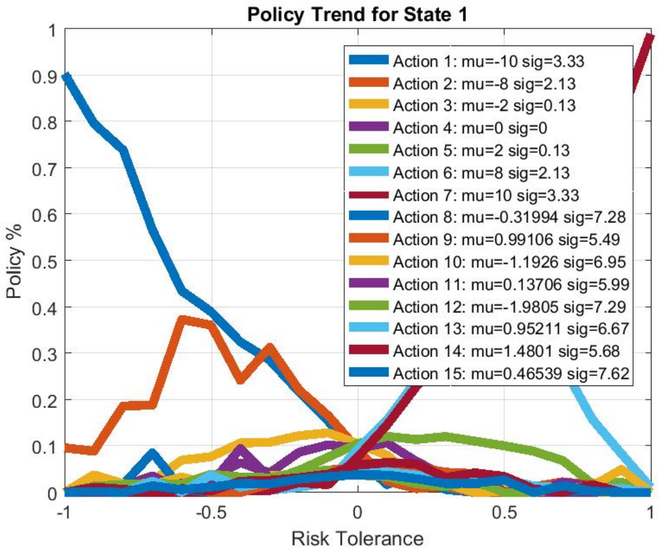


Figure 7.9: Experiment Set 1a Case 1: State One, Seven Pareto Efficient Actions, Eight Significant Pareto Inefficient RTSP

structed MDP is created to demonstrate the impact of short Reward versus long term Return. The swapping of selected preferences at low risk-tolerances and high-risk tolerances is demonstrated in Figure 7.11, the State 1 RTSP.

Scenario 3 and Scenario 4 examine the impact of adding non-repeated Pareto inefficient actions into the decision space. The State 1 RTSP (Figure 7.12) shows the mild Pareto inefficient action (shown in blue) to never be preferred over the other three Pareto efficient actions under a mild in-efficiency setting. Figure 7.13 shows the RTSP of State 1 with the significant Pareto inefficient action included. The impact of the inefficient action (again shown in blue) is clearly even more insignificant having a even more reduced policy representation across all risk-tolerance levels.

### *Experiment Set 1a Case 3: Fully Random MDP*

Case 3 uses randomly generated MDPs to test the policy generation algorithm. Two scenarios of varying complexity are used to exercise the algorithm. The first scenario consists of a maximum of three actions per state. Each action has the opportunity to develop to

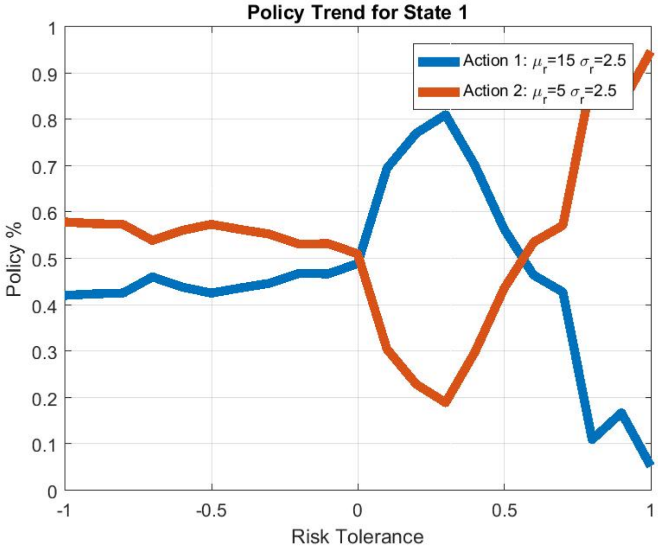


Figure 7.10: Experiment Set 1a Case 2 Scenario 1: State 1 RTSP

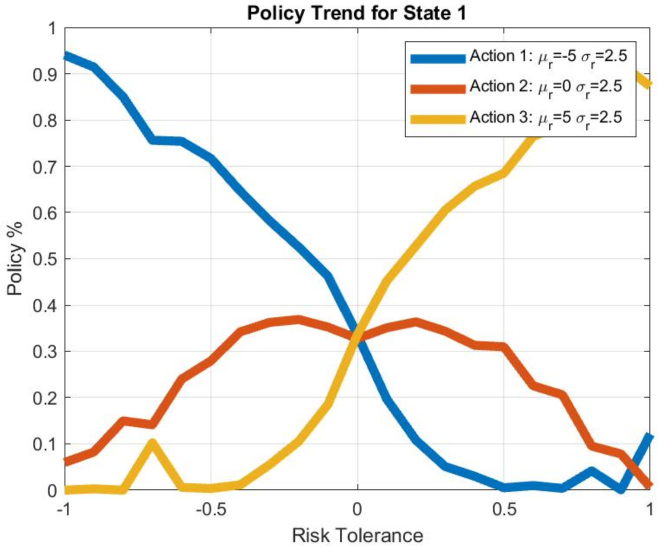


Figure 7.11: Experiment Set 1a Case 2 Scenario 2: State 1 RTSP

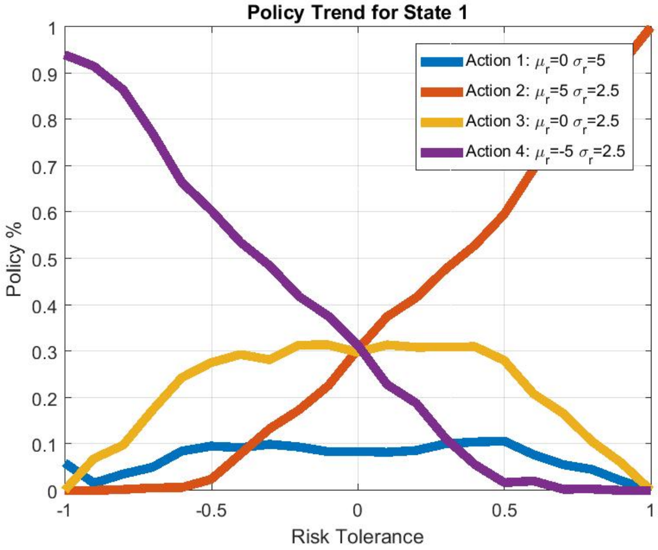


Figure 7.12: Experiment Set 1a Case 2 Scenario 3: State 1 RTSP

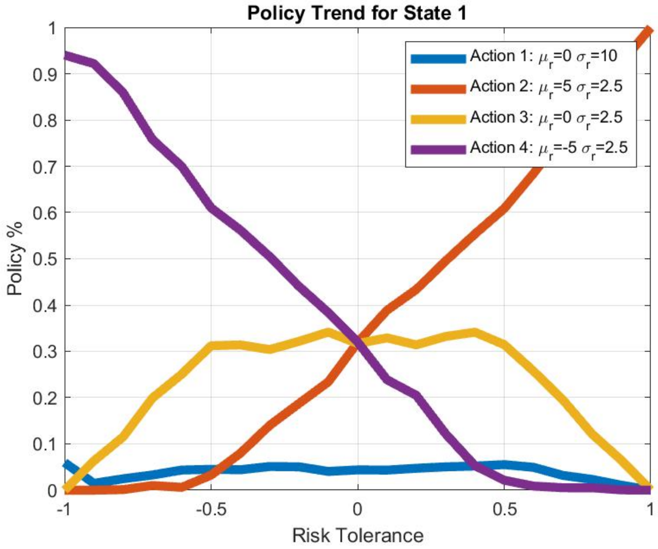


Figure 7.13: Experiment Set 1a Case 2 Scenario 4: State 1 RTSP

multiple states. The second scenario increases the maximum actions to five for any given state and increases the number of resulting states per action.

The random MDPs produce scenarios that are still simple enough to understand and evaluate without the aid of algorithms. The resulting policy trends for each MDP are evaluated along side the randomly generated MDP itself. This allows the determination of the effectiveness of the policy generation algorithm to me measured. States where decisions are present are identified and the policy is evaluated against the anticipated result based on the mean and variance of the reward graph. Moving from a simple defined MDP to a multi-stage, action-decoupled MDP allows the impact of Reward versus Return to be analyzed. Previous experiments tested either single step Returns or repeat-reward profiles where Reward can be approximately considered equal to Return.

**Scenario 1: Three Action Random MDP** The three state randomly generated MDP structure is captured in Figure 7.14 along with the states of interest. The selected states of interest are based on explicit decision points. The MDP, including $s-a-s'$ reward characteristics ($\mu_r(s,a,s')$, $\sigma_r^2(s,a,s')$), is fully captured in Figures 7.15 and Figure 7.16. For example, State 2 is a decision point. The Stakeholder can choose Action 1 ($\mu_r(2,1,4) = 5.1$, $\sigma_r^2(2,1,4) = 85$) or Action 3 ($\mu_r(2,3,5) = 5.1$, $\sigma_r^2(2,3,5) = 14$). The State 2 rewards can be visualized in Figure 7.18.

The resulting Risk-Tolerance Sensitivity Profile generated using the risk-based policy algorithm for State 2 (Figure 7.17) shows an expected trend between the lower-mean, lower-variance action (Action 1) and the higher-mean, higher-variance action (Action 3) as expected. The additional metric of Return ($R(\xi,s,a)$) can be used to evaluate the the policy trends in the Risk-Tolerance Sensitivity Profile (Figure 7.19). The return is presented as a function of the risk-tolerance ($\xi$). The risk-tolerance is varied from worst ($\xi = -1$), to minimum risk ($\xi = 0$), and then maximum risk ($\xi = 1$) just as in the Risk-Tolerance Sensitivity Profile. It is clear that the relative positioning on the mean-variance map of the two

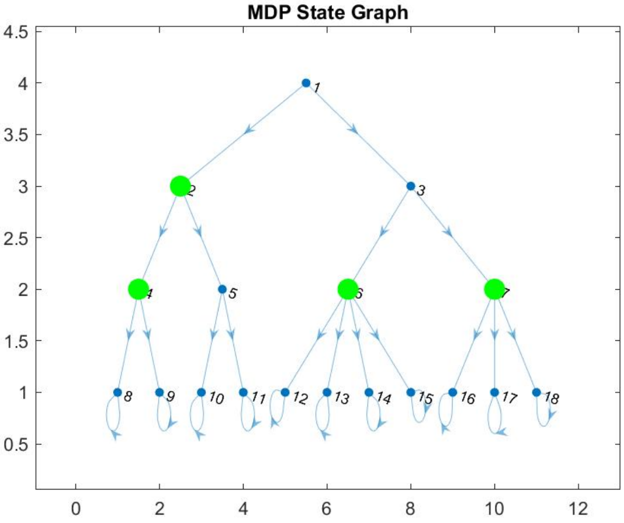


Figure 7.14: Experiment Set 1a Case 3 Scenario 1: MDP Graph and Highlighted Selected States of Interest

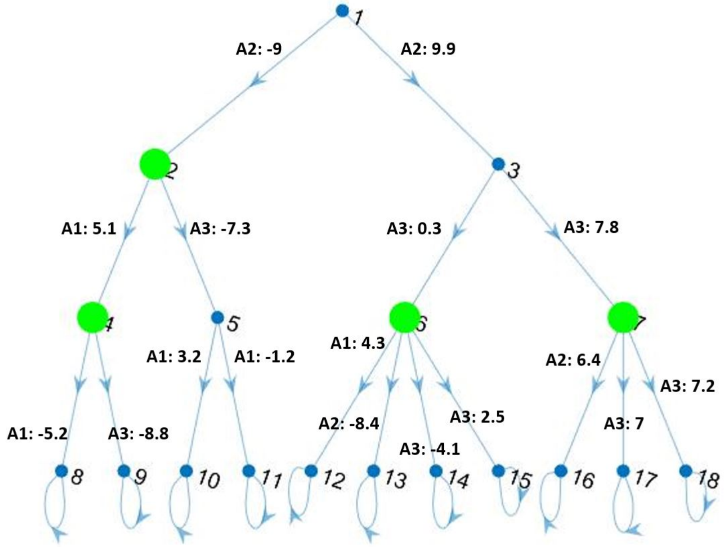


Figure 7.15: Experiment Set 1a Case 3 Scenario 1: Reward Mean

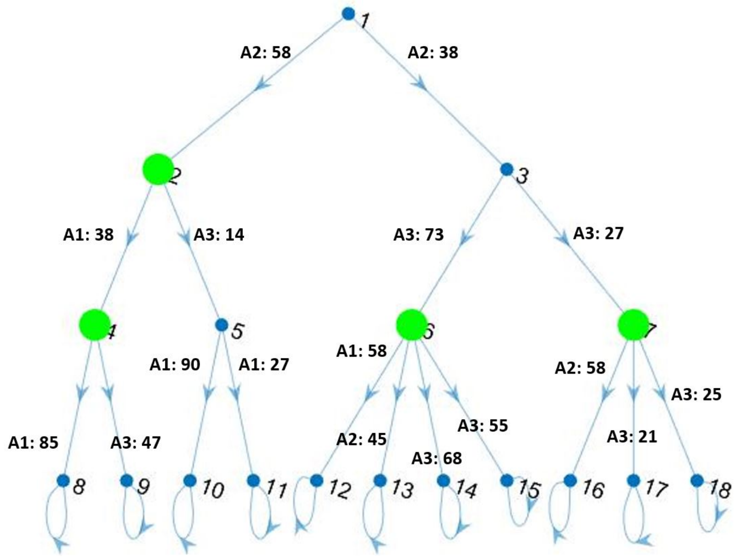


Figure 7.16: Experiment Set 1a Case 3 Scenario 1: Reward Variance

actions remains constant as the risk-tolerance is varied and as they shift the mean-variance plane. The relative position consistency yields the Risk-Tolerance Sensitivity for State 2.

State 4 demonstrates the same trends seen in State 2. The lower-mean, lower-variance versus higher-mean, higher variance reward (Figure 7.21) yields the anticipated policy trend is seen in the Risk-Tolerance Sensitivity Profile (Figure 7.20). The Return mean-variance plot shows the relative position consistency (Figure 7.22) is maintained as $\xi$ is varied.

State 6 adds two additional complexities yet to be seen under previous conditions evaluated. Three actions are present and one of the actions (Action 3) can result in two different states. Each state represents a different $s - a - s' - r$ tuple (Figure 7.24). It might be expected to see Action 2 preferred at the worst $\xi$ case, Action 1 preferred at the minimum risk case, and Action 3 and maximum risk case based the results from Experiment Set 1a Case 1 and Case 2. The resulting Risk-Tolerance Sensitivity Profile (Figure 7.23) shows Action 3 preferred with Action 1 less preferred at maximum risk.

The explanation is found in the mean-variance plot of State 6 Return (Figure 7.25). The combination and transition probabilities show that in the worst case ($\xi = -1$) that Action 3

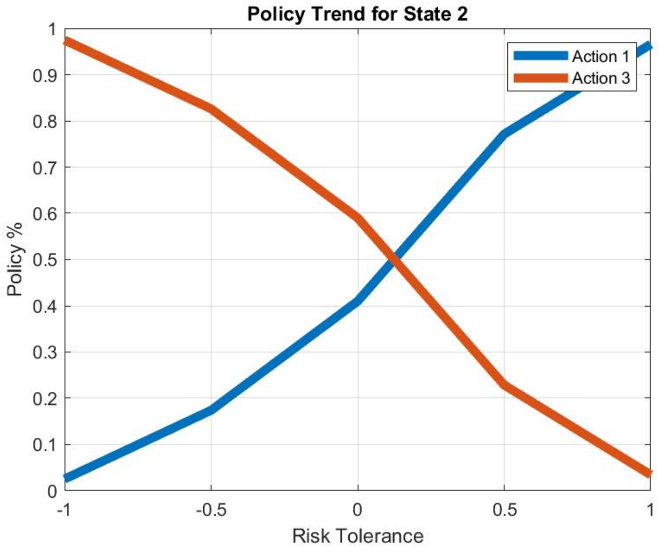


Figure 7.17: Experiment Set 1a Case 3 Scenario 1: RTSP for State 2

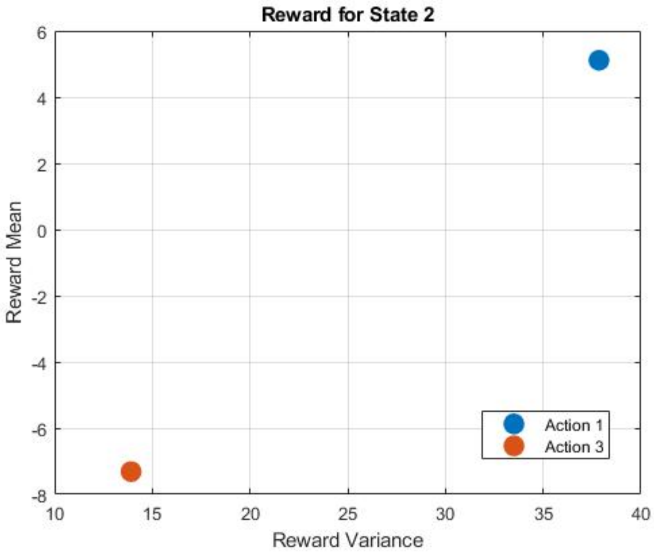


Figure 7.18: Experiment Set 1a Case 3 Scenario 1: Immediate Reward for State 2

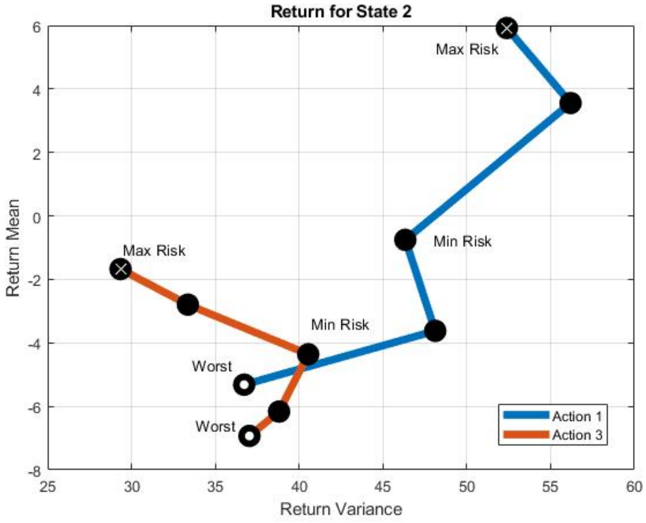


Figure 7.19: Experiment Set 1a Case 3 Scenario 1: Long Term Return for State 2

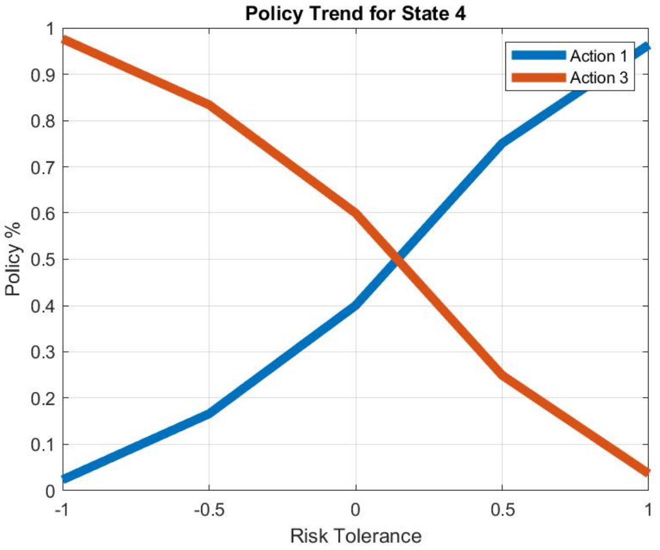


Figure 7.20: Experiment Set 1a Case 3 Scenario 1: RTSP for State 4

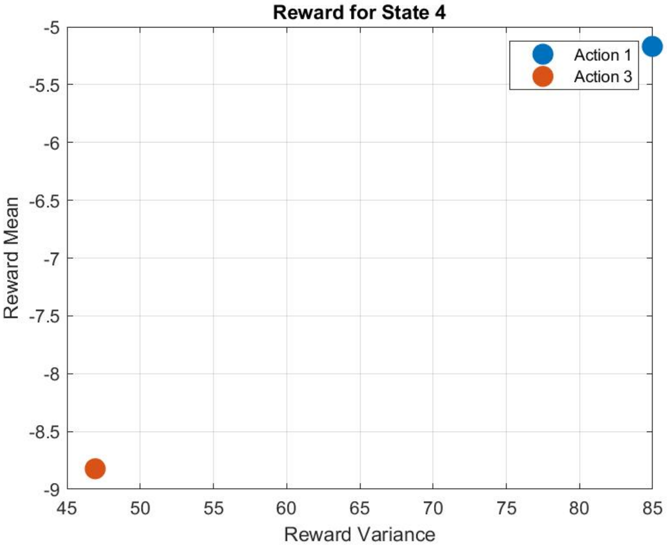


Figure 7.21: Experiment Set 1a Case 3 Scenario 1: Immediate Reward for State 4

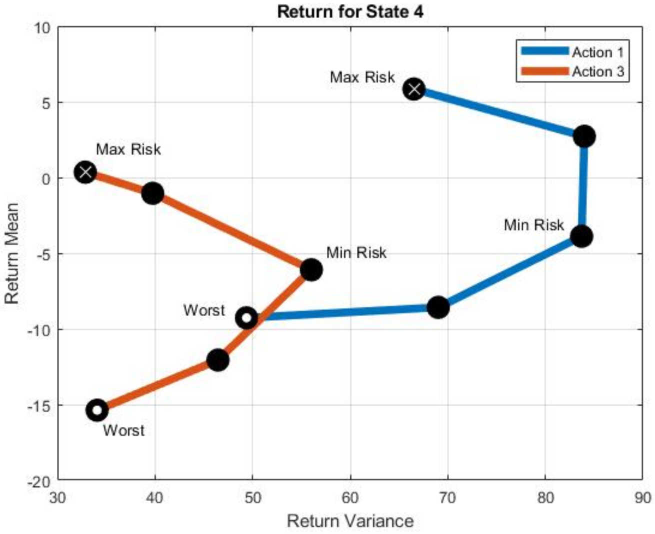


Figure 7.22: Experiment Set 1a Case 3 Scenario 1: Long Term Return for State 4

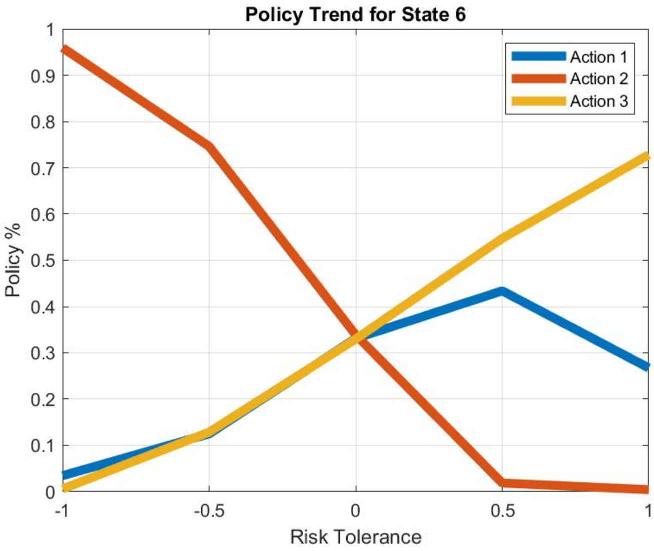


Figure 7.23: Experiment Set 1a Case 3 Scenario 1: RTSP for State 6

is considered the highest risk option (note Action 2 dominates at this risk-tolerance because it is the worst option). As $\xi$ is varied toward 1 (max risk case) the Action 1 becomes the riskiest action at the minimum risk point ($\xi = 0$). Action 3 once again takes the riskiest action position as $\xi$ reaches 1 for the maximum risk case. This patter is a result of the combined Return resulting from multiple state results from Action 3 and the impacts of $\xi$ on the relative Return mean-variance.

State 7 tests another condition resulting from the relative Return mean-variance shifting as the risk-tolerance in varied. State 7 is another two action state with a lower-mean, lower-variance action and higher-mean, higher variance action (Figure 7.27). The resulting Risk-Tolerance Sensitivity Profile (Figure 7.26) shows the same action (Action 1) preferred at the lowest and highest risk-tolerance levels. This is a result of the shifting relative Return mean-variance as the risk-tolerance is varied (Figure 7.28).

State 6 and State 7 have Risk-Tolerance Sensitivities Profiles that yield results not directly aligned to the $s-a-s'$ Rewards. The profiles are a result of the long term Return and not just the immediate Reward. The change difference can be attributed to the change in decisions made after the specified decisions point. Return accounts for future decisions

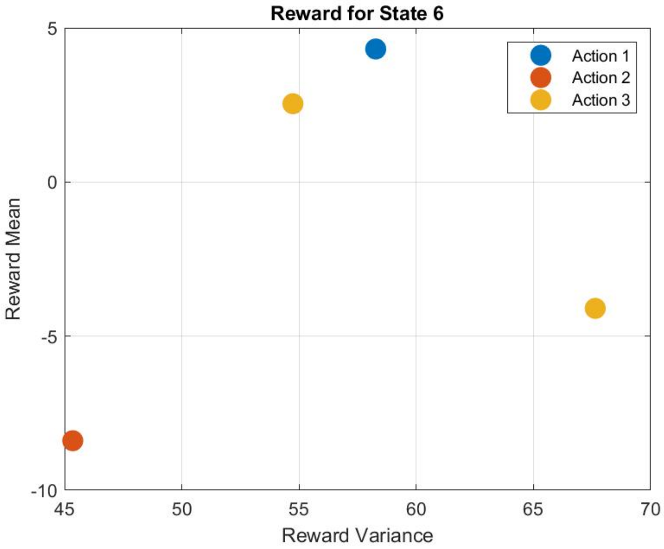


Figure 7.24: Experiment Set 1a Case 3 Scenario 1: Immediate Reward for State 6

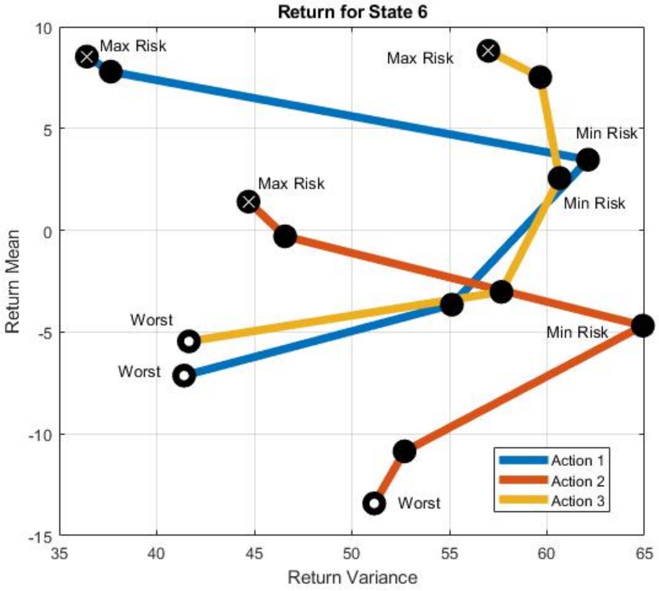


Figure 7.25: Experiment Set 1a Case 3 Scenario 1: Long Term Return for State 6

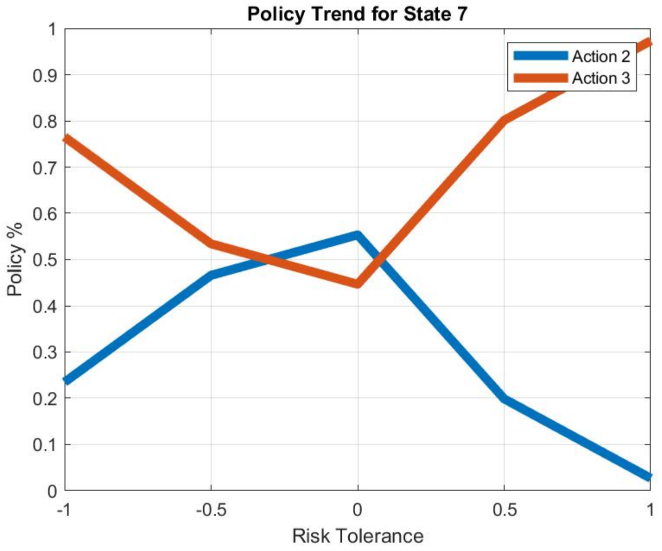


Figure 7.26: Experiment Set 1a Case 3 Scenario 1: RTSP for State 7

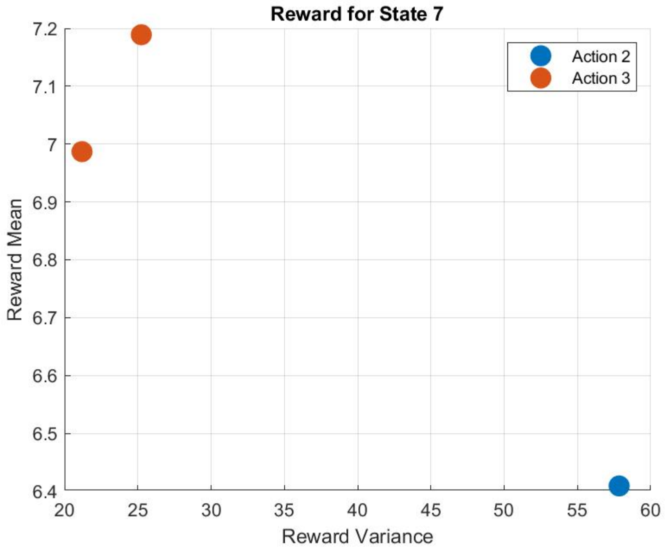


Figure 7.27: Experiment Set 1a Case 3 Scenario 1: Immediate Reward for State 7

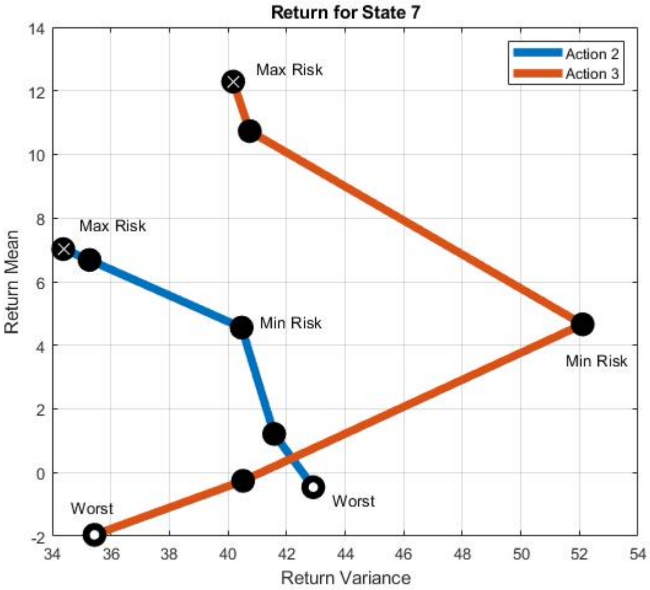


Figure 7.28: Experiment Set 1a Case 3 Scenario 1: Long Term Return for State 7

and results in deviations from the immediate Rewards of a given state and action.

**Scenario 2: Five Action Random MDP** Four decisions states were selected for analysis and are highlighted in Figure 7.29. The first state (State 1) represents a nominal three action case. The second (State 2) represents a four action case with a Pareto inefficient action. The third (State 3) represents a four action case without a Pareto inefficient action. The fourth (State 4) represents a five action scenario.

Three actions are available at State 1 with Action 1 resulting in three separate states (Figure 7.31). The immediate reward for all states resulting from Action 3 have a higher mean but less variance than the other Actions. Action 4 results in a moderate-mean but high-variance. The Risk-Tolerance Sensitivity Profile for State 1 (Figure 7.30) results in Action 2 preferred at the worst case, Action 4 peaking near the minimum risk case, and Action 1 preferred at both the minimum and maximum risk case. This would yield a determination that Action 4 is mildly Pareto inefficient but is a potential viable alternative for a minimum risk profile. The Return mean-variance plot shows Action 4 dominated by

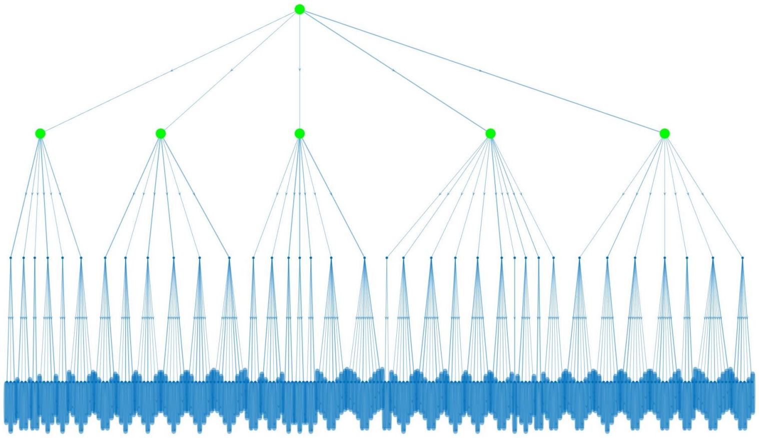


Figure 7.29: Experiment Set 1a Case 3 Scenario 2: MDP Graph and Highlighted Selected States of Interest

Action 1 and Action 2 at each risk-tolerance level (7.32). Just above the minimum risk point ($\xi$ just above 0), the Return means from Action 1 and Action 4 are near equal. The offset in variance at that point results in a slight preference to Action 1 just above the minimum risk point in the Risk-Tolerance Sensitivity Profile.

The four action state (State 2) results align with those expected from Experiment 1a Case 1 and Case 2 regarding Pareto inefficient actions. Results built from the view point of the immediate state-action rewards (Figure 7.34) would yield the potential of Action 1 being dominated by Action 3, Action 4 being the worst action, and Action 5 peaking in preference just under the minimum risk point. The results seen in the Risk-Tolerance Sensitivity Profile for State 2 (Figure 7.33) result in a different preference profile. Action 3 is fully dominated and inefficient across all $\xi$. This is a result of the balance between resulting states ensuring that, given the uncertainty, the resulting mean and variance of the multi-$s'$ action will be dominated by the single-$s'$ actions. The relative dominance can been visualized in the Return mean-variance plot (Figure 7.35) where the relative mean-variance between Actions is relatively constant and Action 3 has a moderate-mean and high-variance.

State 3 with no Pareto inefficient actions can be juxtaposed against State 2. Each action

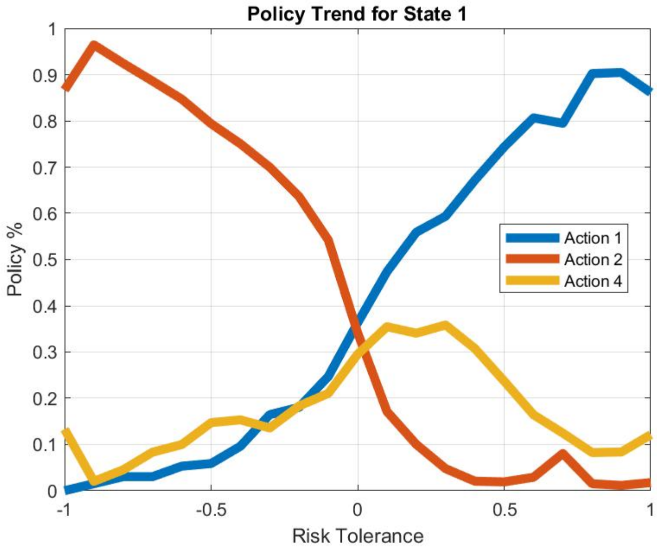


Figure 7.30: Experiment Set 1a Case 3 Scenario 2: RTSP for State 1

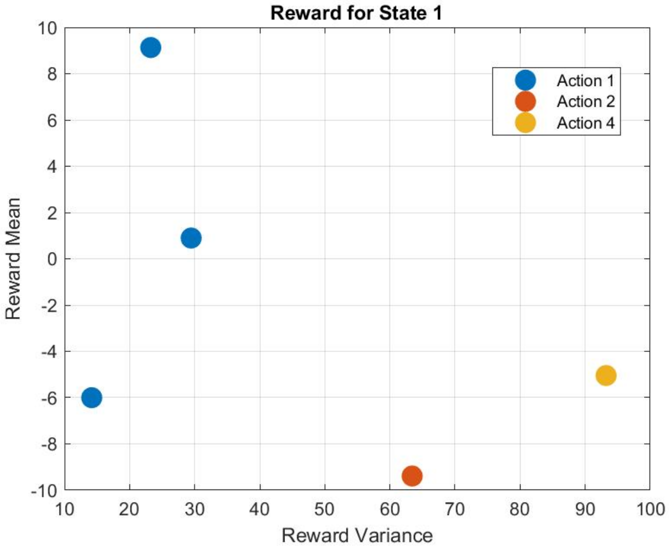


Figure 7.31: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 1

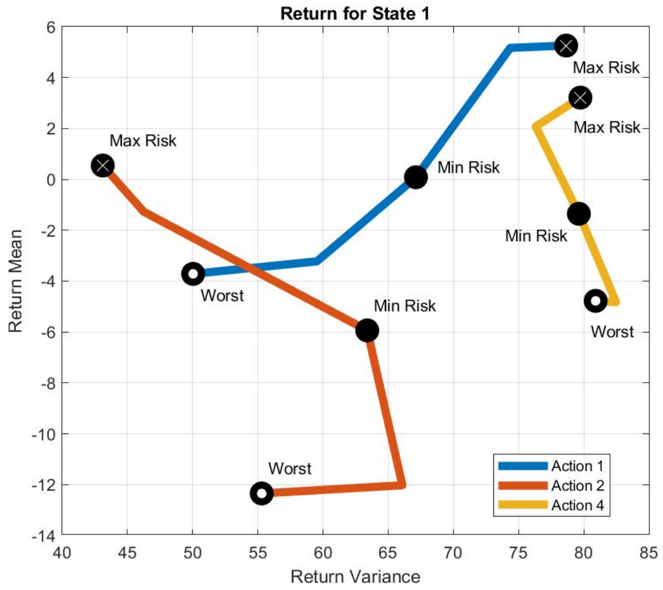


Figure 7.32: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 1

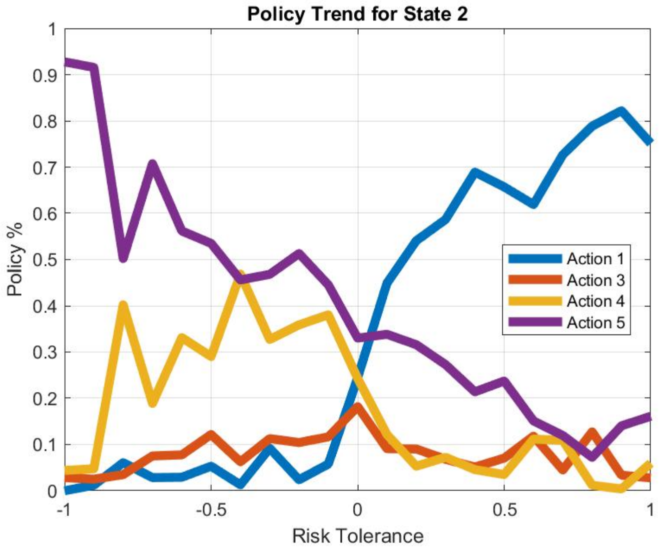


Figure 7.33: Experiment Set 1a Case 3 Scenario 2: RTSP for State 2

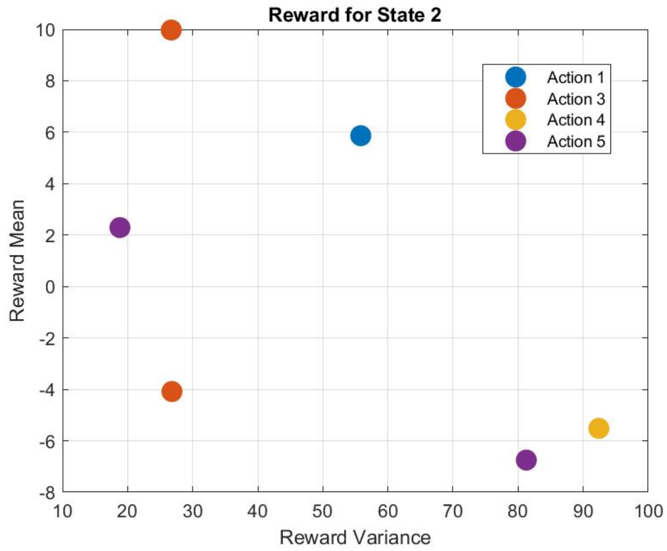


Figure 7.34: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 2

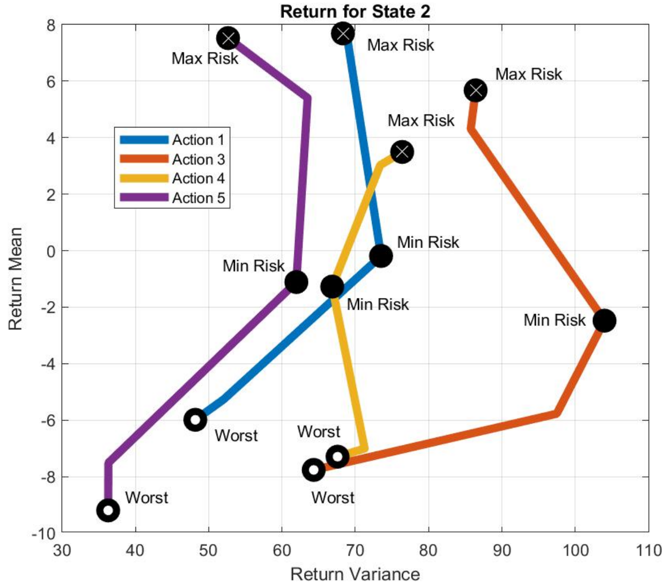


Figure 7.35: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 2

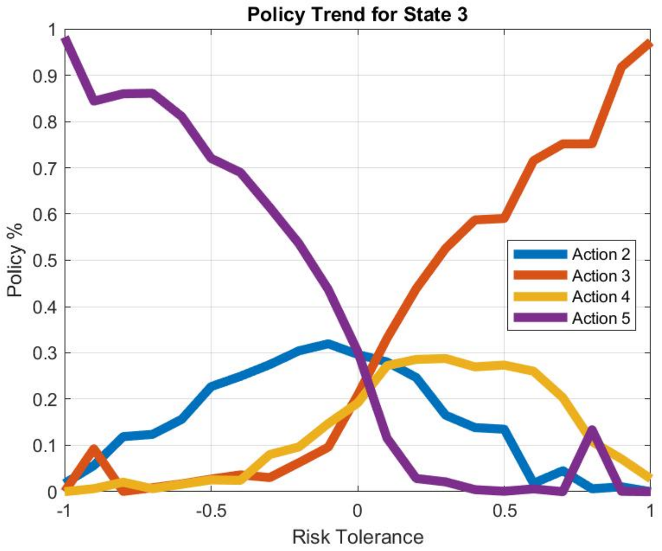


Figure 7.36: Experiment Set 1a Case 3 Scenario 2: RTSP for State 3

reward is grouped relatively close (Figure 7.37) and provides a gradual mean and variance trend. The Risk-Tolerance Sensitivity Profile shows the gradual policy trend from Action 5 (worst) to Action 2 or 4 peaking (minimum risk) to Action 3 (maximum risk). This is in line with the results of a Pareto efficient action scenario from Experiment Set 1a Case 2. The Return mean-variance (Figure 7.36) supports the above trend and shows that each action, as it peaks in policy contribution as a function of $\xi$, is not fully dominated by other actions. Note that Action 3 and Action 4 are near equal in mean and variance between minimum and maximum risk points. This yields the peak in policy contribution by Action 4 just above the minimum risk that is near the contribution of Action 3.

The selected five action state (State 4 and 5) provides one more step in complexity and begins to pus the boundaries of useful intermediate metrics. There are a few unique aspects to the action rewards (Figure 7.40). Action 1 has a lower-mean but split variance. Action 4 has a lower variance but a split mean. The resulting Risk-Tolerance Sensitivity Profile (Figure 7.39) shows that Actions 2 and 5 are preferred under high risk-tolerant conditions with Actions 1, 3, & 4 preferred at negative risk tolerance levels. It appears that Action 1 dominates Actions 3 and 4 as the risk-tolerance approaches the worst case. The Return

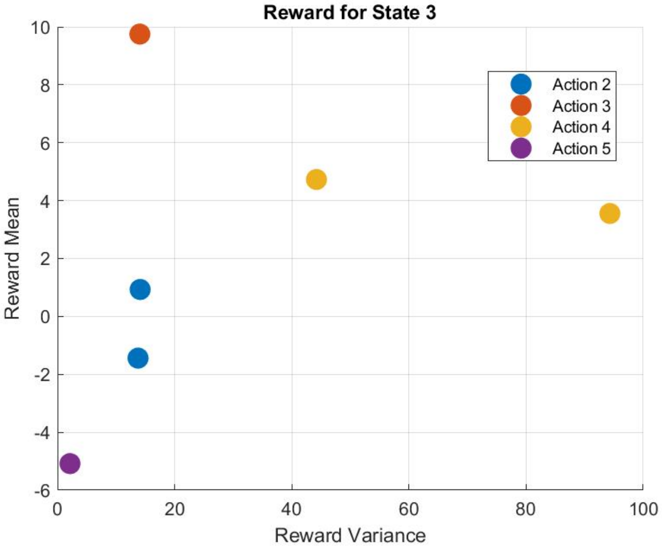


Figure 7.37: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 3

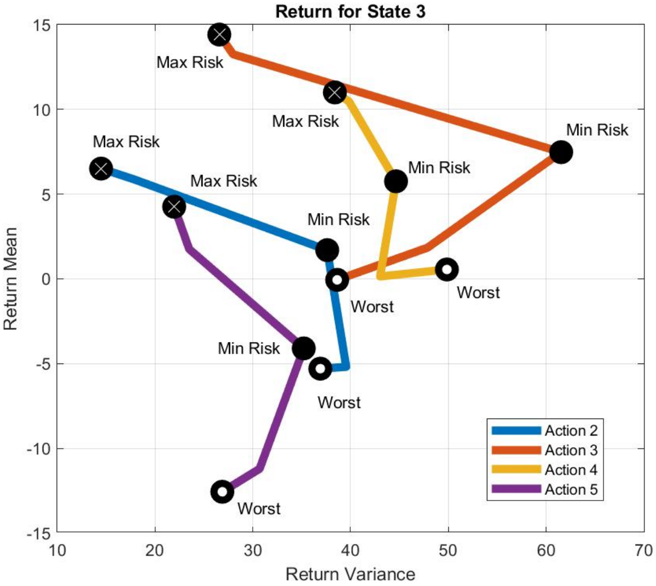


Figure 7.38: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 3

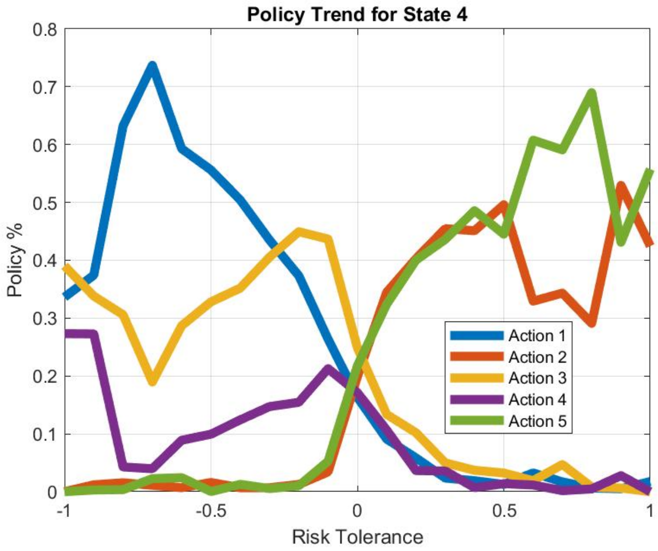


Figure 7.39: Experiment Set 1a Case 3 Scenario 2: RTSP for State 4

mean-variance supports the trend (Figure 7.41) where Actions 2 and 5 continually are the highest risk options across all $\xi$. Action 1 dominates as the worst options across all $\xi$ with Action 3 dominating at the minimum risk point across all $\xi$. Action 4 is continually dominated by Action 1 across all $\xi$ for the worst action though the mean-variance trends are closely aligned.

The second five action state (State 5) has multiple actions with mare than a single resulting state (Figure 7.43). The Risk-Tolerance Sensitivity Profile (Figure 7.42) shows the dominance of Action 3 at high risk-tolerances over Action 5. At negative risk-tolerance Action 2 dominates Actions 1 and 4. This is directly in line with the measured Return mean and variance (Figure 7.44). There is a clear bifurcation between Actions 3 & 5 and Actions 1, 2, & 4. Action 3 maintains a variance advantage relative to Action 5 over all $\xi$.

The final state of interest (State 6) has only three actions but Action 3 results in three states with a highly divers mean-variance Reward for each (Figure 7.46). The mean and variance of the Returns as a function of risk-tolerance (Figure 7.47) show Action 3 consistently with the lowest-mean and highest-variance. This makes it consistently the worst action. This is present in the Risk-Tolerance Sensitivity Profile for State 6 (Figure 7.45).

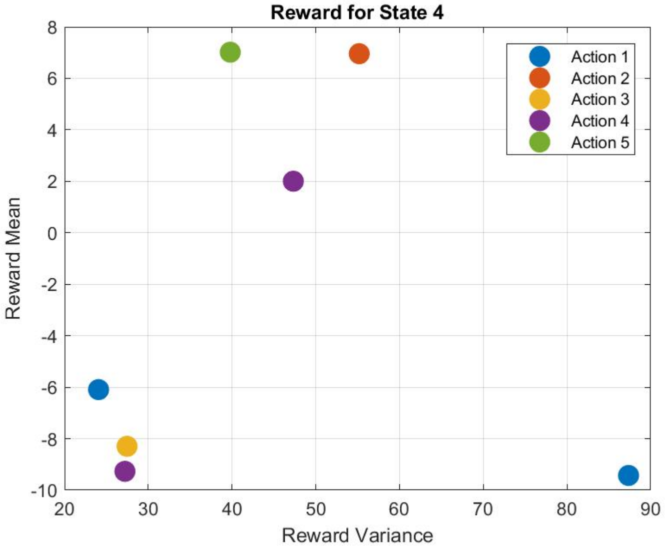


Figure 7.40: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 4

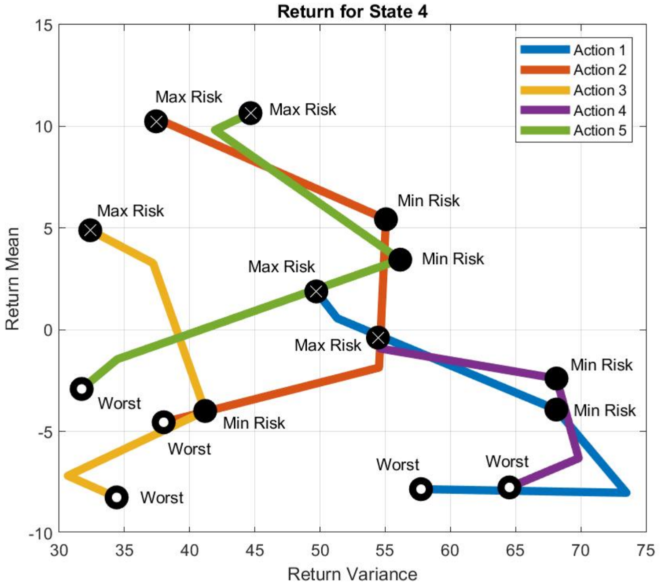


Figure 7.41: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 4

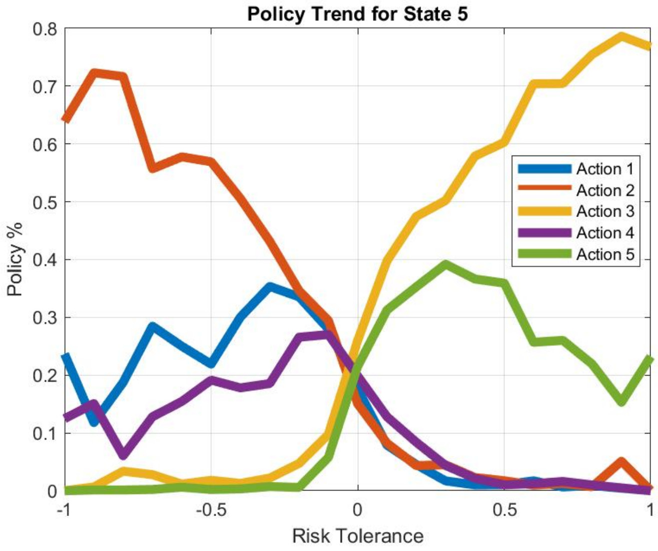


Figure 7.42: Experiment Set 1a Case 3 Scenario 2: RTSP for State 5

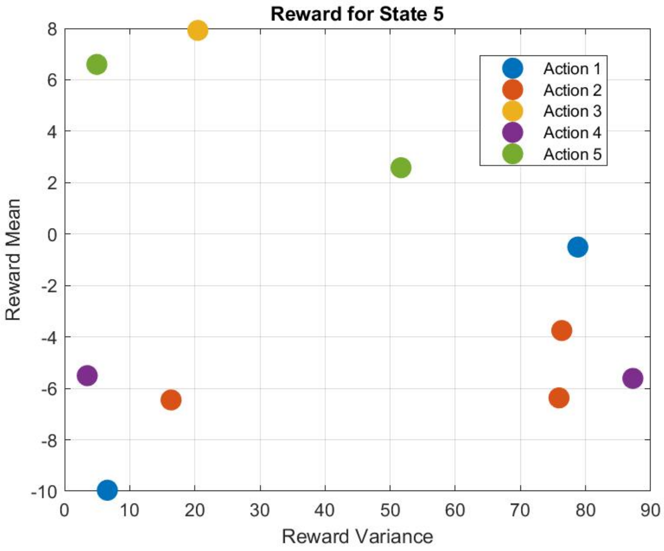


Figure 7.43: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 5

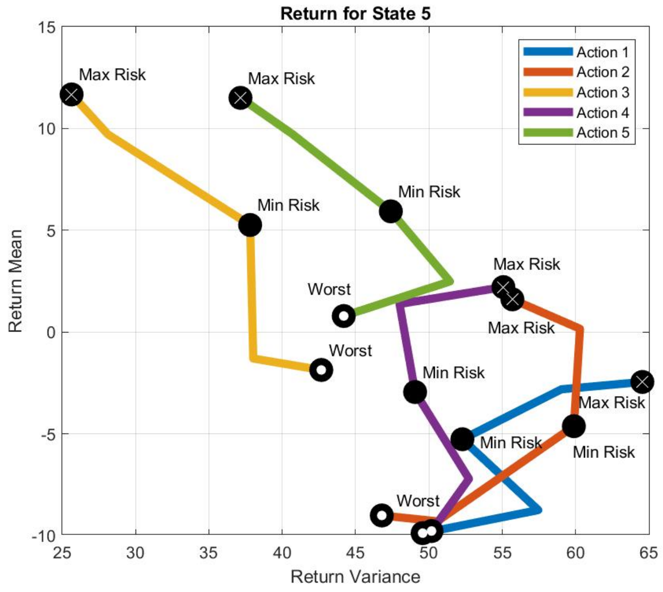


Figure 7.44: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 5

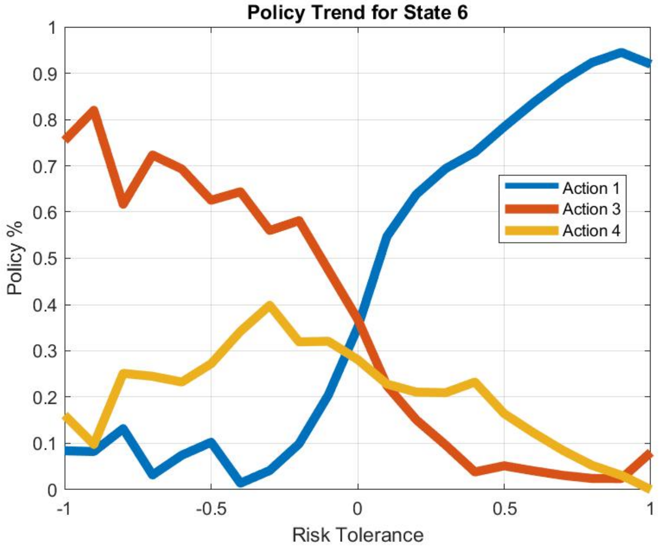


Figure 7.45: Experiment Set 1a Case 3 Scenario 2: RTSP for State 6

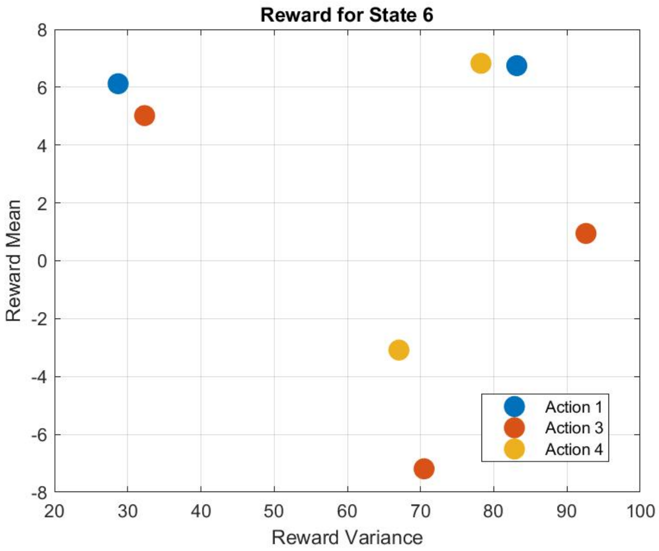


Figure 7.46: Experiment Set 1a Case 3 Scenario 2: Immediate Reward for State 6

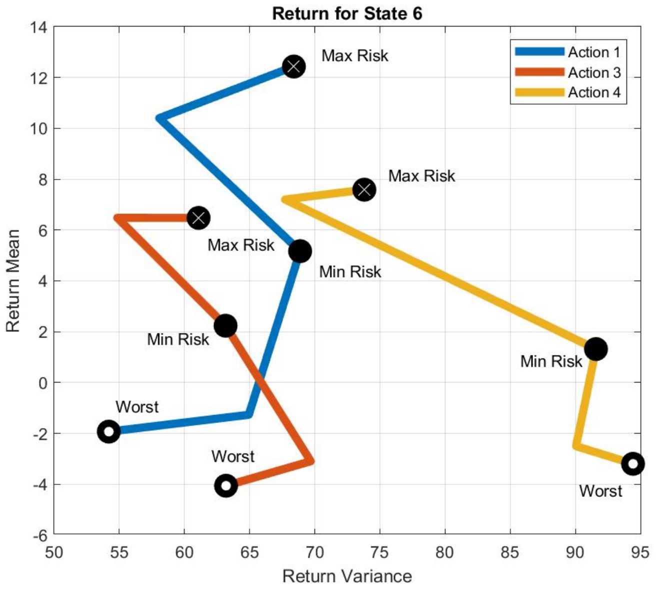


Figure 7.47: Experiment Set 1a Case 3 Scenario 2: Long Term Return for State 6

### 7.1.2 Experiment Set 1b: Sequential Decision Making

Experiment Set 1b defines scenarios and samples it's representation from a full Truth Model. The samples are then used to generate a full MDP on which to apply the risk-based policy algorithm.

*Experiment Set 1b Case 1: Repeated Pareto Efficient Actions*

The experimental setup for Case 1 is depicted in Chapter 6 which describes the Truth Model setup. Included in Case 1 is the sampling of the Truth Model. Sampling metrics as a function of sample size are depicted in Figure 7.48. Three metrics are tracked to identify state-action space sampling health. The number of unique states and actions track the growth of the decision space. The number of states and actions will plateaus as all state-action pairs are discovered and episodes begin to only sample already discovered states and actions. Ten thousand episodes were run to populate the decision space for Case 1 with the small action space fully sampled and the state space near fully sampled.

The sampled $s - a - s'$ reward for all samples shows the gradual fan out in performance uncertainty (Figure 7.49). The Probability Density Function (PDF) of reward at each time step gives a different view, Figure 7.50. In this simple scenario, peaks at near similar states can be seen at $t = 5$ with a gradual fan out. Additionally, the total sample size of each bin decreases as the time steps increase. The distribution mean and 3-$\sigma$ are displayed for each time step.

The resulting MDP can be visualized in the structure graph depicting state to state transitions, Figure 7.51. The small actions space allows it to be fully visualized (Figure 7.52). At any given state with actions, three actions should be present for Case 1. It should be noted that there are cases where previous actions and the resulting timelines do not allow the selection of one acquisition action. Actions 2 through 4 can be seen as options at the initialization state ($t = 0$). Action 1, the wait action which is present due to resources being allocated based on previous actions, is first present after the initial state as systems are now

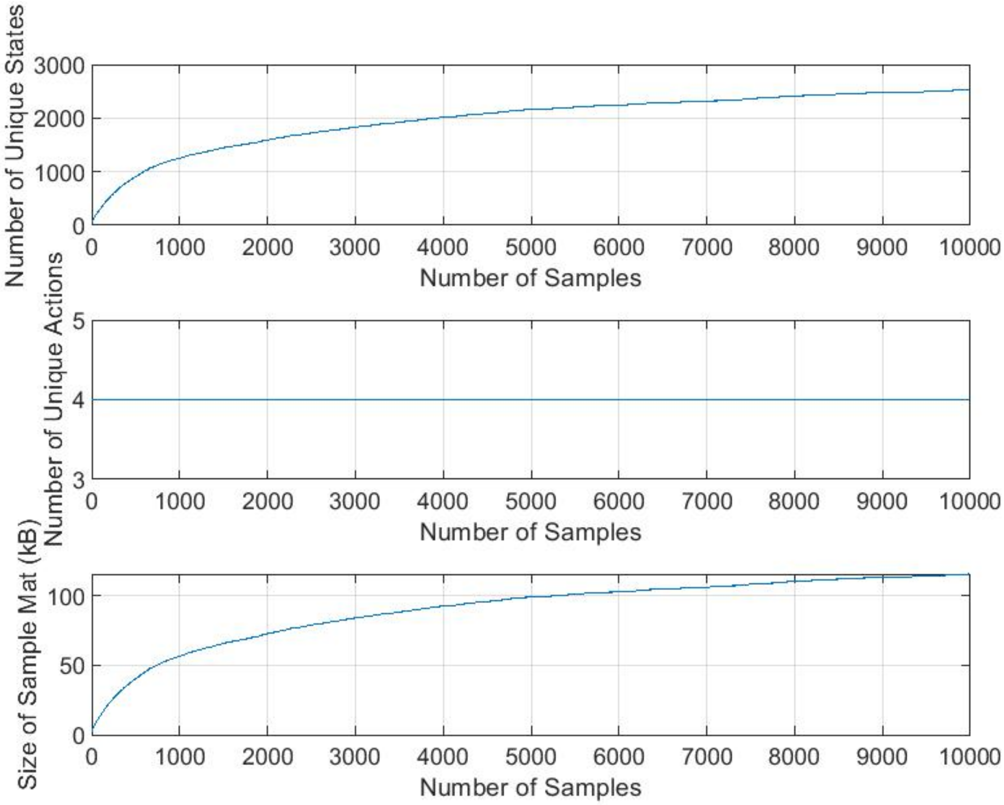


Figure 7.48: Experiment Set 1b Case 1: Episode Sample Metrics

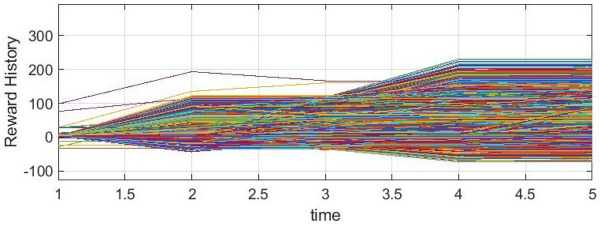


Figure 7.49: Experiment Set 1b Case 1 Baseline: Sampled Reward versus Time

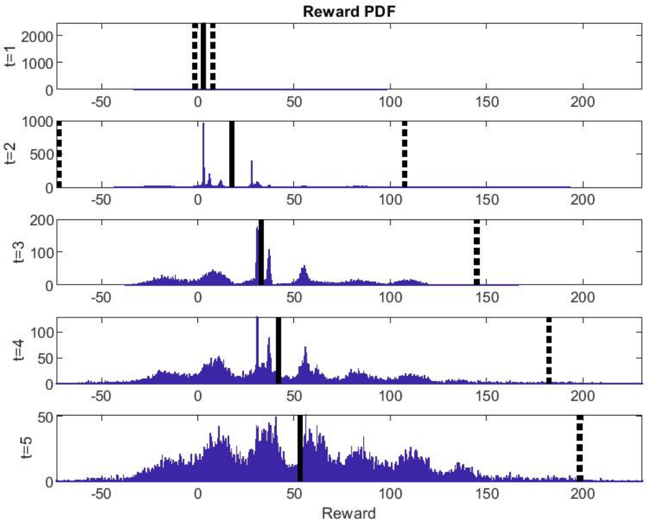


Figure 7.50: Experiment Set 1b Case 1 Baseline: Sampled Reward PDFs

being developed. Given the uncertainty in development, acquisition actions are available at $t = 1$ though only in a few resulting state where the acquisition time was less than or equal to a single time step.

The resulting Risk-Tolerance Sensitivity Profile for State 1 (Figure 7.53) generated based on the above MDP should resemble the three action Pareto efficient action scenario from Experiment 1a Case 1. The preferences follow the implied Pareto action frontier despite the increase in complexity (multi-step impact of actions and temporal uncertainty). The addition of a system with mild Pareto inefficient capability (Figure 7.54, it an be seen that the mild inefficient action is mildly dominated across $\xi$. A significant increase in inefficiency of the system results in the acquisition option of the system to be fully dominated. The policy preference for the acquisition of the inefficient system nears zero for all $\xi$ (Figure 7.55).

### *Experiment Set 1b Case 2: Acquire vs. Develop Scenario*

Case 2 introduces varying decision spaces across states. Case 1 used specifically defined set of acquisition options that were available at every decisions point. Case 2 allows the

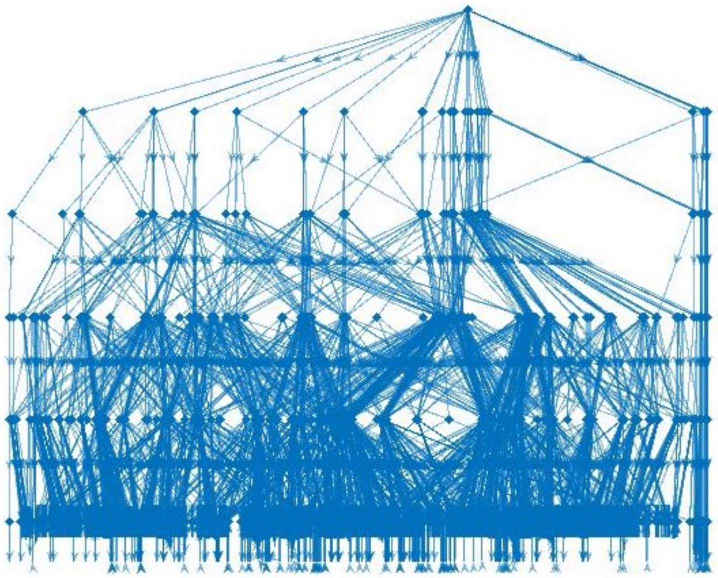


Figure 7.51: Experiment Set 1b Case 1: Full MDP Graph

simulation to play out based on initial conditions with out pre-orchestrating the decision space. States can be grouped by their available actions. These groupings represent specific decision spaces that exist across a subset of states. Each state will have it's own Risk-Tolerance Sensitivity Profile and each decision space will have states with the same actions available. This creates a portfolio of Risk-Tolerance Sensitivity Profiles by all states. Two selected decision spaces are highlighted for further investigation. The first is the Acquire System 1 or Develop System 2 (AS1vDS2) decision space. The second is Acquire System 1 or Acquire System 2 or Develop System 3 (AS1vAS2vDS3).

The AS1vDS2 decision space is one of the first that is encountered. The specific states are highlighted in Figure 7.56. States include the initial state and subsequent states where the Acquire System 1 was selected. Selected Risk-Tolerance Sensitivity profiles from the AS1vDS2 decision space show a common trend in decision preference (Figure 7.57). A preference can be seen for Acquiring System 1 as risk-tolerance increases.

The second decision space is realized once System 2 has been developed. The second decision space states (Figure 7.58) have three actions available to the Stakeholder. Selected

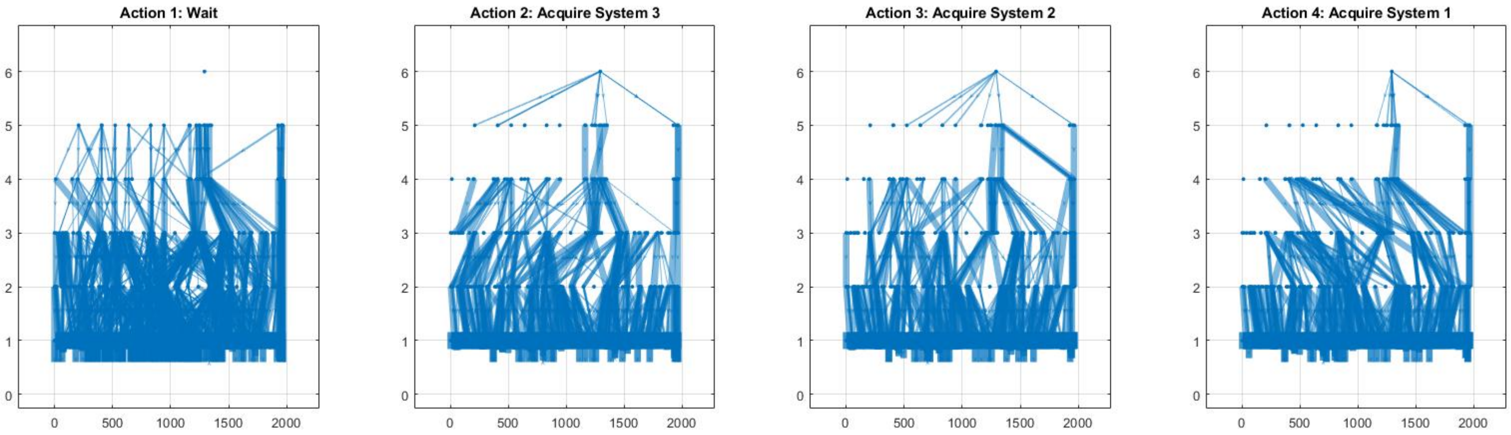


Figure 7.52: Experiment Set 1b Case 1: Action Graph

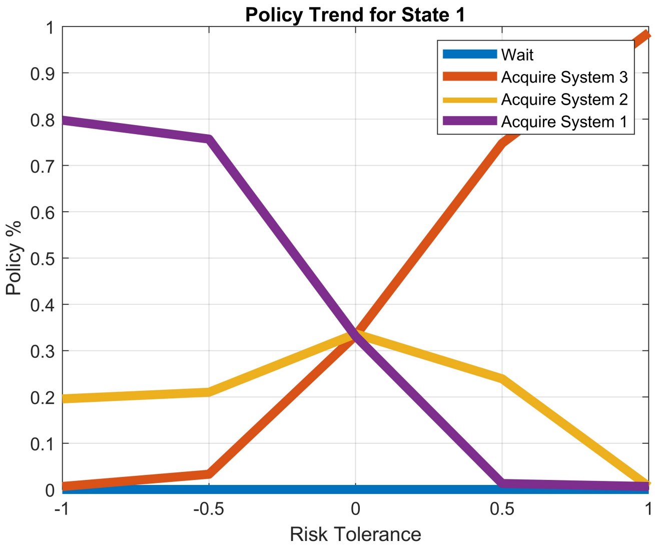


Figure 7.53: Experiment Set 1b Case 1: Baseline RTSP

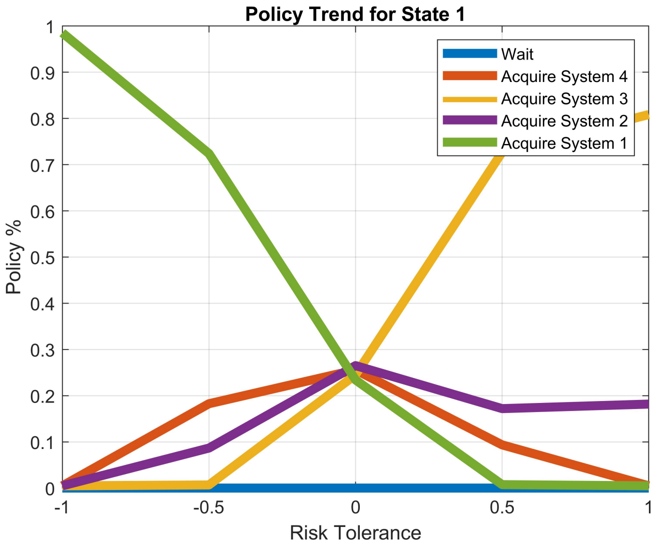


Figure 7.54: Experiment Set 1b Case 1 Mild Inefficiency: Risk-Tolerance Sensitivity Profile

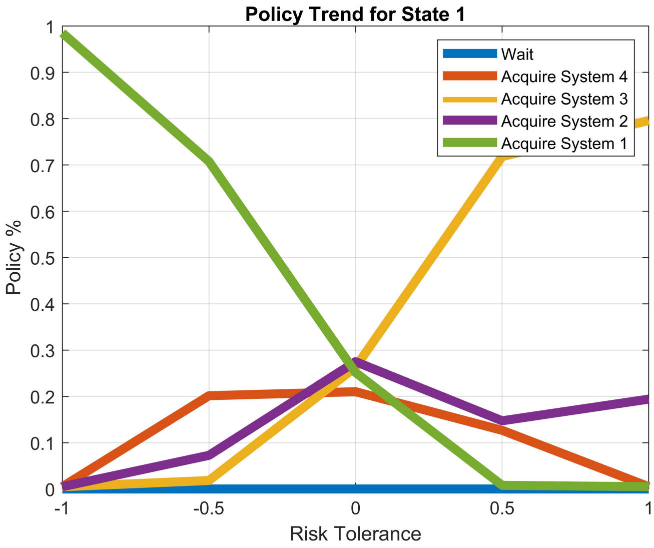


Figure 7.55: Experiment Set 1b Case 1: Significant Inefficiency RTSP

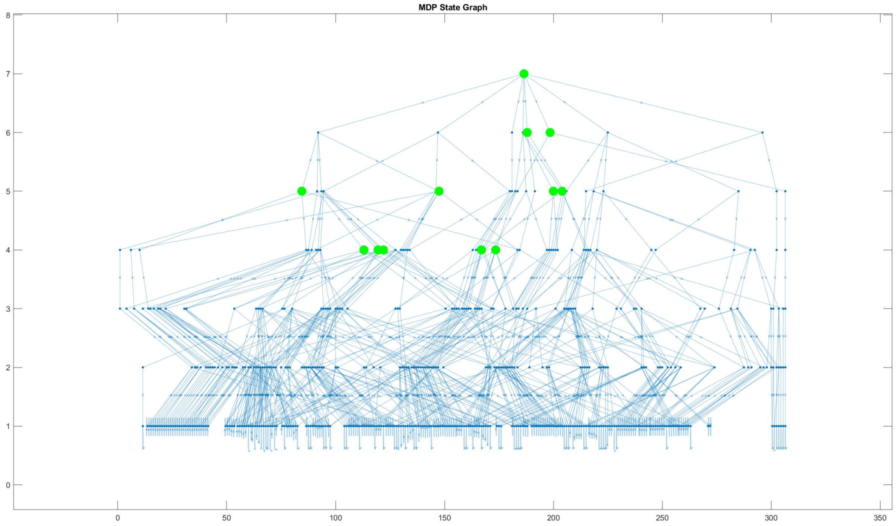


Figure 7.56: Experiment Set 1b Case 2: Decision Space States for Acquire System 1 or Develop System 2 States

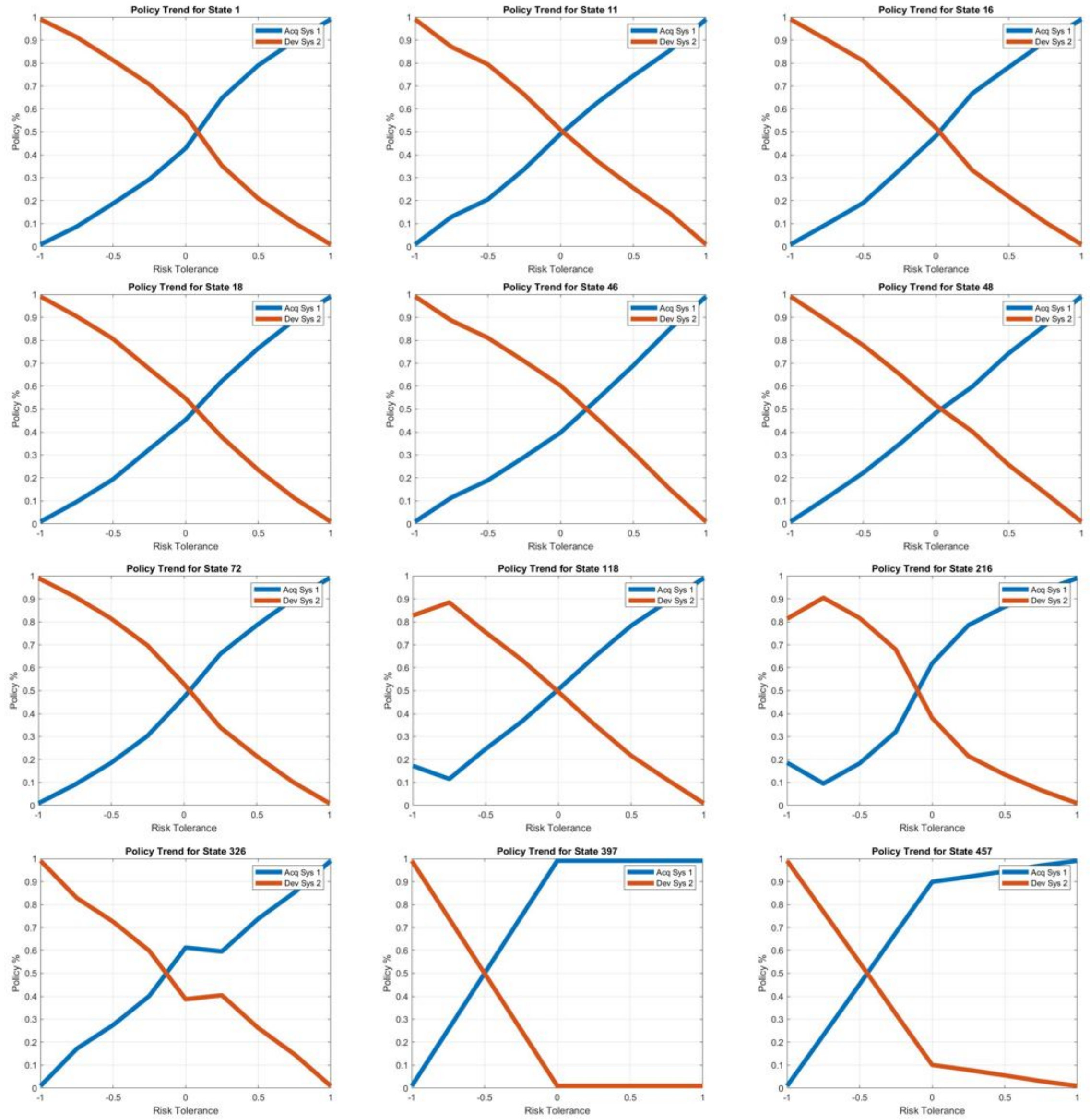


Figure 7.57: Experiment Set 1b Case 2: Decision Space States for Acquire System 1 or Develop System 2 RTSPs

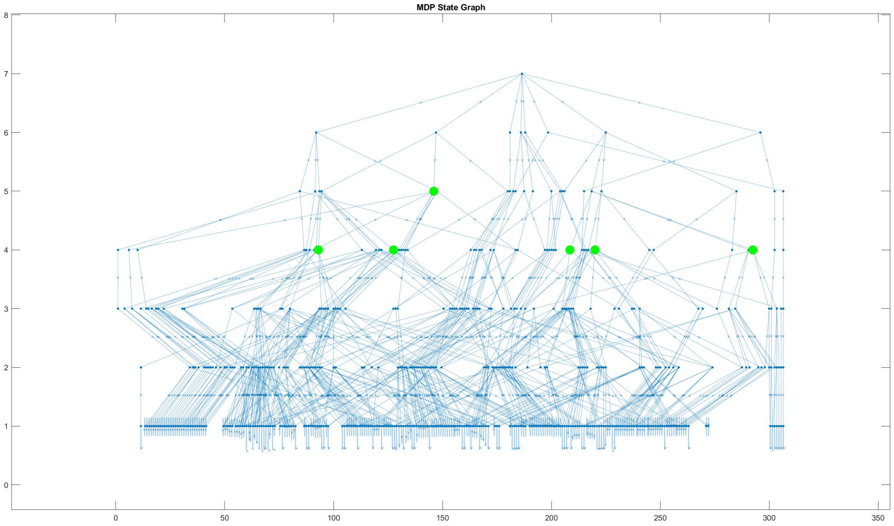
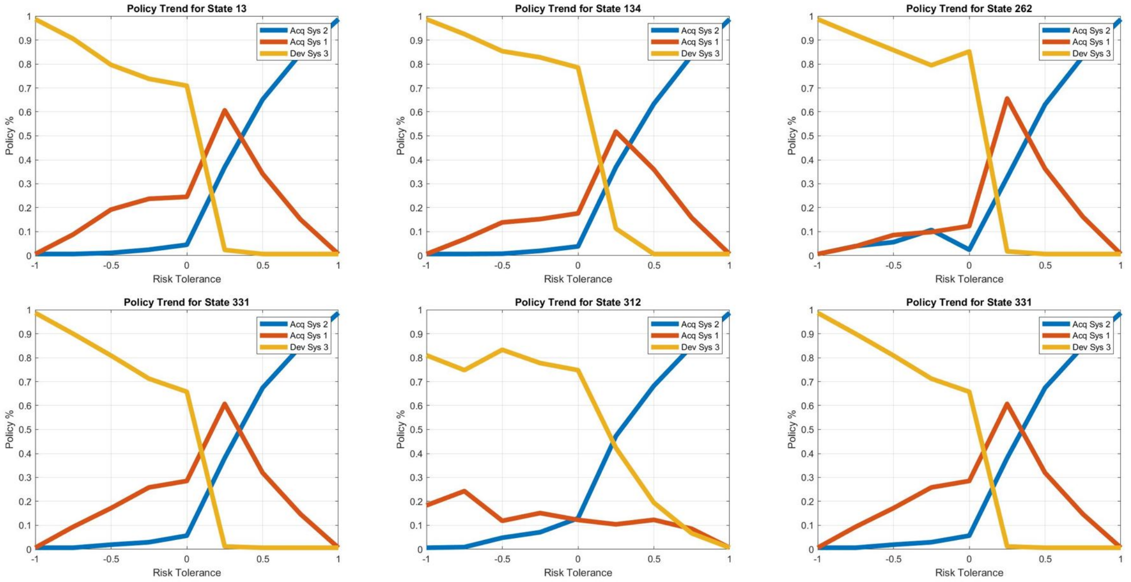


Figure 7.58: Decision Space Acquire System 1 or Acquire System 2 or Develop System 3 States

state Risk-Tolerance Sensitivity Profiles show a consistent pattern of preference as a function of risk-tolerance (Figure 7.59). Acquisition of System 2, a high-mean high-variance capability system, is preferred at the maximum risk point. The lower-mean lower-variance option, Acquiring System 1, is preferred at the minimum risk point. Development of System 3 is the worst option. Horizon timeline is important and the development of System 3, even if selected at it's earliest time, does not yield a fully acquired and ready for allocation System 3 within the considered time horizon. In short, the Return due to developing System 3 is not accounted for due to the time horizon constraint.

The second variant under consideration (Scenario 2) is artificially increasing the mean capability of System 2 (Figure 7.60) as compared to the nominal case above (Figure 7.61). The expectation is for Developing System 2 to begin to be preferred at the initial state. The difference between the nominal and inflated System 2 capability scenarios State 1 Risk-Tolerance Sensitivity Profiles (Figure 7.62) show the impact. Development of System 2 is preferred at almost all risk-tolerance levels sans the minimum risk point.

Once again, an examination of the relative Return as a function of $\xi$ provides insight.


Policy Trend for State 13

Policy Trend for State 134

Policy Trend for State 262

Policy Trend for State 331

Policy Trend for State 312

Policy Trend for State 331

Figure 7.59: Decision Space Acquire System 1 or Acquire System 2 or Develop System 3 Risk-Tolerance Sensitivity Profiles

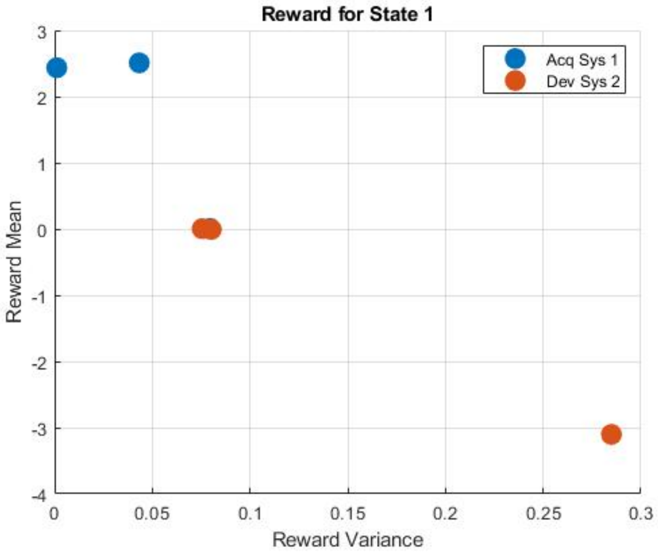

Reward for State 1

Figure 7.60: State One Immediate Action-State Reward for Higher System 2 Performance

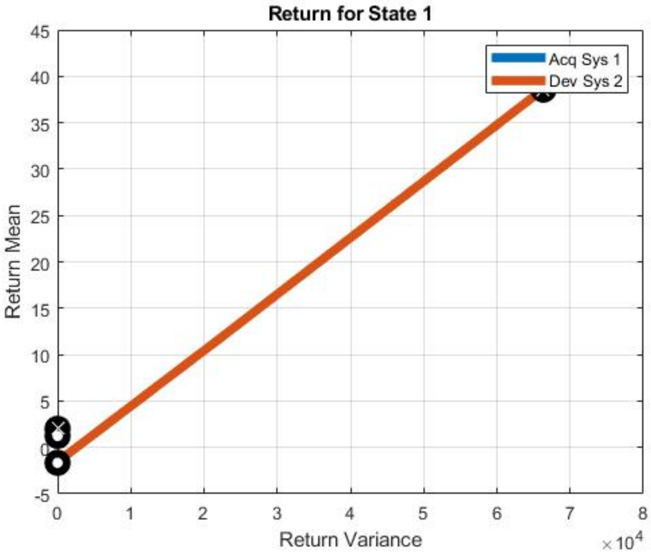


Figure 7.61: State One Risk-Tolerant Action Return Profile for Higher System 2 Performance

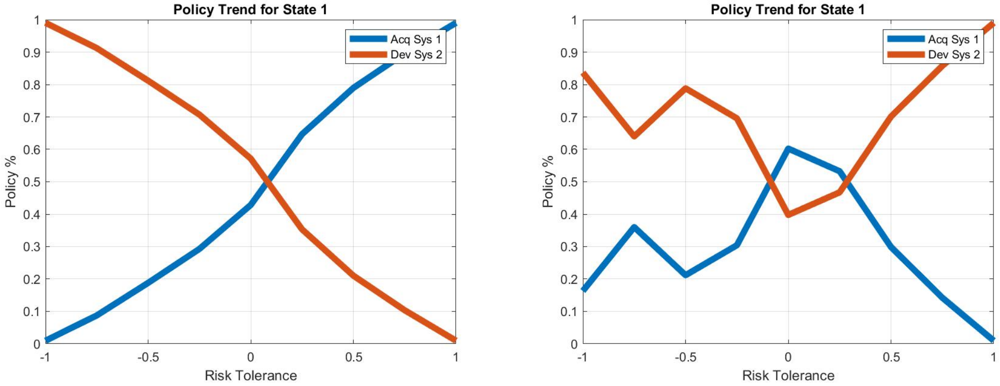


Figure 7.62: Low (left) Versus High (right) System 2 Capability Risk-Tolerance Sensitivity Profiles

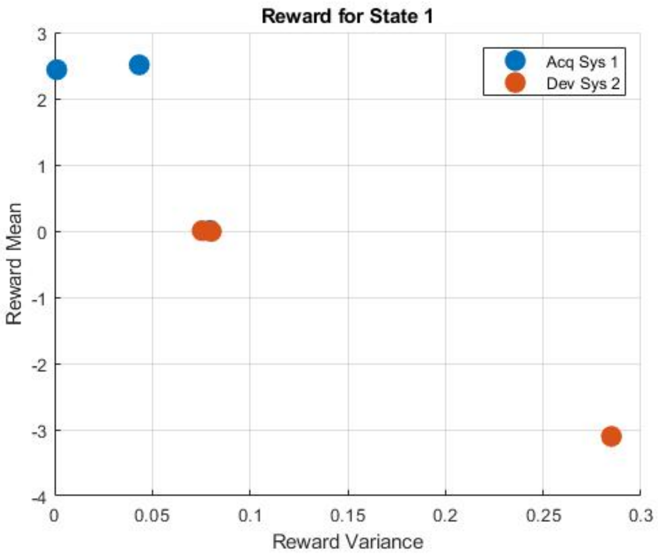


Figure 7.63: State One Immediate Action-State Reward

The nominal Return mean-variance plot (Figure 7.63) shows little variation in absolute or relative Return mean and variance as a function of risk-tolerance. In the inflated case (Figure 7.64) there is a swapping of which action has the lowest-mean and highest-variance. This swap is caused by selecting Develop System 2 and no capitalizing on it's development in the worse case risk-based policy case ($\xi = 0$). Decisions past the initial state have a direct impact on future Rewards and the initial state action Returns. If Developing System 2 takes time to develop and it's never capitalized on, the action then becomes the worse case option. Acquiring System 2 is preferred in later states as the risk-tolerance is increased. This shift lets the high mean capability of System 2 to be realized in the Return of the initial state.

*Experiment Set 1b Case 3: Multi-Mission Acquire vs. Develop*

Experiment Set 1b Case 3 introduces the new complexity of resource constraints, system refresh, increased stakeholders, and system allocations. The state and action space significantly increases with these complexities. The results are analyzed as a function of decision spaces or groupings of states with the same available actions to the stakeholder of interest. Each of the three stakeholders have a varied resource pool (e.g. budget) an are ana-

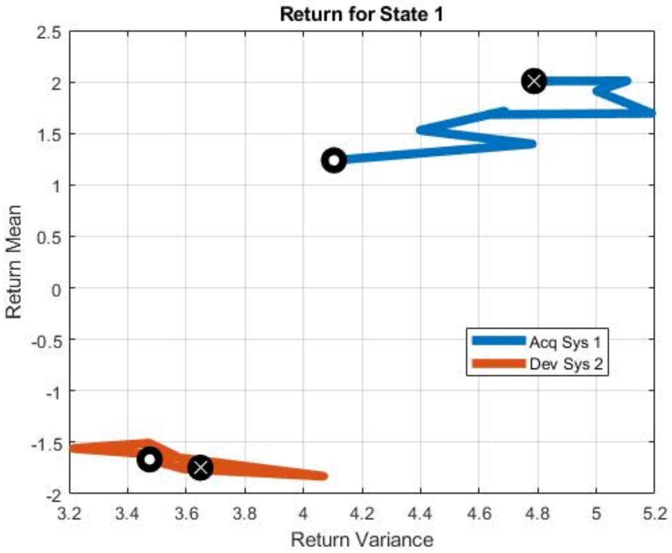


Figure 7.64: State One Risk-Tolerant Action Return Profile

lyzed relative to each other. Stakeholder 1 allows a look at allocation decisions of varying complexity and asset creation decisions. Stakeholder 2 allows the evaluation of increased allocation space. Finally, Stakeholder 3 enables the impact of no resource constraints.

**Allocation Only Decision Space** Resource constraints on Stakeholder 1 result in a limited decision space. At any given state, there is a single acquisition or development decision that can be feasibly selected. The limitation leads to allocation only decision spaces to dominate Stakeholder 1 decisions spaces. Three selected significant decision spaces allow the evaluation of the risk-based policy generation algorithm applied to a multi-mission problem.

The first decision space consists of four feasible actions defined by allocation (Table ds13ActionTable). Stakeholder 1 can allocate System 1 to either Mission 1 or Mission 2. Recall from the experiment setup described in Chapter 6, Stakeholder 1 prefers Mission 1 and System 1 provides more capability to Mission 1 than Mission 2. The Risk-Tolerance Sensitivity Profile portfolio for the decision space shows a consistent solution (Figure 7.65).

Action 7 represents System 1 applied to Mission 1. Action 7 is the preferred action as

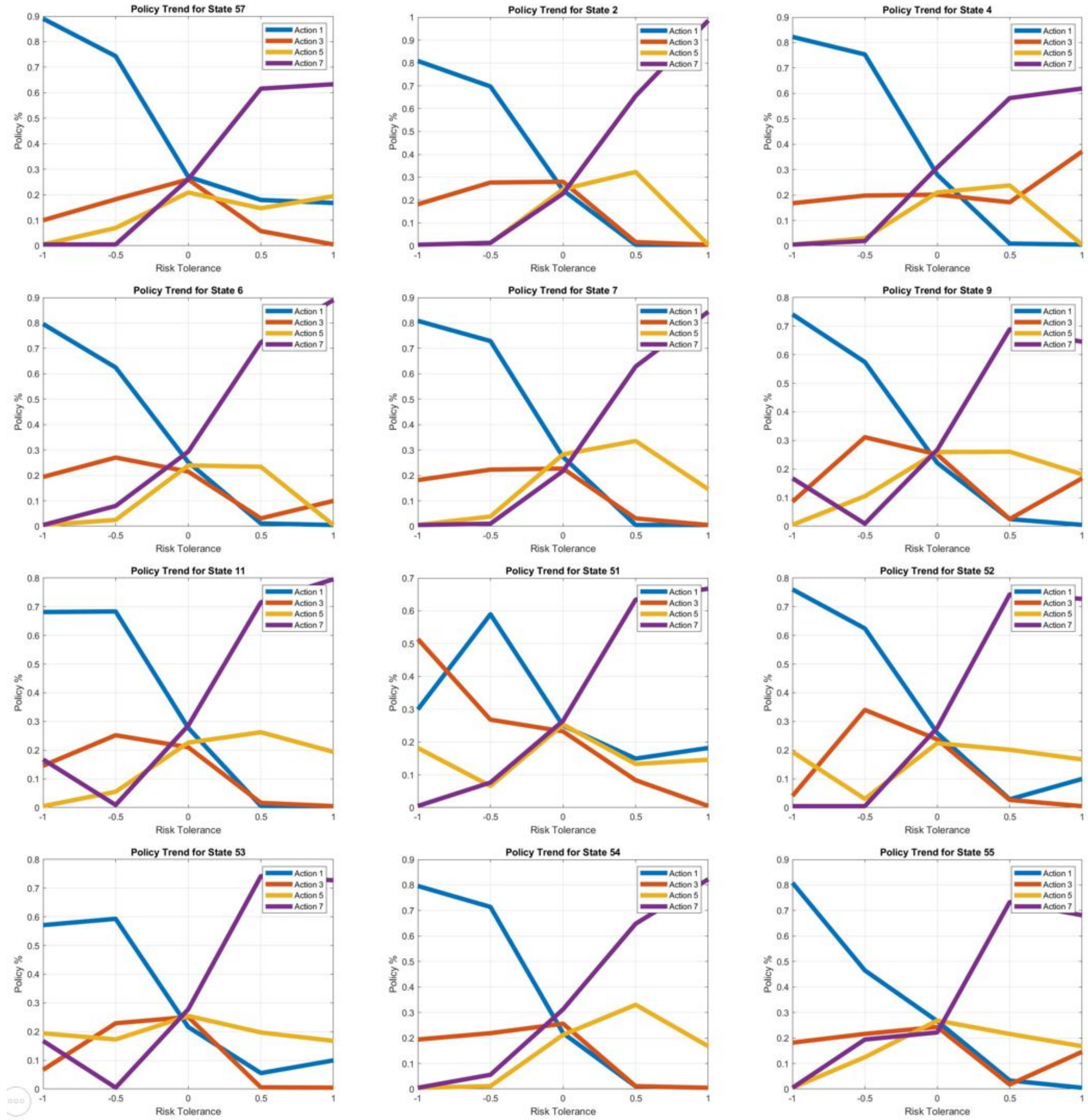


Figure 7.65: Allocation of Three Systems Only Selected Risk-Tolerance Sensitivity Profiles

Table 7.1: Allocation of Three Systems Only Decision Space

| **Action** | **$n$ System 1 to Mission 1** | **$n$ System 1 to Mission 2** |
|---|---|---|
| Action 1 | 0 | 3 |
| Action 3 | 1 | 2 |
| Action 5 | 2 | 1 |
| Action 7 | 3 | 0 |

the risk-tolerance level is increased and aligns with exceptions. Similarly, Action 1 is preferred at the worst risk-tolerance levels in accordance with expectations. The intermediate options (Action 3 and Action 5) behave as do moderate-mean lower-variance options due in Pareto efficient action spaces. This correlates to the moderate allocations for the two actions.

The second decision space is characterized by an increase in available systems from three to four (Table 7.6). Selected profiles from the Risk-Tolerance Sensitivity portfolio show consistency to trends across states with similar actions (Figure 7.66). The trends match a preference for allocating System 1 to Mission 1 at the maximum risk point with allocating System 1 to Mission 2 at the worst point. The now equal allocation (Action 6) shows a peak near the minimum risk point. Action 4 and Action 8 show a symmetry about the minimum risk point.

The third decision space is characterized by an increase in available systems and the consistent selection of Acquiring System 2 (Table 7.6). Selected profiles from the Risk-Tolerance Sensitivity portfolio show consistency to trends across states with similar actions (Figure 7.67). The trends match a preference for allocating System 1 to Mission 1 as

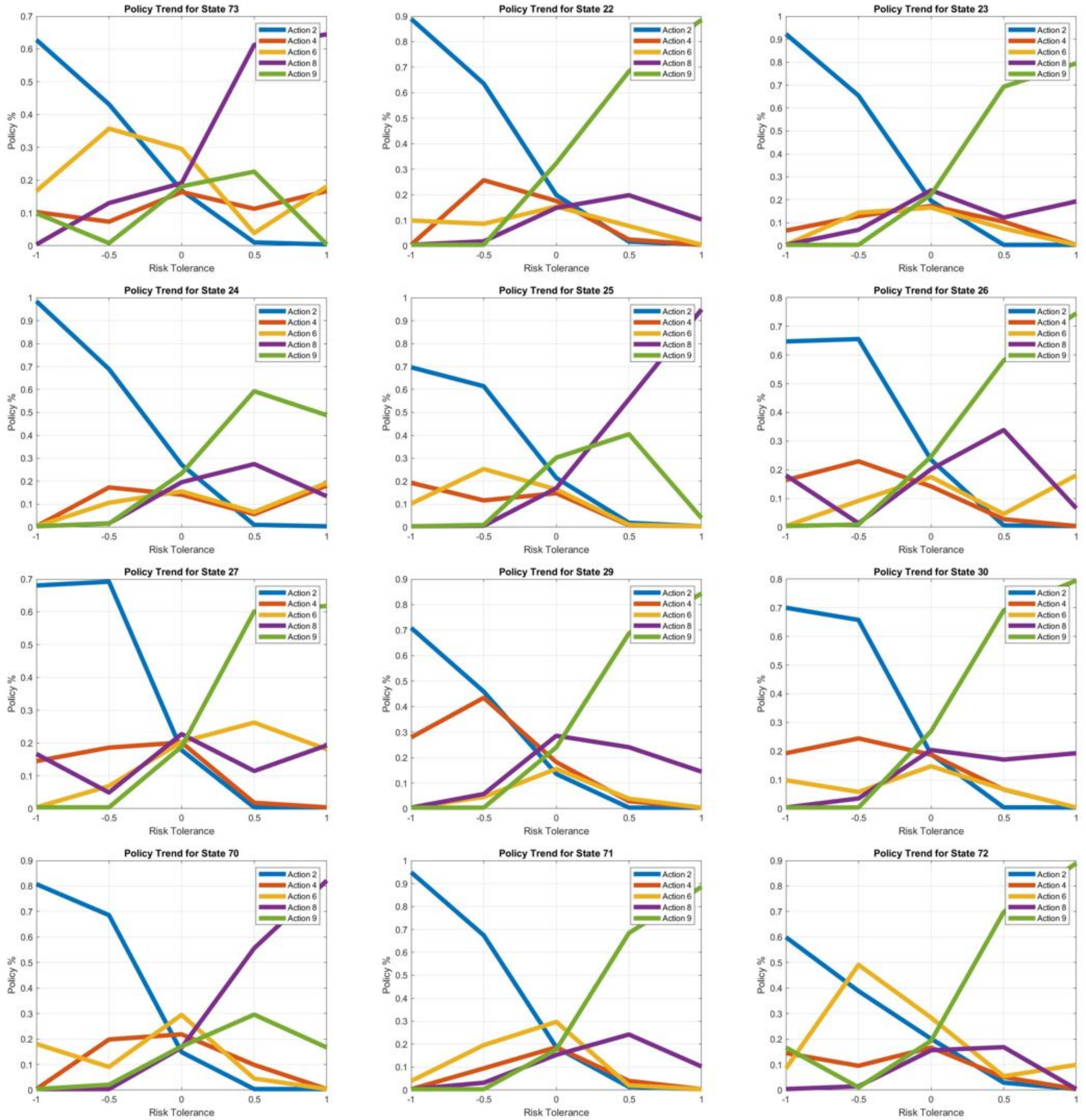


Figure 7.66: Allocation of Four Systems Selected RTSPs

Table 7.2: Allocation of Four Systems Decision Space

| **Action** | $n$ **System 1 to Mission 1** | $n$ **System 1 to Mission 2** |
|---|---|---|
| Action 2 | 0 | 4 |
| Action 4 | 1 | 3 |
| Action 6 | 2 | 2 |
| Action 8 | 3 | 1 |
| Action 9 | 4 | 0 |

anticipated. The now equal allocation (Action 17) shows a peak near the minimum risk. Action 14 and Action 20 show a symmetry about the minimum risk point as expected.

**Allocation and Asset Creation** Stakeholder 1 does have a decision space that includes both asset creation and asset allocation (Table 7.5). Asset creation includes acquisition and development of systems. The decision space represents the largest and most diverse examined thus far. There is little distinguishable trend upon initial examination of the profile portfolio (Figure 7.68).

The individual actions within the decision space can be grouped and the cumulative policy examined. The mission grouped actions represented in the selected policy portfolio (Figure 7.69) show clear trends in line with the allocation only decision space evaluations.

**Acquire New Asset or Refresh Old Asset** The balance of budget and system action opportunities provides an opportunity to evaluate an acquisition of a new asset versus refresh of an old asset (Table 7.5). When the decision space is split by allocation (Figure 7.70) a clear correlation between risk-tolerance and allocation is clear. Stakeholder 2 prefers Mission 2 allocations which aligns with Stakeholder 2's preferred mission.

Table 7.3: Allocation of Four Systems with Single Acquisition Decision Space

| **Action** | **Acquire System 2** | **$n$ System 1 to Mission 1** | **$n$ System 1 to Mission 2** |
|---|---|---|---|
| Action 11 | ✓ | 0 | 4 |
| Action 14 | ✓ | 1 | 3 |
| Action 17 | ✓ | 2 | 2 |
| Action 20 | ✓ | 3 | 1 |
| Action 22 | ✓ | 4 | 0 |

Table 7.4: Merging the Decision Space Decision Space

| **Action** | **Acquire System 1** | **Develop System 2** | **$n$ System 1 to Mission 1** | **$n$ System 1 to Mission 2** | **Mission Preference** |
|---|---|---|---|---|---|
| Action 12 | ✓ |  | 0 | 5 | M2 |
| Action 15 | ✓ |  | 1 | 4 | M2 |
| Action 18 | ✓ |  | 2 | 3 | None |
| Action 21 | ✓ |  | 3 | 2 | None |
| Action 23 | ✓ |  | 5 | 0 | M1 |
| Action 24 | ✓ |  | 1 | 4 | M2 |
| Action 25 |  | ✓ | 2 | 3 | None |
| Action 26 |  | ✓ | 4 | 1 | M1 |
| Action 27 |  | ✓ | 5 | 0 | M1 |
| Action 28 |  | ✓ | 0 | 5 | M2 |
| Action 29 |  | ✓ | 3 | 2 | None |
| Action 30 | ✓ |  | 4 | 1 | M1 |

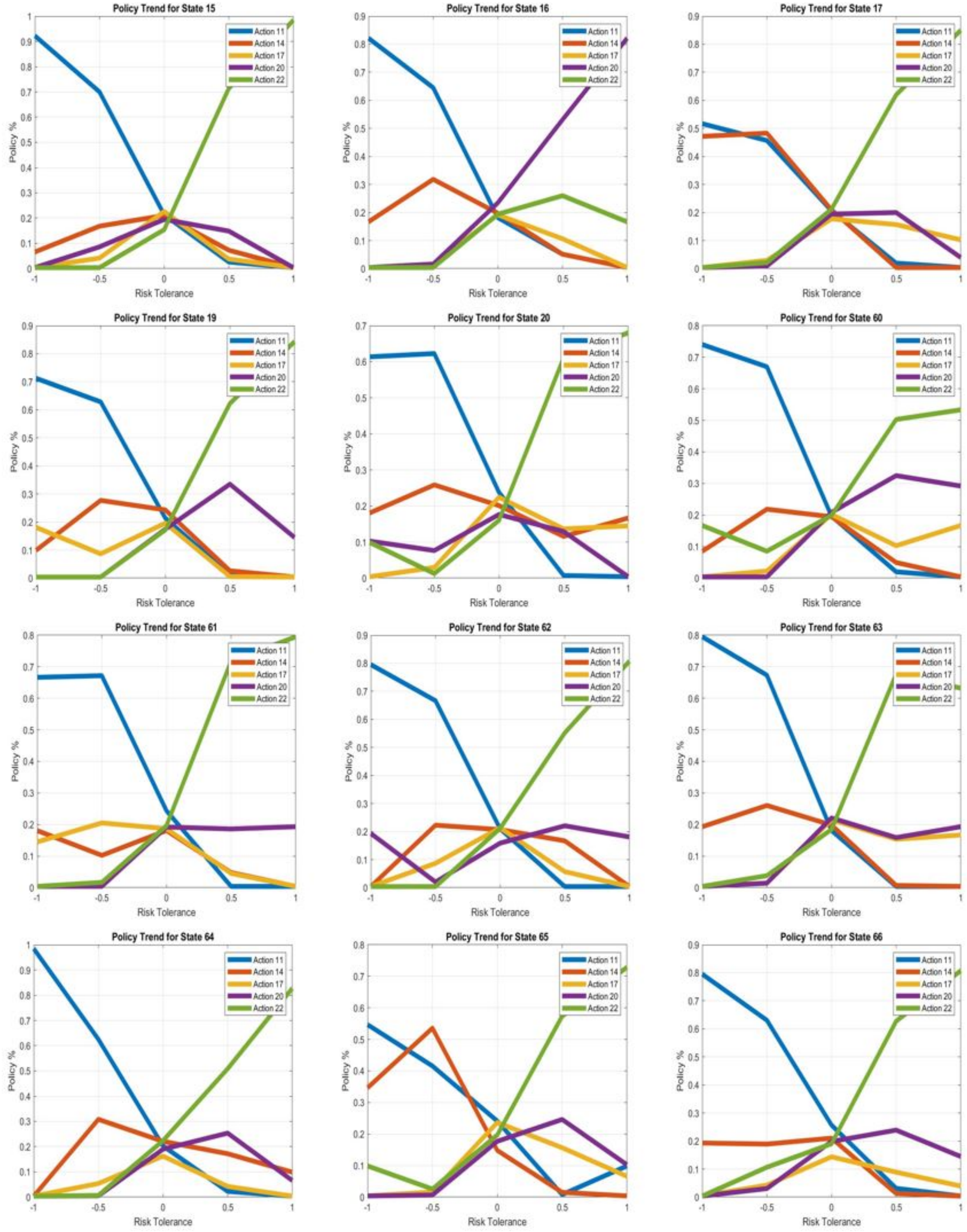


Figure 7.67: Allocation of Four Systems with Single Acquisition Selected RTSPs

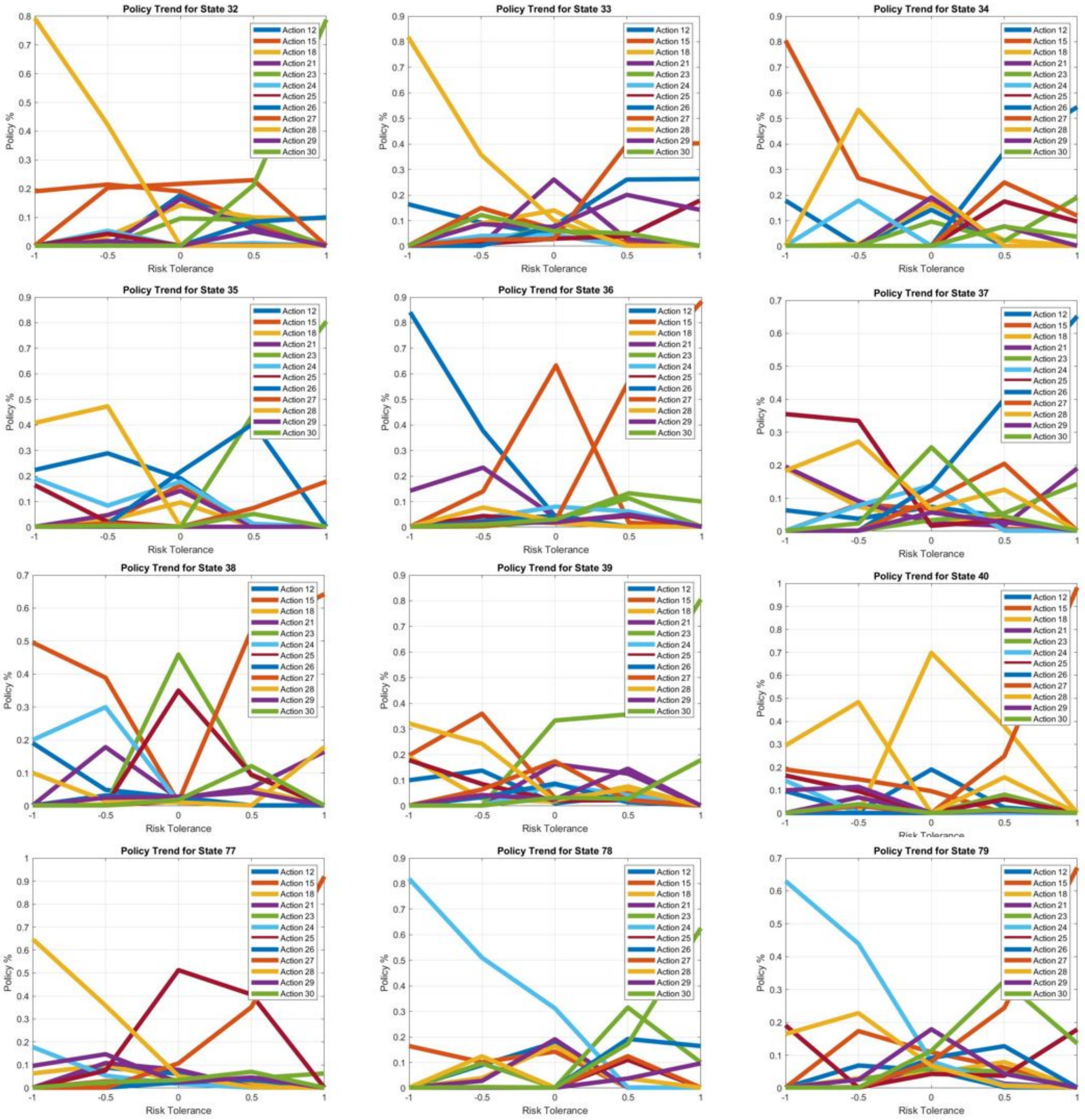


Figure 7.68: Allocation and Acquisition Decision Space RTSPs

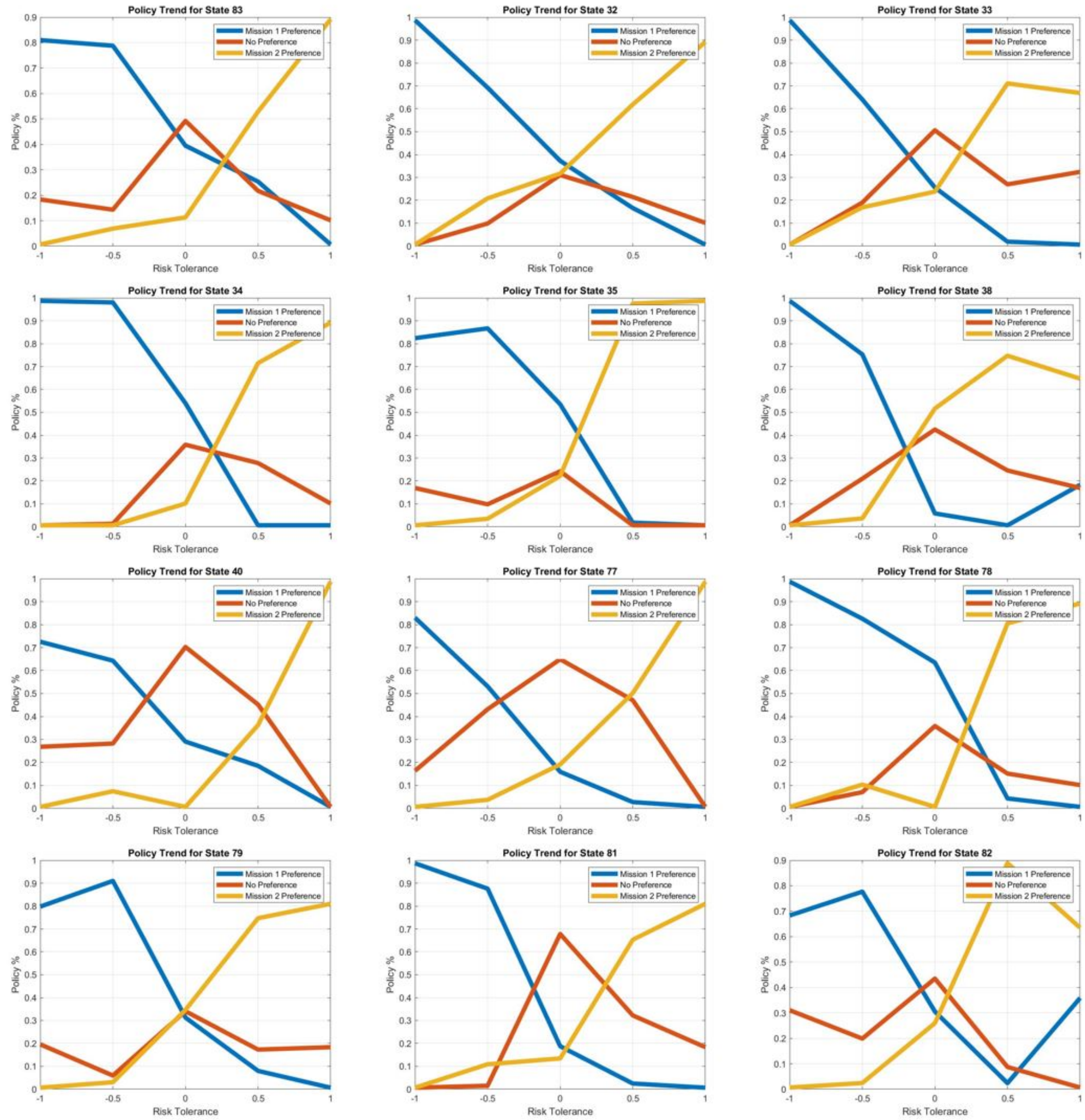


Figure 7.69: Allocation Grouped Decision Space RTSPs

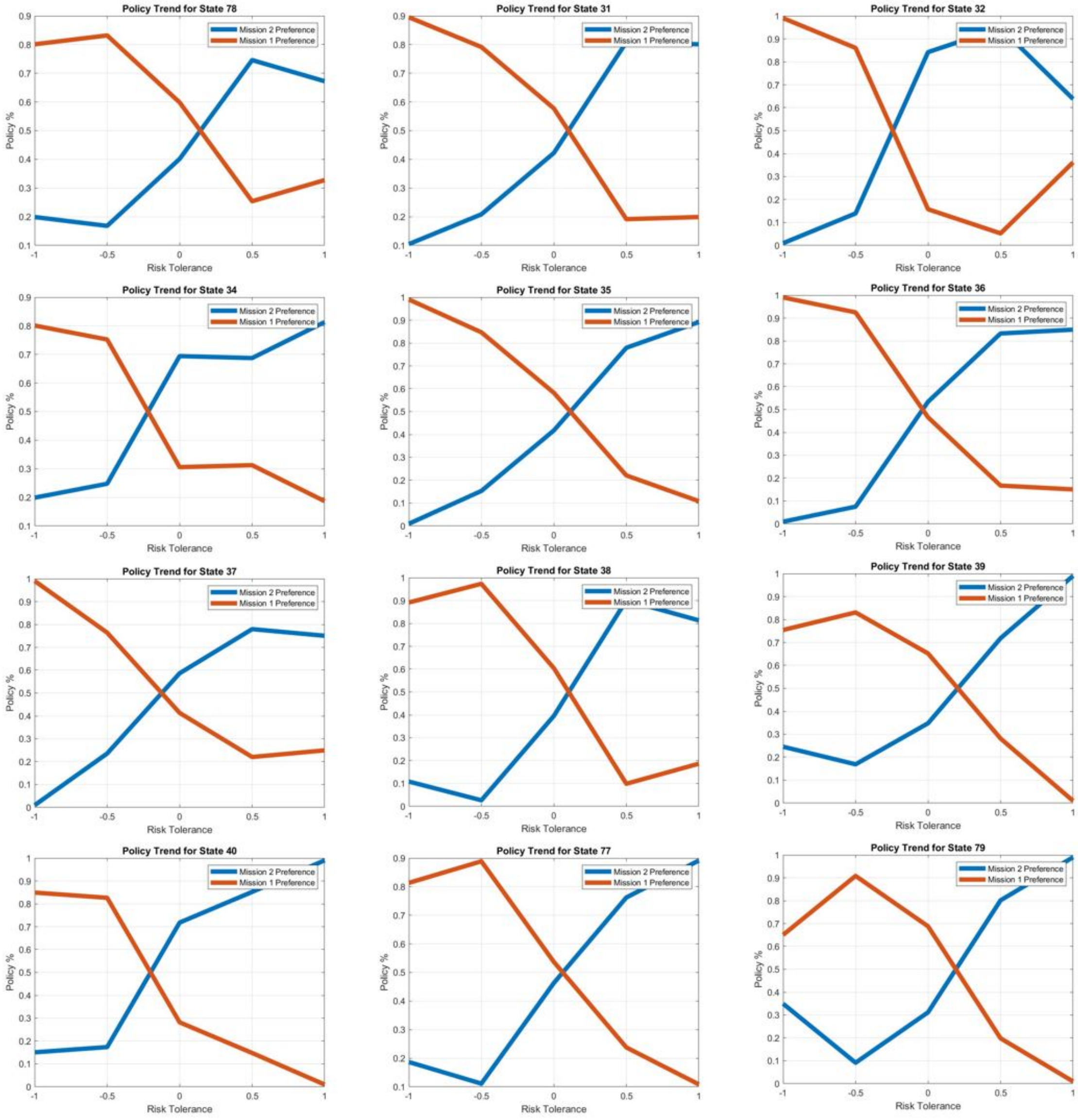


Figure 7.70: Merged Acquire, Develop, and Allocate Decision Space RTSPs

Table 7.5: Acquire, Develop, and Allocate Decision Space

| **Action** | **Acquire System 5** | **Develop System 6** | **Acquire System 7** | **$n$ System 5 to Mission 1** | **$n$ System 5 to Mission 2** | **Mission Preference** |
|---|---|---|---|---|---|---|
| Action 5 | | | ✓ | 0 | 3 | M2 |
| Action 6 | | | ✓ | 1 | 2 | M2 |
| Action 7 | | | ✓ | 2 | 1 | M1 |
| Action 11 | ✓ | ✓ | | 0 | 3 | M2 |
| Action 12 | ✓ | ✓ | | 1 | 2 | M2 |
| Action 13 | ✓ | ✓ | | 2 | 1 | M1 |
| Action 14 | ✓ | ✓ | | 3 | 0 | M1 |
| Action 15 | | | ✓ | 3 | 0 | M1 |

**Impact on Equal Mission Preference and No Budget Constraints** Stakeholder 3 represents a fully unconstrained scenario. At each decision point, all available acquisition and development decisions can be selected. There is no forced decision due to resource constraints. The decision space of interest for Stakeholder 3 (Table 7.6) represents a significant allocation space (seven total assets to allocate). Stakeholder 3 does not have a mission preference by design. The impact can be seen in the allocation grouped Risk-Tolerance Sensitivity Profile portfolio (Figure 7.71)

## 7.2 Experiment Set 2: State Space Compression

Experiment Set 2 evaluates the impact of compression ratio on Risk-Tolerance Sensitivity Profiles for initial and subsequent states. The setups for Experiment Set 1b cases are used as inputs and a state compression ratio of 1 to 0.1 is applied to the full MDP examined in the previous experiment. Risk-Tolerance Sensitivity Profiles are generated from each compressed MDP, or meta-model, for selected states across time steps. The relative change

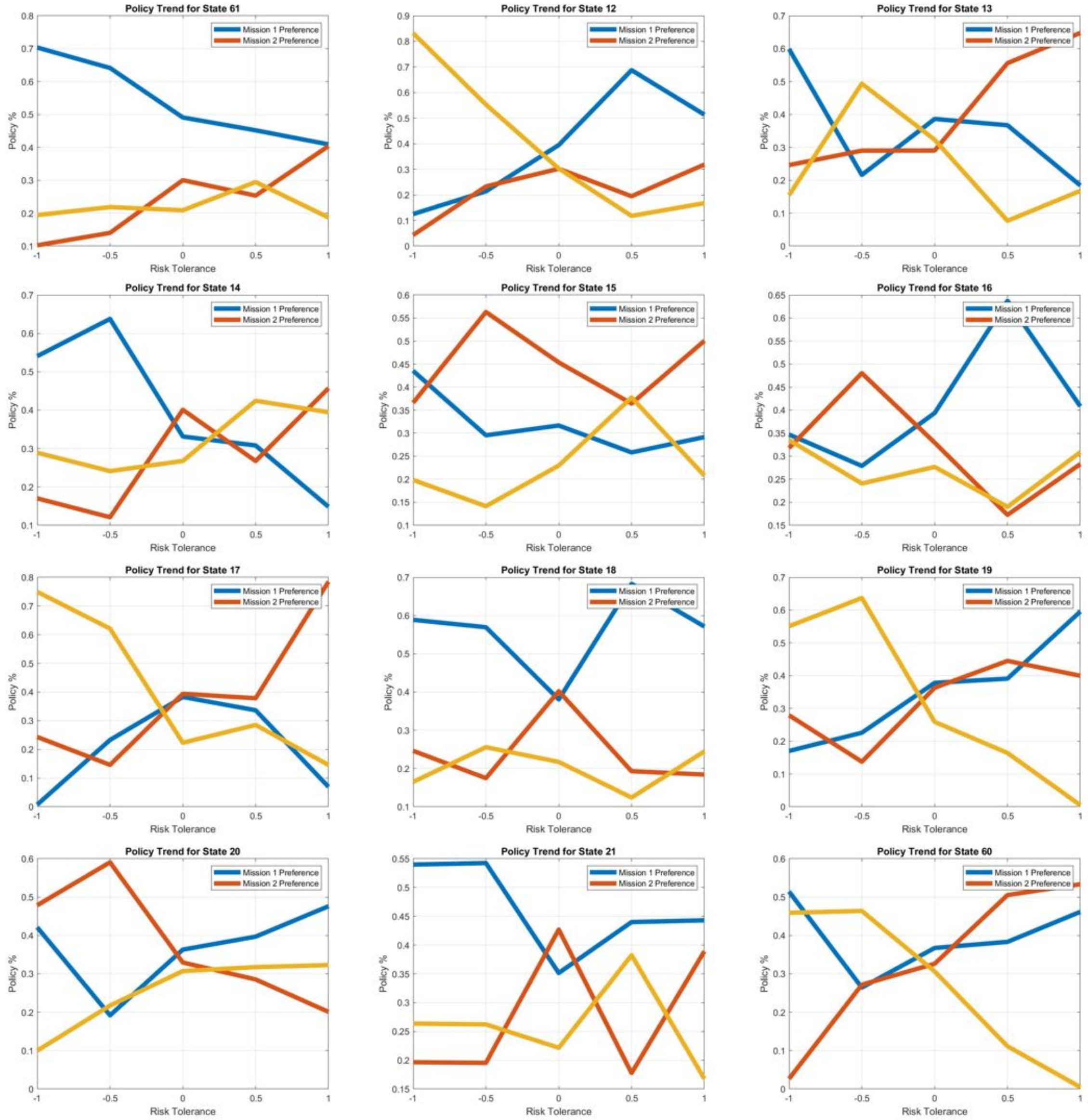


Figure 7.71: Allocation with No Utility Preference Decision Space RTSPs

Table 7.6: Allocation with No Utility Preference Decision Space

| **Action** | **Acquire System 9** | ***n* System 9 to Mission 1** | ***n* System 9 to Mission 2** |
|---|---|---|---|
| Action 10 | ✓ | 1 | 6 |
| Action 11 | ✓ | 2 | 5 |
| Action 12 | ✓ | 3 | 4 |
| Action 13 | ✓ | 4 | 3 |
| Action 14 | ✓ | 6 | 1 |
| Action 31 | ✓ | 0 | 7 |
| Action 32 | ✓ | 7 | 0 |
| Action 36 | ✓ | 5 | 2 |

in profiles is used to judge the impact of the compression algorithm. Additionally, the relative time to calculate the risk-based policies is evaluated as a function of compression ratio.

### 7.2.1 Experiment Set 2a: Repeated Pareto Efficient Actions

*Compressed MDP Visualization*

The Truth Model setup was held constant for Experiment Set 2a. The state compression ratio used in the generation of the meta-model MDP was varied. The resulting risk-policy trends generated using each meta-model were measured along with the mean time to compute a risk-tolerance policy.

Figure 7.72 depicts the full MDP graph. The depiction is a visual baseline for the size and complexity of the three action simulation. At each time step a decision point state is highlighted. A decision point is a state where a acquisition selection can be made by Stakeholder 1. These highlighted decision points are selected points where the risk-

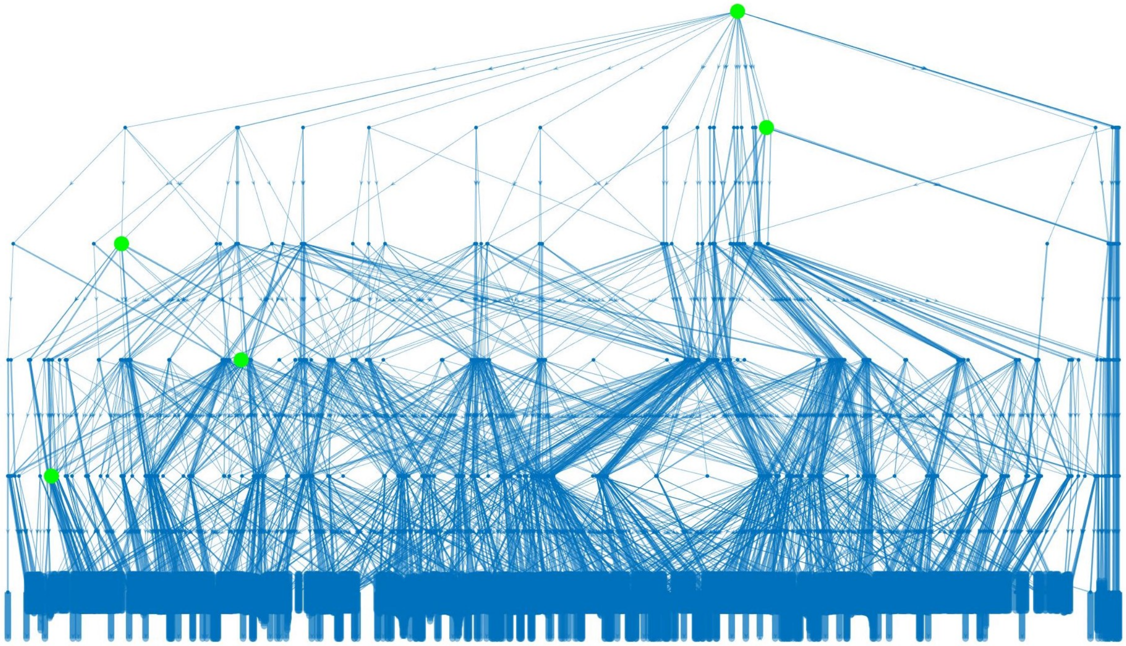


Figure 7.72: Experiment Set 2a: Full MDP Graph with Highlighted States of Interest

tolerance sensitivity will be compared across compression ratios.

The graph for a 75% reduction in states space meta-model is captured in Figure 7.73. Little visual change can be noted. As the compression is decreased to 50% (Figure 7.74), 25% (Figure 7.75), and 10% (Figure 7.76) the visual change becomes more apparent. The breakdown of the original structure can clearly be seen as the compression ratio reaches 10%.

*Risk-Tolerance Sensitivity Comparison*

At time step $t = 0$ there is a single state. This state is the initializing state for all episodes that sample the Truth Model. The initial state is the most sampled state and has the most sampling of future states. The resulting risk-tolerance sensitivity as a function of meta-model compression ratio is depicted in Figure 7.77. The resulting risk-tolerance sensitivity is maintained through a 50% compression. Anticipated, Identifiable trends can be observed at the lower compression ratios of 25% and 10%. At the lower compression ratio mild

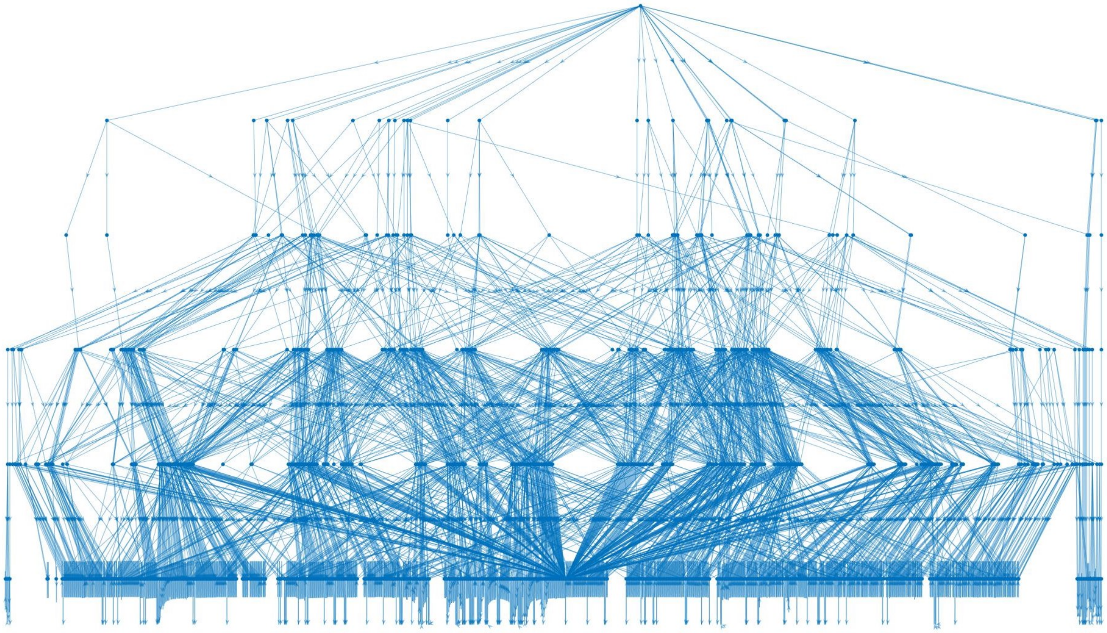


Figure 7.73: Experiment Set 2a: 75% State Compressed MDP Graph

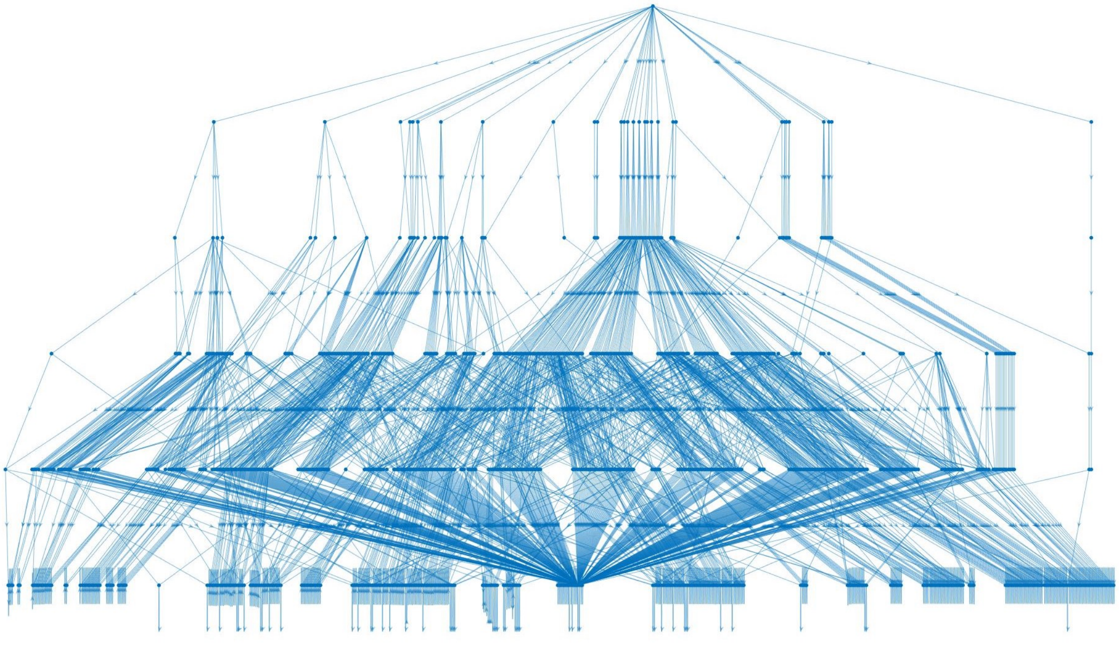


Figure 7.74: Experiment Set 2a: 50% State Compressed MDP Graph

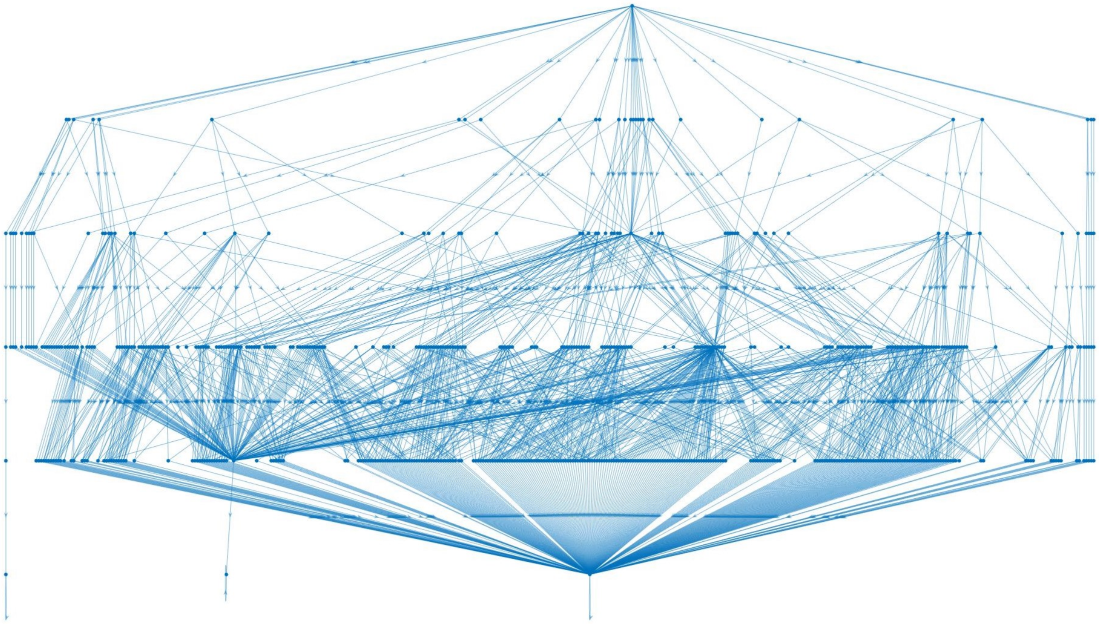


Figure 7.75: Experiment Set 2a: 25% State Compressed MDP Graph

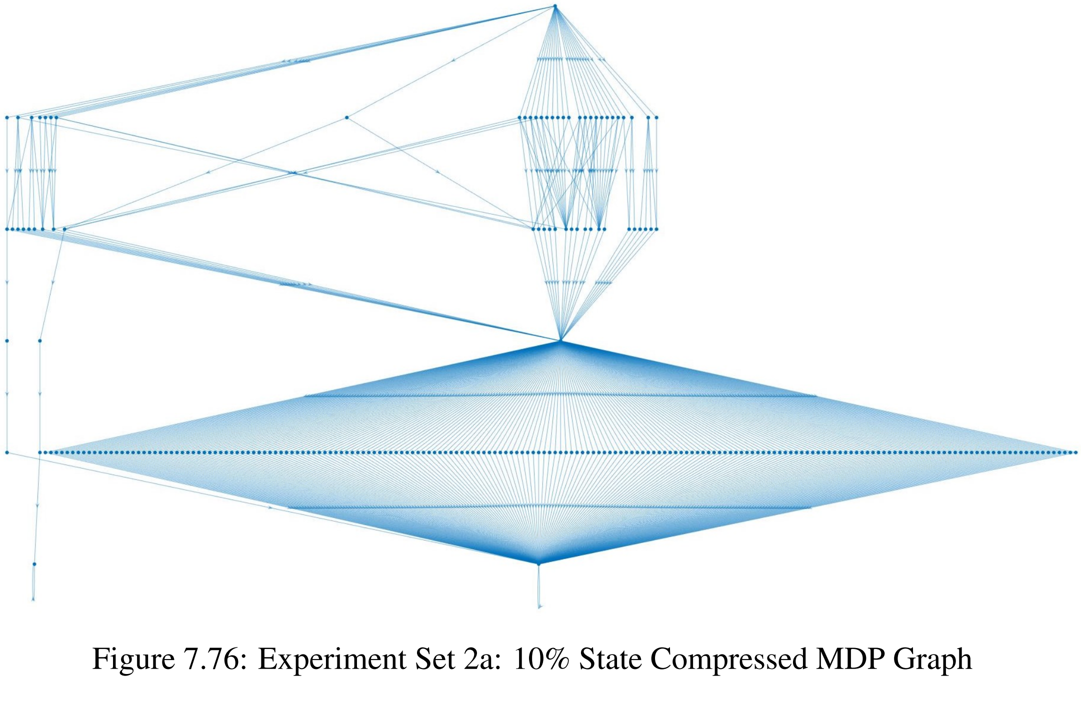


Figure 7.76: Experiment Set 2a: 10% State Compressed MDP Graph

anomalies can be seen in increased asymmetry.

At time step $t = 1$, the selected state has two options (Acquire System 1 or Acquire System 2). This presents a simpler action space than what was seen at $t = 0$. Little fidelity in the risk-sensitivity profile is lost as the compression ratio reaches 50% (Figure 7.78). The shifting of the equal policy point and breakdown of symmetry at 25% and 10% is once again seen.

At time step $t = 2$, there are far less state and future state samples. The breakdown in symmetry begins to be seen at a compression ratio of 50% (Figure 7.79). At a compression ratio of 10% the risk-tolerance sensitivity profile is fully lost.

As the time steps increase ($t = 3$ and $t = 4$) the ability to sample future states decreases along with the total number of state samples. The information gathered and used to generate earlier time steps remains valid. Future sampling for later states is not available. The resulting risk profiles (Figure 7.80 and Figure 7.81) show skewed profiles and earlier breakdown as the compression ratio is decreased.

*Policy Generation Computation Time*

The average computation time for risk-based policy generation across risk-tolerance values significantly decreased as the compression ratio decreased (Figure 7.82). A linear relationship in computation time and the number of states can be observed as expected.

### 7.2.2 Experiment Set 2b: Acquire vs. Develop Scenario

The Truth Model setup was held constant for Experiment Set 2b. The state compression ratio used in the generation of the meta-model MDP was varied. The resulting risk-policy trends generated using each meta-model were measured along with the mean time to compute a risk-tolerance policy.

Figure 7.83 depicts the full MDP graph. The depiction is a visual baseline for the size and complexity of the three action simulation. The selected decision points, or states

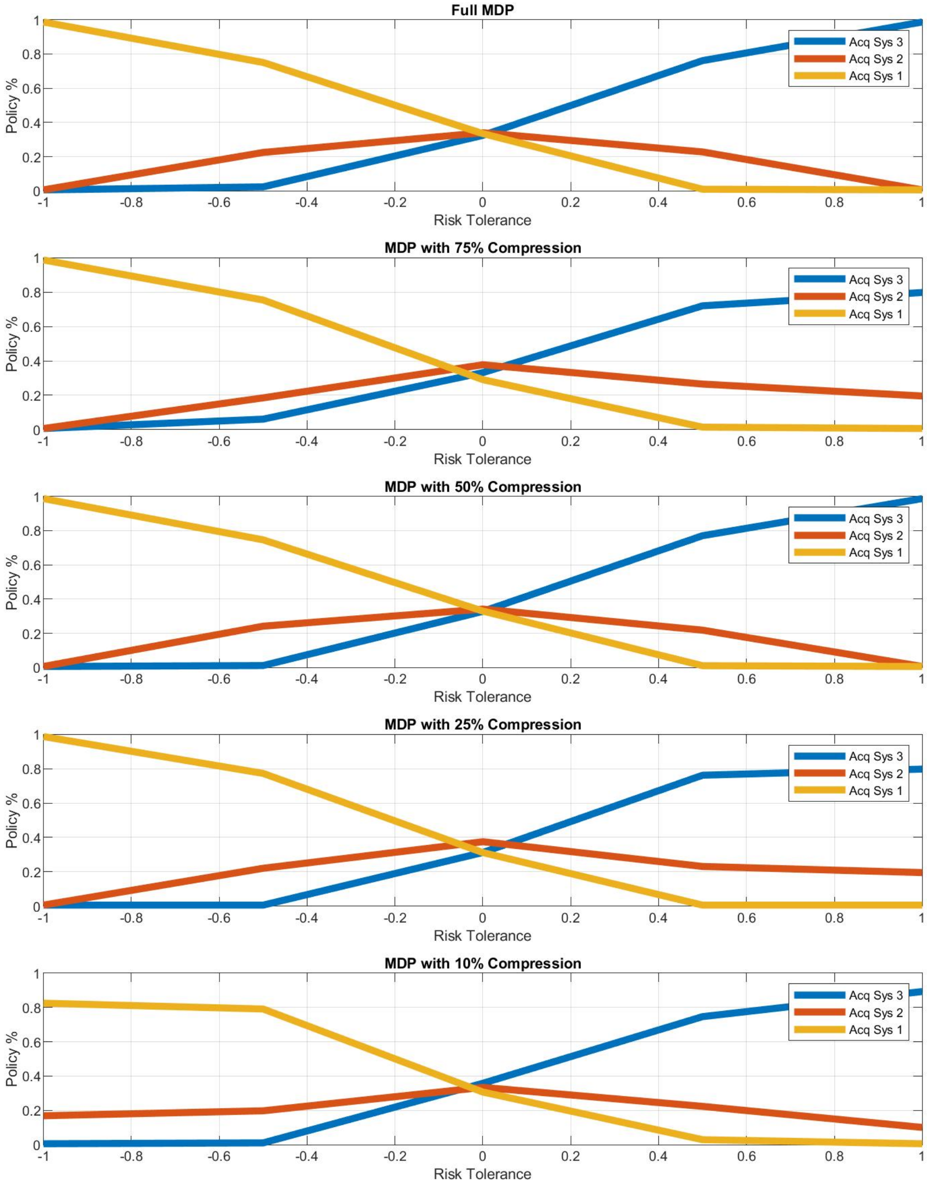


Figure 7.77: Experiment Set 2a: Time Step $t = 0$ State of Interest RTSP versus State Compression Ratio

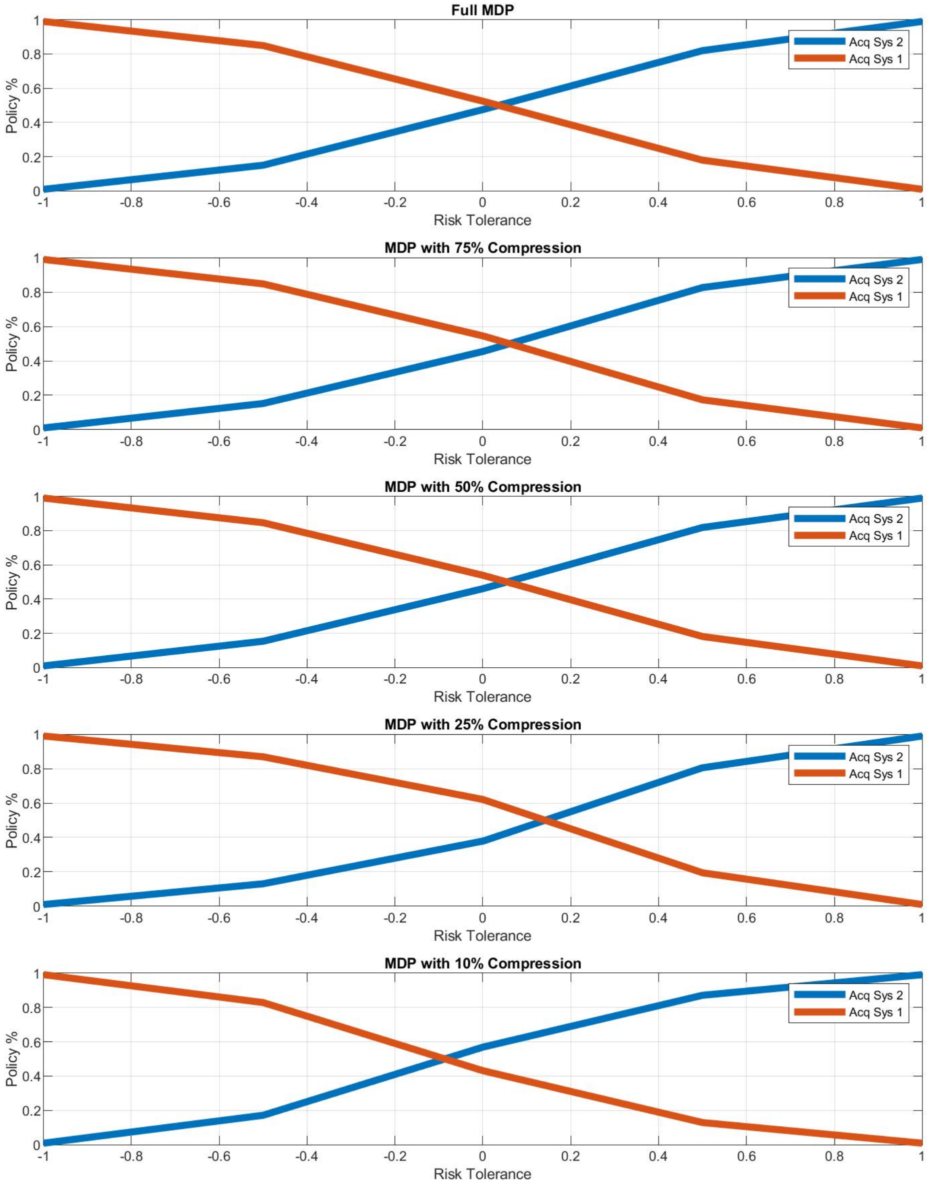


Figure 7.78: Experiment Set 2a: Time Step $t = 1$ State of Interest RTSP versus State Compression Ratio

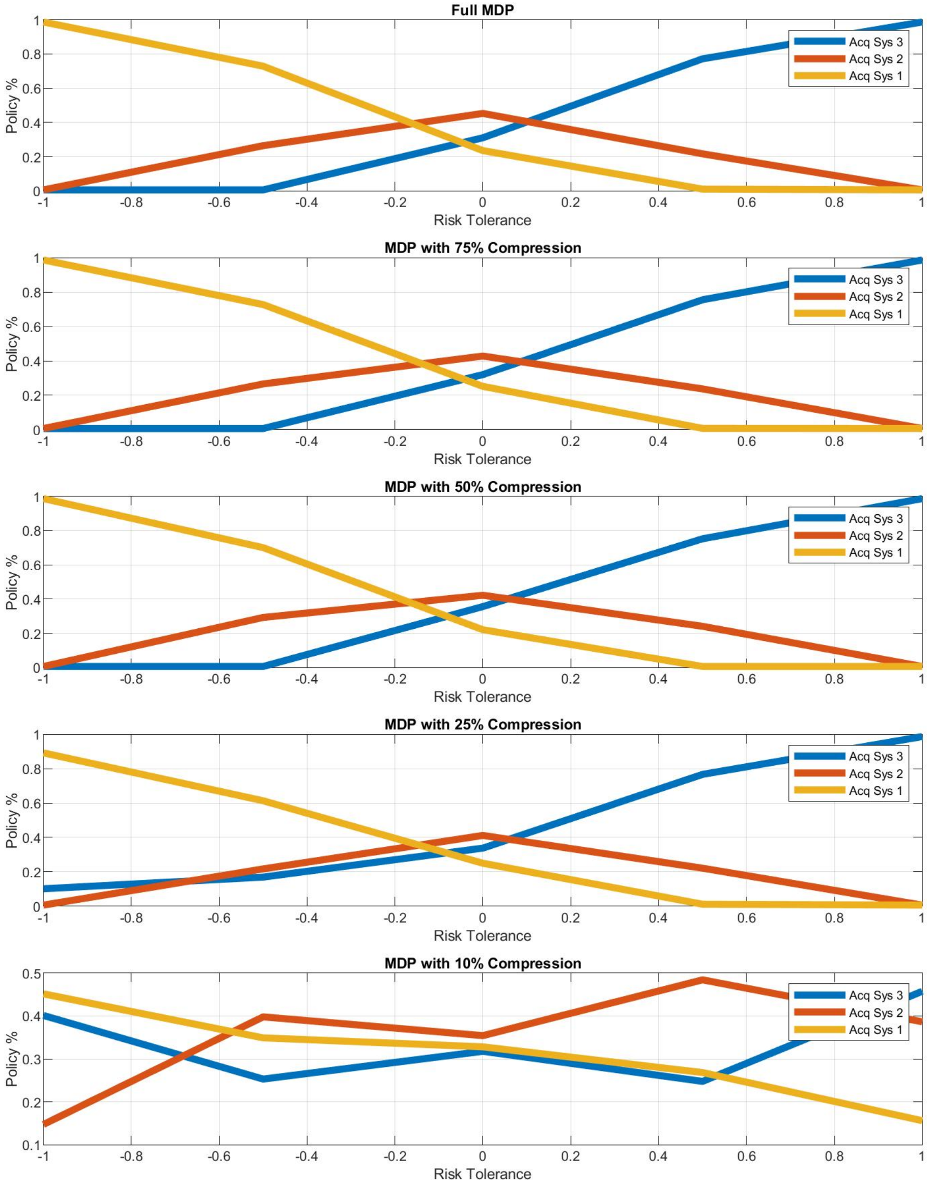


Figure 7.79: Experiment Set 2a: Time Step $t = 2$ State of Interest RTSP versus State Compression Ratio

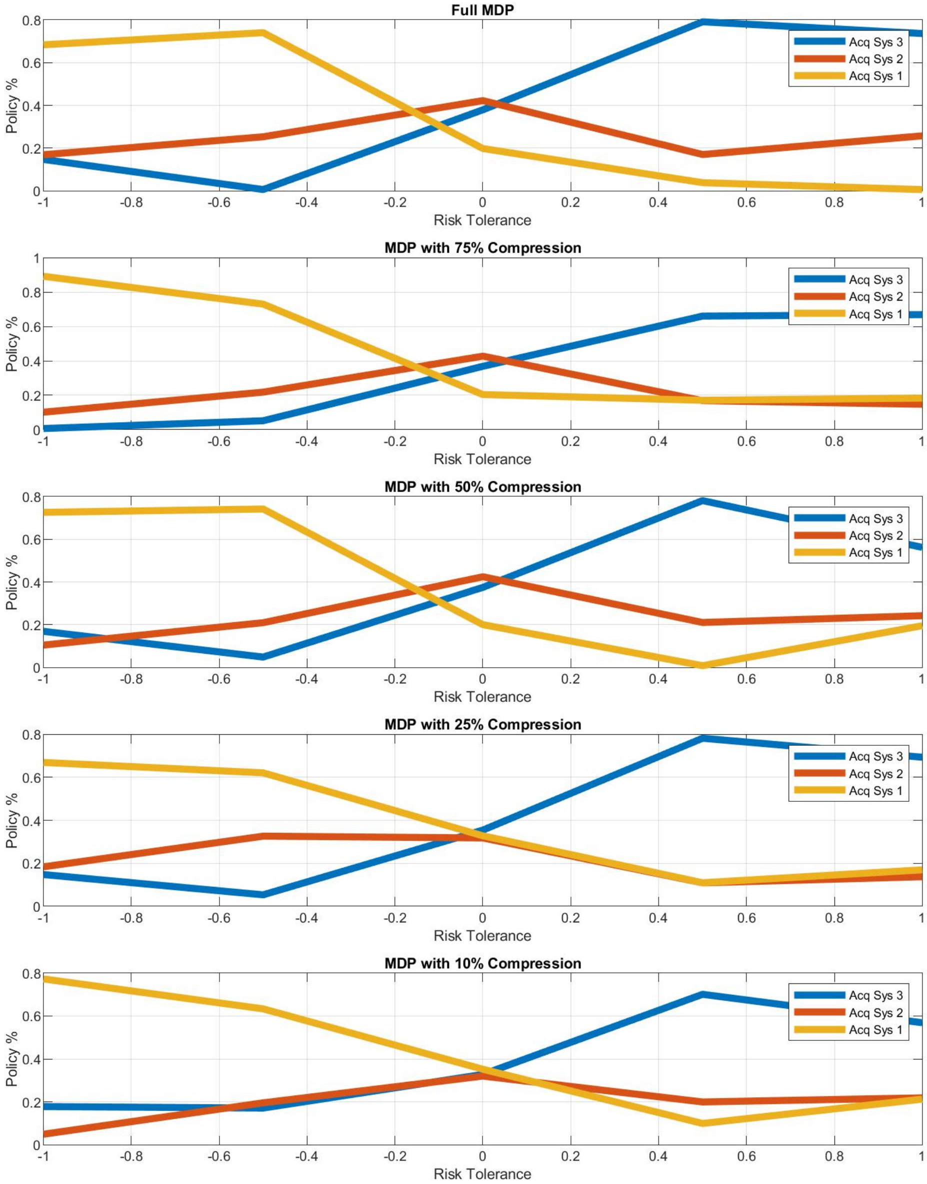


Figure 7.80: Experiment Set 2a: Time Step $t = 3$ State of Interest RTSP versus State Compression Ratio

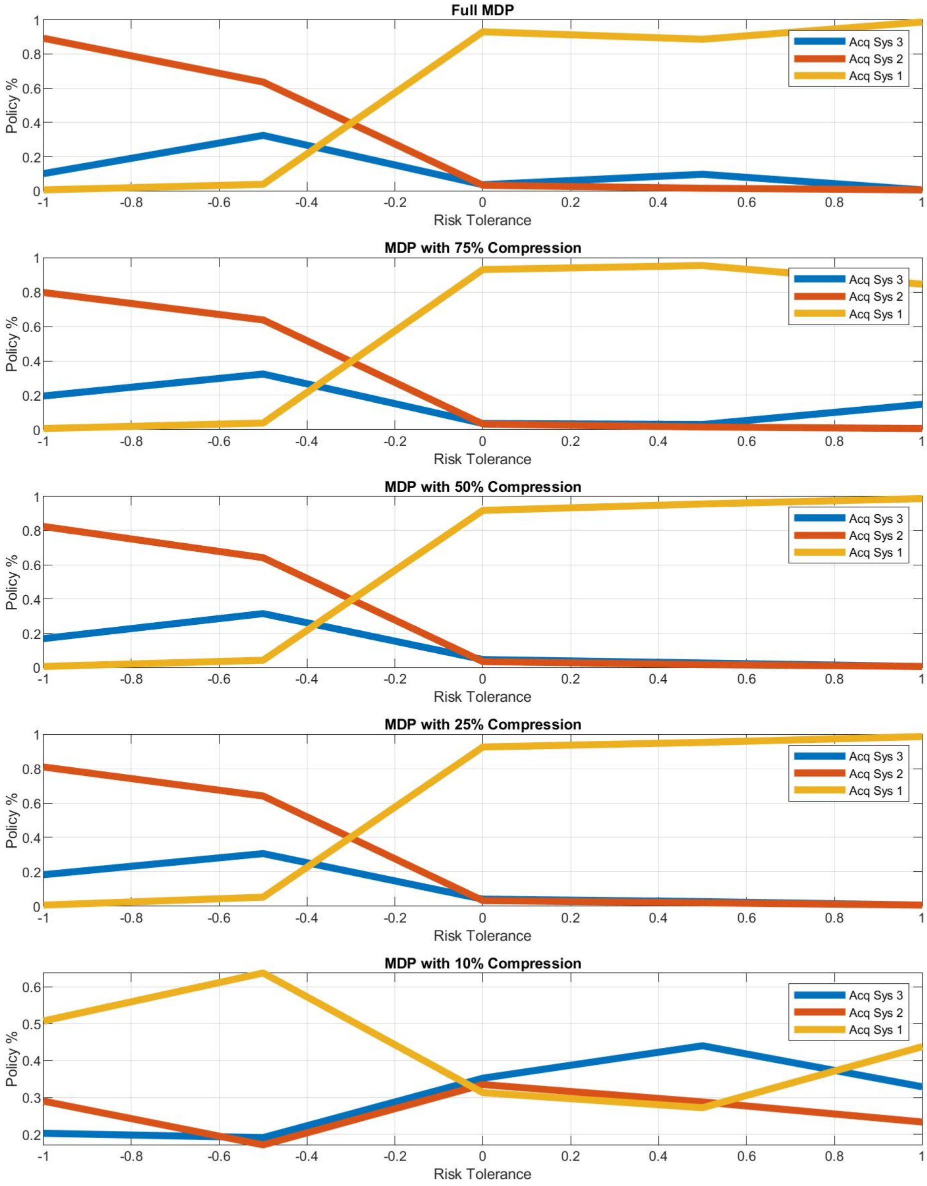


Figure 7.81: Experiment Set 2a: Time Step $t = 4$ State of Interest RTSP versus State Compression Ratio

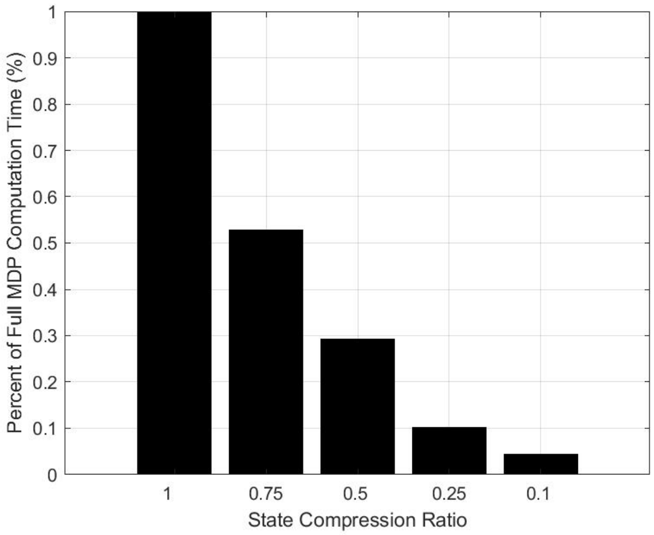


Figure 7.82: Experiment Set 2a: Computation Time

where multiple actions are available, for evaluation are highlighted. The policy profiles and computation time examined as a function of compression ratio.

The graph for a 75% reduction in states space meta-model is captured in Figure 7.84. Again, little visual change can be noted. As the compression is decreased to 50% (Figure 7.85), 25% (Figure 7.86), and 10% (Figure 7.87) the visual change becomes more apparent. The breakdown of the original structure can clearly be seen as the compression ratio reaches 10%.

*Risk-Tolerance Sensitivity Comparison*

The initializing state is the sole state at $t = 0$. This state is the initializing state for all episodes that sample the Truth Model. The resulting risk-tolerance sensitivity as a function of meta-model compression ratio for the first state is depicted in Figure 7.88. The resulting risk-tolerance sensitivity maintained through a 50% compression. Anticipated, Identifiable trends can be observed at the lower compression ratios of 25% and 10%. At the lower compression ratio mild anomalies can be seen in increased asymmetry.

At time step $t = 1$, the selected state has two options (Acquire System 1 or Develop

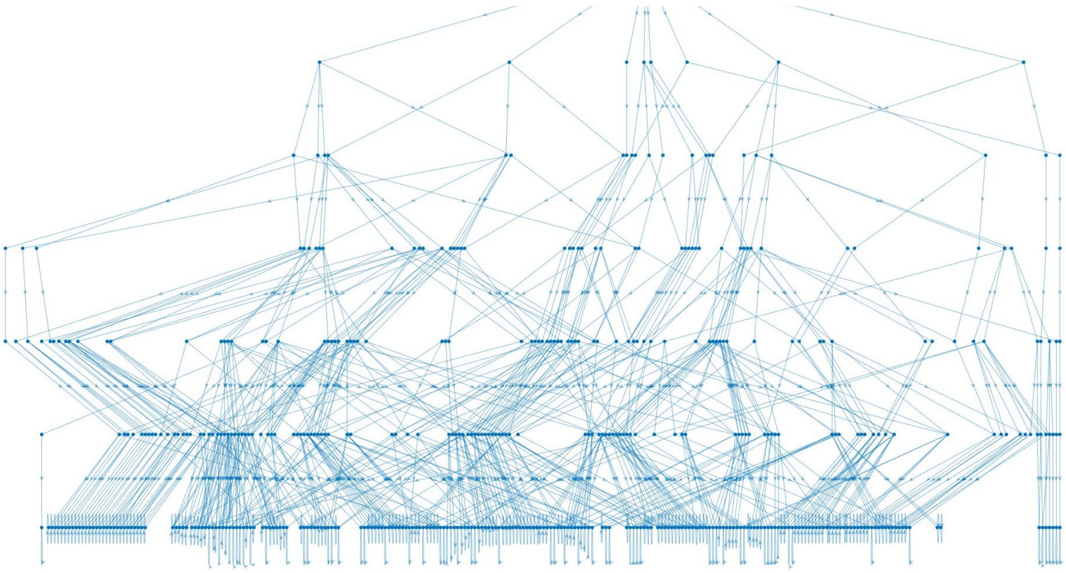


Figure 7.83: Experiment Set 2b: Full MDP Graph with Highlighted States of Interest

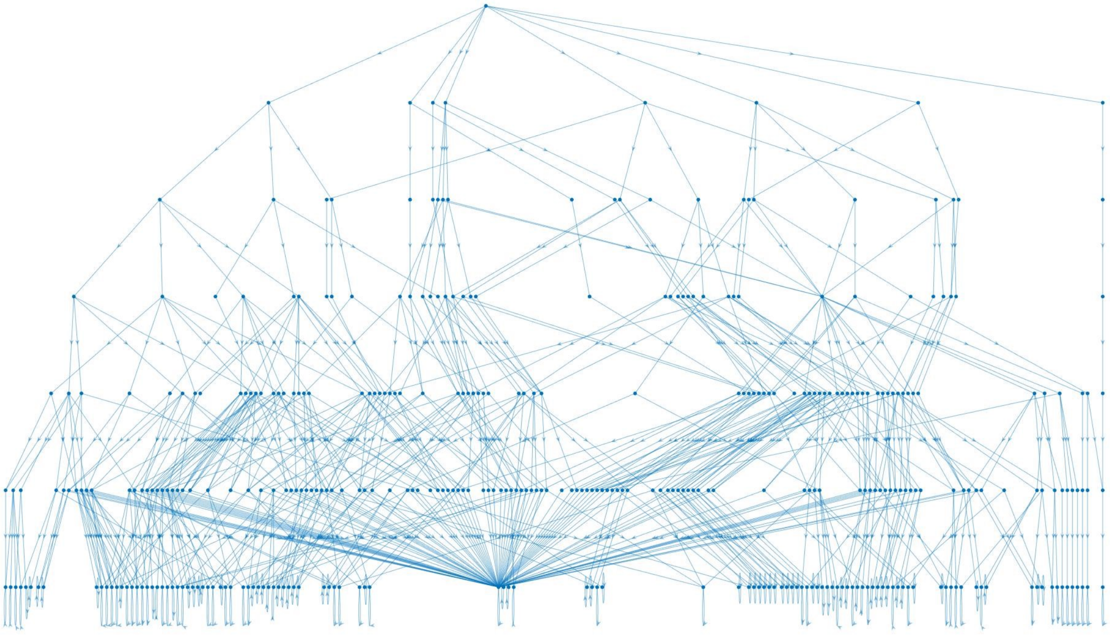


Figure 7.84: Experiment Set 2b: 75% State Compressed MDP Graph

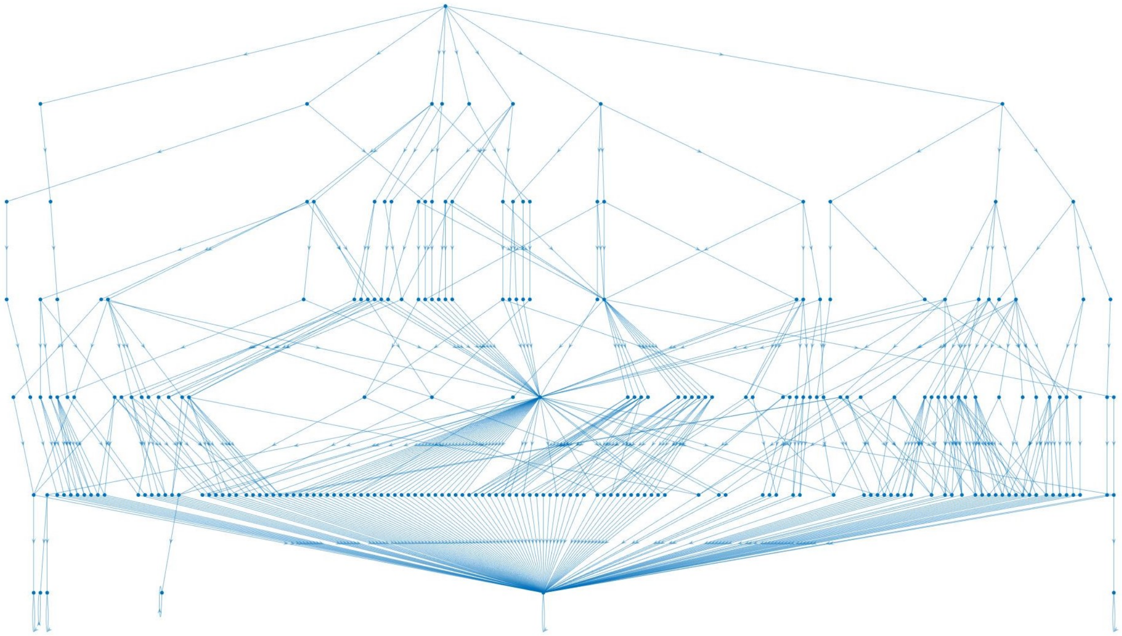


Figure 7.85: Experiment Set 2b: 50% State Compressed MDP Graph

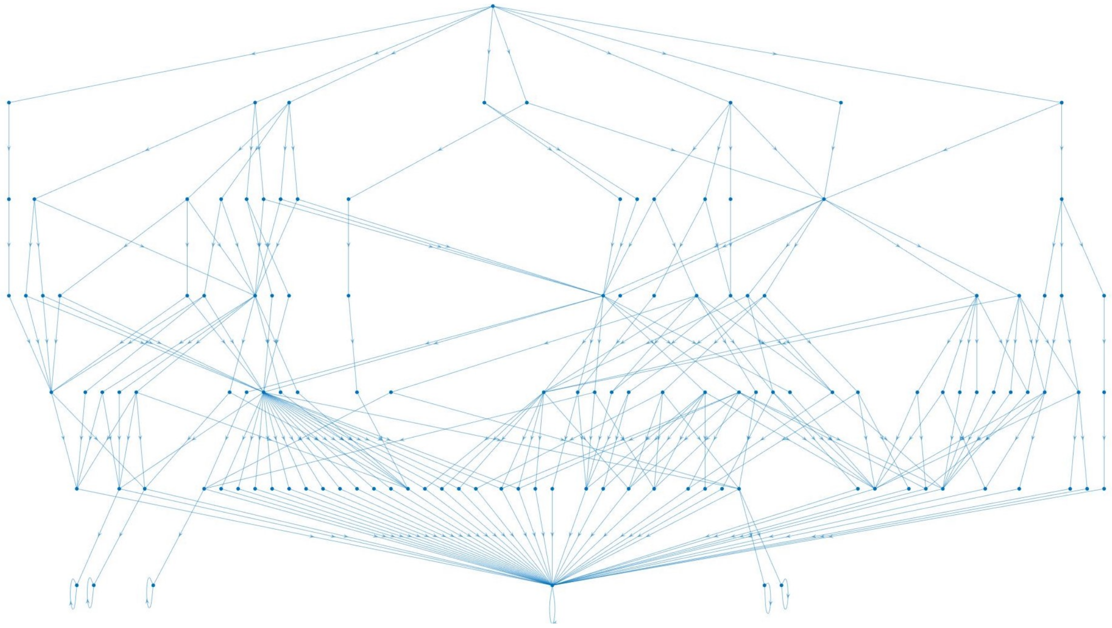


Figure 7.86: Experiment Set 2b: 25% State Compressed MDP Graph

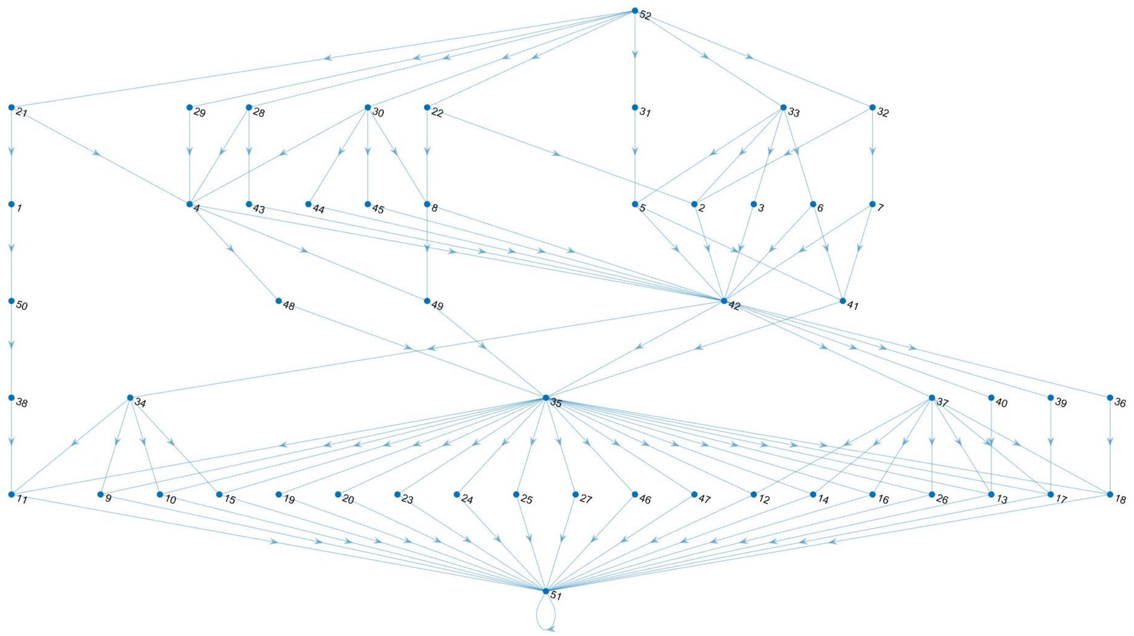


Figure 7.87: Experiment Set 2b: 10% State Compressed MDP Graph

System 2). Little fidelity in the risk-sensitivity profile is lost as the compression ratio reaches 50% (Figure 7.89). A loss in fidelity is seen at a 10% compression ratio as in the high risk-tolerance response.

At time step $t = 2$, there are far less state and future state samples. The breakdown in consistency is observed at a compression ratio of 10% (Figure 7.90). Little is lost at higher compression ratios.

As the time steps increase ($t = 3$) the ability to sample future states decreases along with the total number of state samples. The information gathered and used to generate earlier time steps remains valid. Future sampling for later states is not available. The resulting risk profiles (Figure 7.91 show skewed profiles and earlier breakdown as the compression ratio is decreased. At previous time steps the breakdown was observed by a 10% compression ratio. At $t = 3$ the breakdown is seen of the profile is seen by 25% percent.

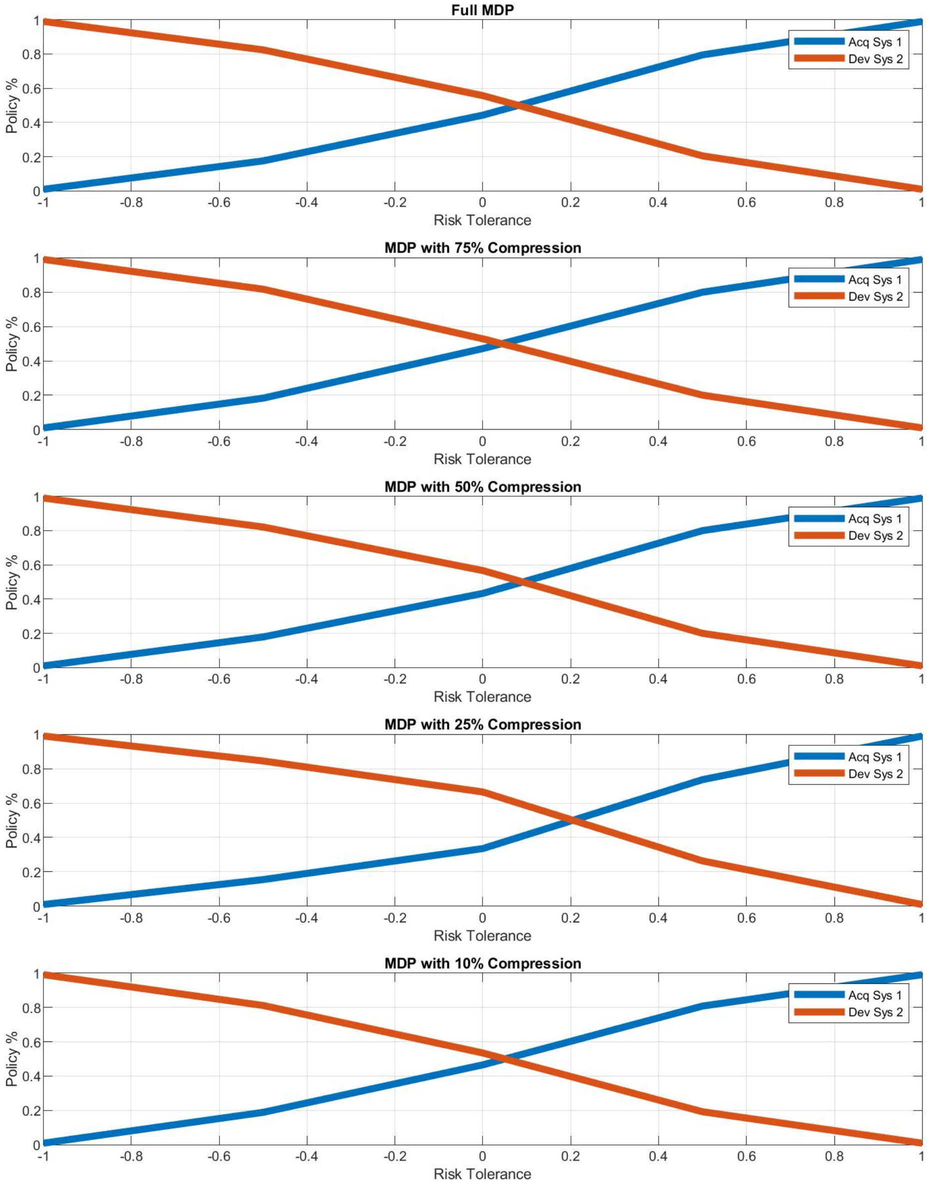


Figure 7.88: Experiment Set 2b: Time Step $t = 0$ State of Interest RTSP versus State Compression Ratio

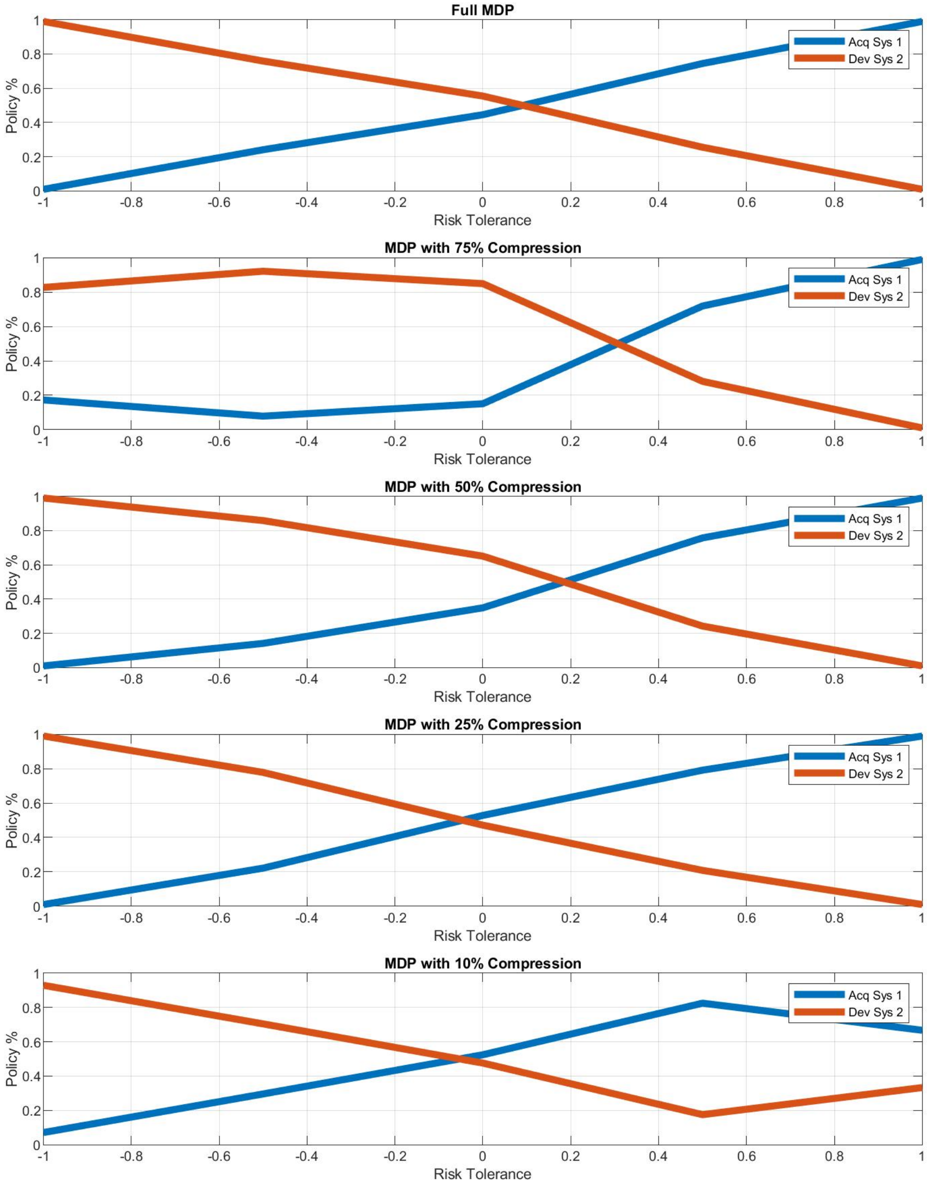
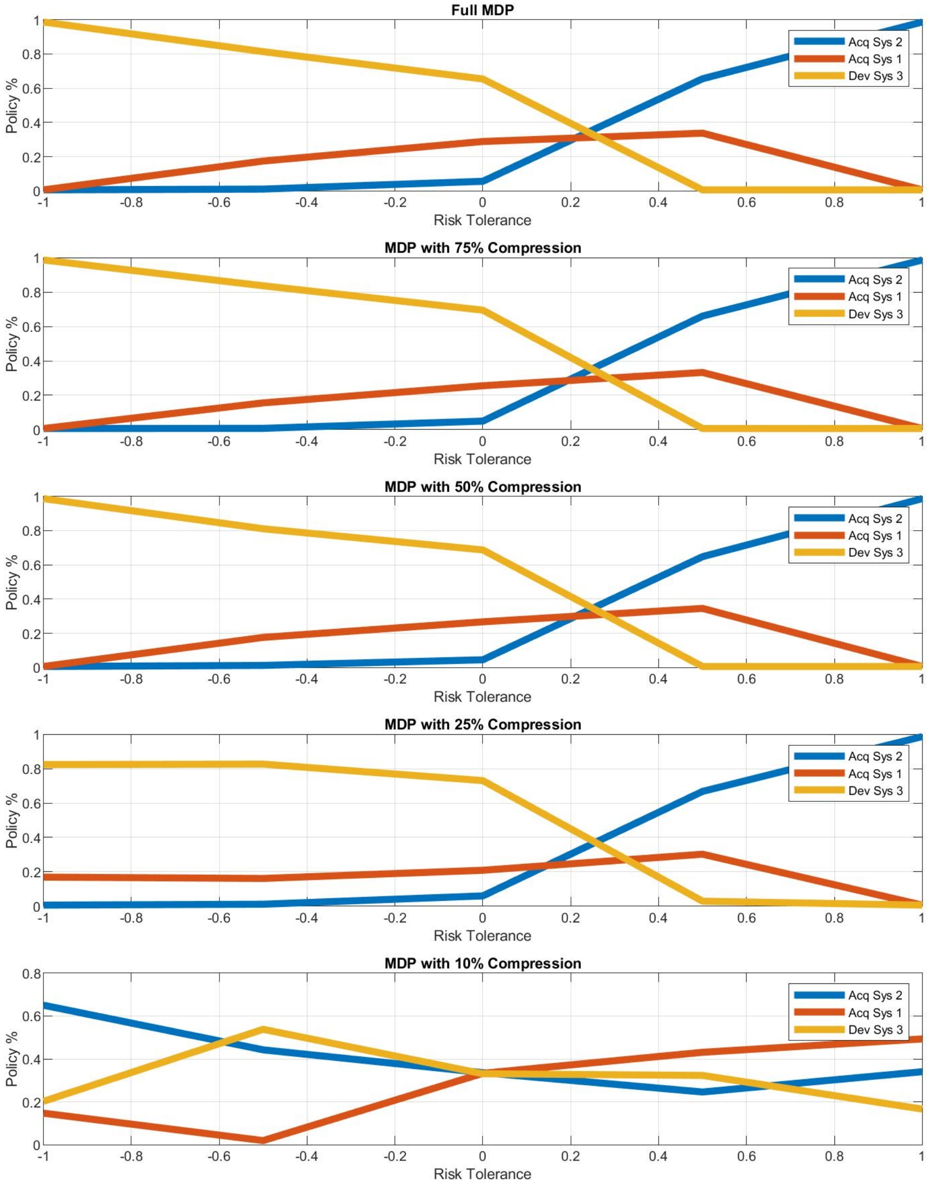


Figure 7.89: Experiment Set 2b: Time Step $t = 1$ State of Interest RTSP versus State Compression Ratio



Full MDP

MDP with 75% Compression

MDP with 50% Compression

MDP with 25% Compression

MDP with 10% Compression

Figure 7.90: Experiment Set 2b: Time Step $t = 2$ State of Interest RTSP versus State Compression Ratio

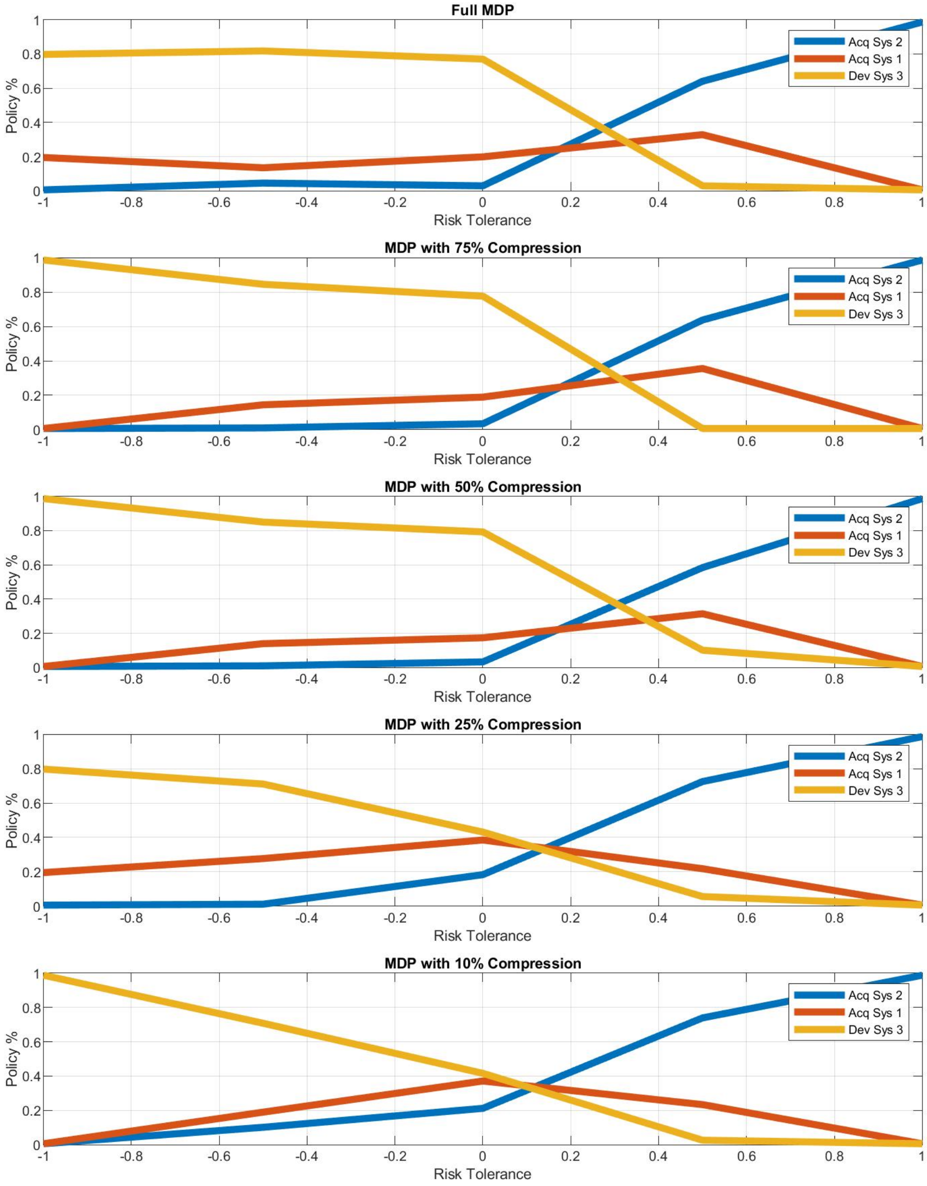


Figure 7.91: Experiment Set 2b: Time Step $t = 3$ State of Interest RTSP versus State Compression Ratio

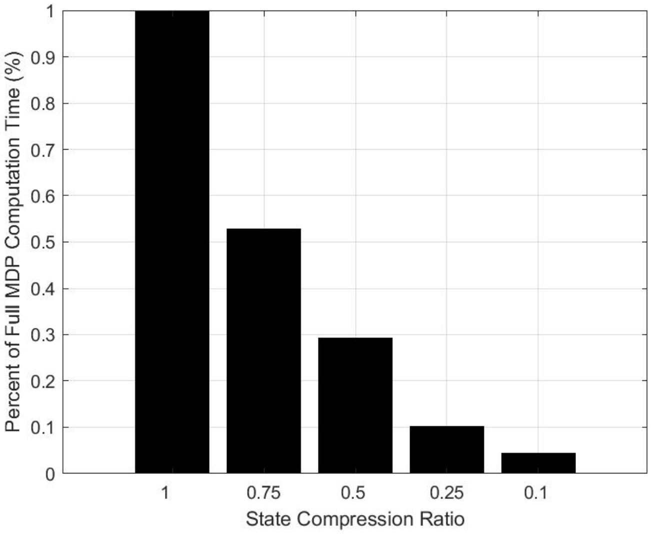


Figure 7.92: Experiment Set 2b: Computation Time

*Policy Generation Computation Time*

The average computation time for risk-based policy generation across risk-tolerance values significantly decreased as the compression ratio decreased (Figure 7.92). A significant negative relationship between computation time and the number of states can be observed as expected.

### 7.2.3 Experiment Set 2c: Multi-Mission Acquire vs. Develop Scenario

Experiment Set 2c builds on the Experiment Set 1b Case 3 and the multiple stakeholders. Each stakeholder represents a varying degrees of complexity between asset allocation and creation decisions.

*Stakeholder 1*

The Stakeholder 1 MDP structure from full representation to 10% compression ratio are depicted in Figure 7.93, Figure 7.94, Figure 7.95, Figure 7.96, and Figure 7.97. The selected evaluation states are highlighted in Figure 7.98.

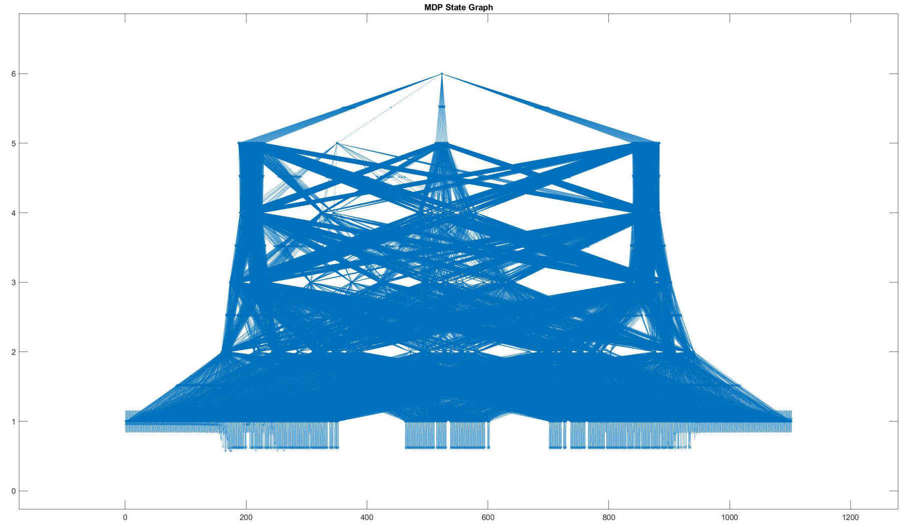


Figure 7.93: Experiment Set 2c: Stakeholder 1 Full MDP Graph with Highlighted States of Interest

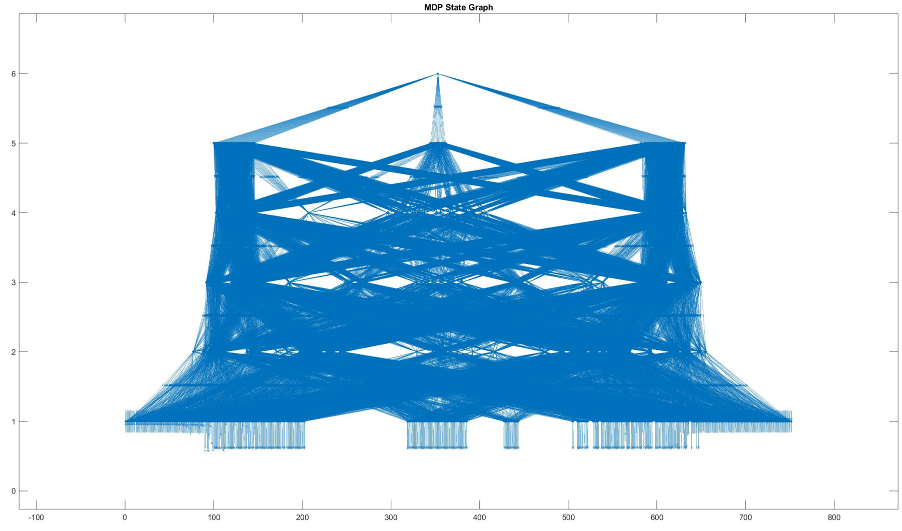


Figure 7.94: Experiment Set 2c: Stakeholder 1 75% State Compressed MDP Graph

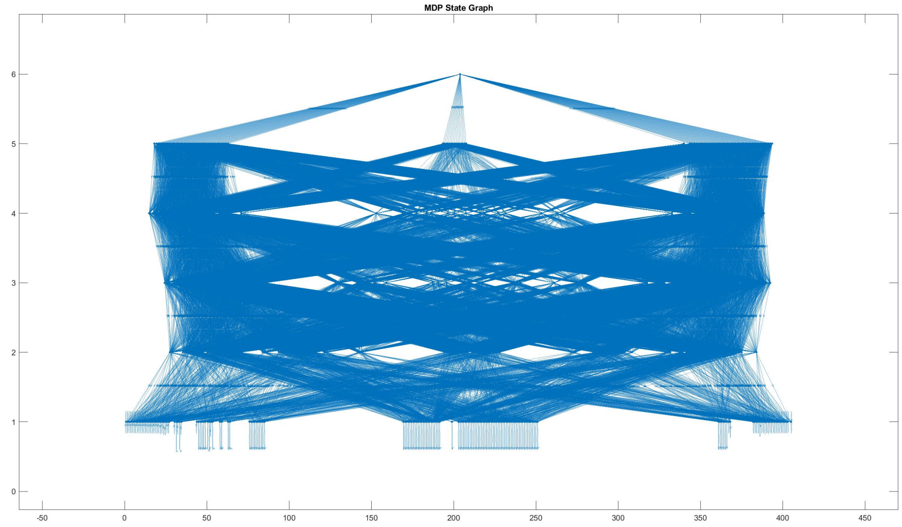


Figure 7.95: Experiment Set 2c: Stakeholder 1 50% State Compressed MDP Graph

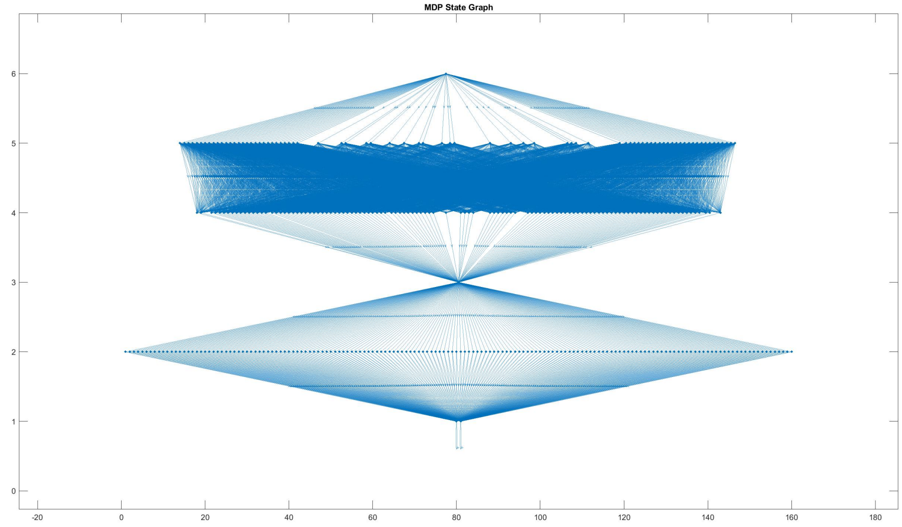


Figure 7.96: Experiment Set 2c: Stakeholder 1 25% State Compressed MDP Graph

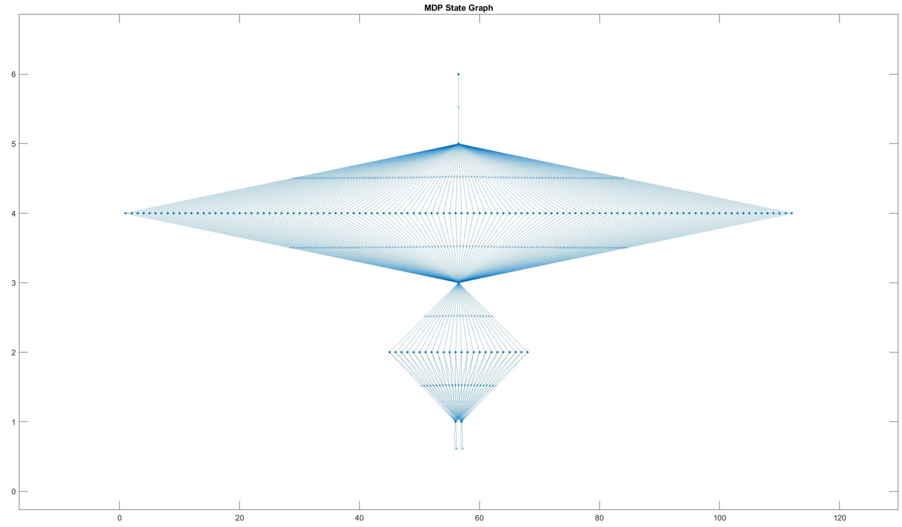


Figure 7.97: Experiment Set 2c: Stakeholder 1 10% State Compressed MDP Graph

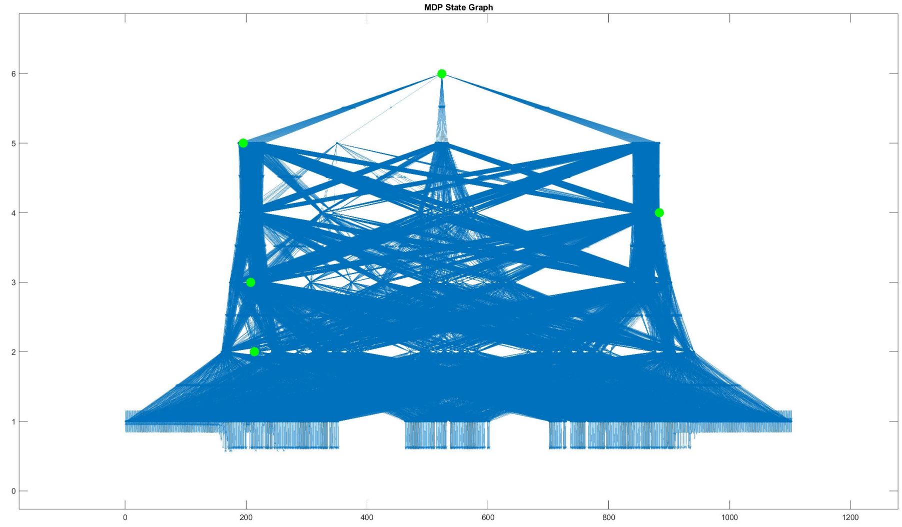


Figure 7.98: Experiment Set 2c: Stakeholder 1 Selected States for Evaluation

The selected state Risk-Tolerance Sensitivity Profiles are depicted in Figure 7.99, Figure 7.99, Figure 7.101, Figure 7.102, and Figure 7.103. Each sensitivity profile maintains the originating full profile through at compression ratio of at least 50% with artifacts appearing by a compression ratio of 10%.

#### *Stakeholder 2*

. The Stakeholder 2 MDP structure from full representation to 10% compression ratio are depicted in Figure 7.104, Figure 7.105, Figure 7.106, Figure 7.106, and Figure 7.107. The selected evaluation states are highlighted in Figure 7.108.

The selected state Risk-Tolerance Sensitivity Profiles are depicted in Figure 7.109, Figure 7.109, Figure 7.111, Figure 7.112, and Figure 7.113. Each sensitivity profile maintains the originating full profile through at compression ratio of at least 50% with artifacts appearing by a compression ratio of 10%.

#### *Stakeholder 3*

The Stakeholder 3 MDP structure from full representation to 10% compression ratio are depicted in Figure 7.114, Figure 7.115, Figure 7.116, Figure 7.117, and Figure 7.118. The selected evaluation states are highlighted in Figure 7.119.

The selected state Risk-Tolerance Sensitivity Profiles are depicted in Figure 7.109, Figure 7.120, Figure 7.122, Figure 7.123, and Figure 7.124. Each sensitivity profile maintains the originating full profile through at compression ratio of at least 50% with artifacts appearing by a compression ratio of 10%.

#### *Computation Time*

The computation time for all three stakeholders mainta)ins the signficant negative correlation relationship with compression ratio seen in above in less complex Experiment Set 2 sub-experiments (Figure 7.125a, Figure 7.125b, and Figure 7.125c.

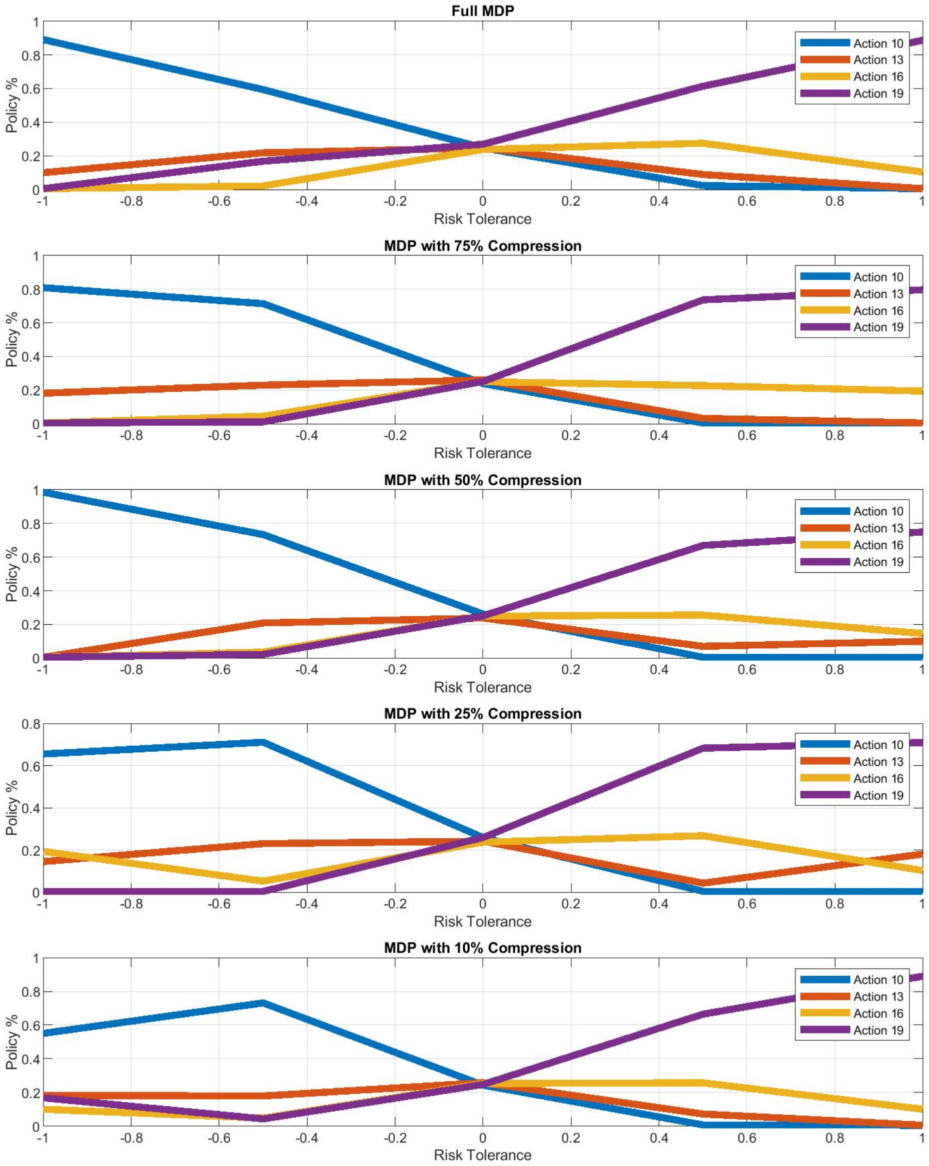


Figure 7.99: Experiment Set 2c: Stakeholder 1 Time Step $t = 0$ State of Interest RTSP versus State Compression Ratio

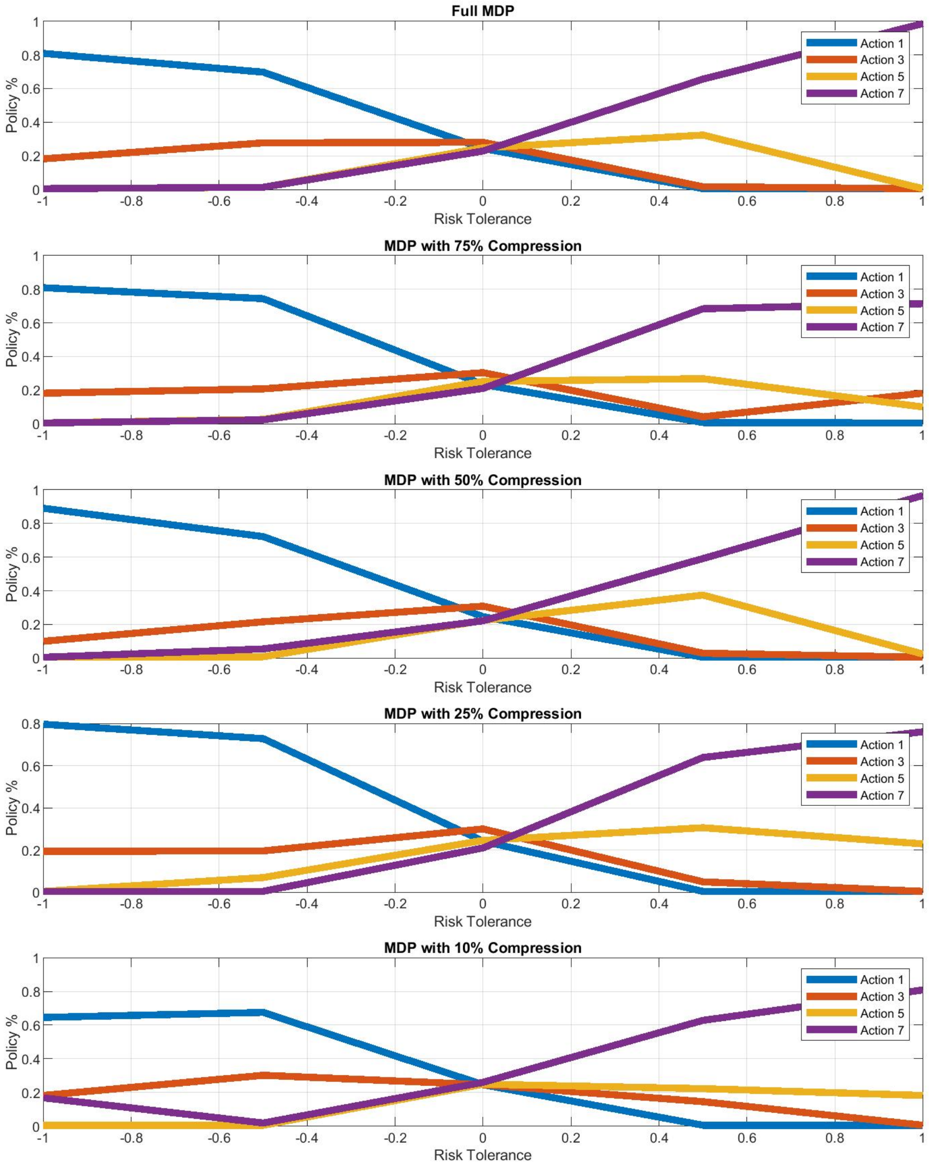


Figure 7.100: Experiment Set 2c: Stakeholder 1 Time Step $t = 1$ State of Interest RTSP versus State Compression Ratio

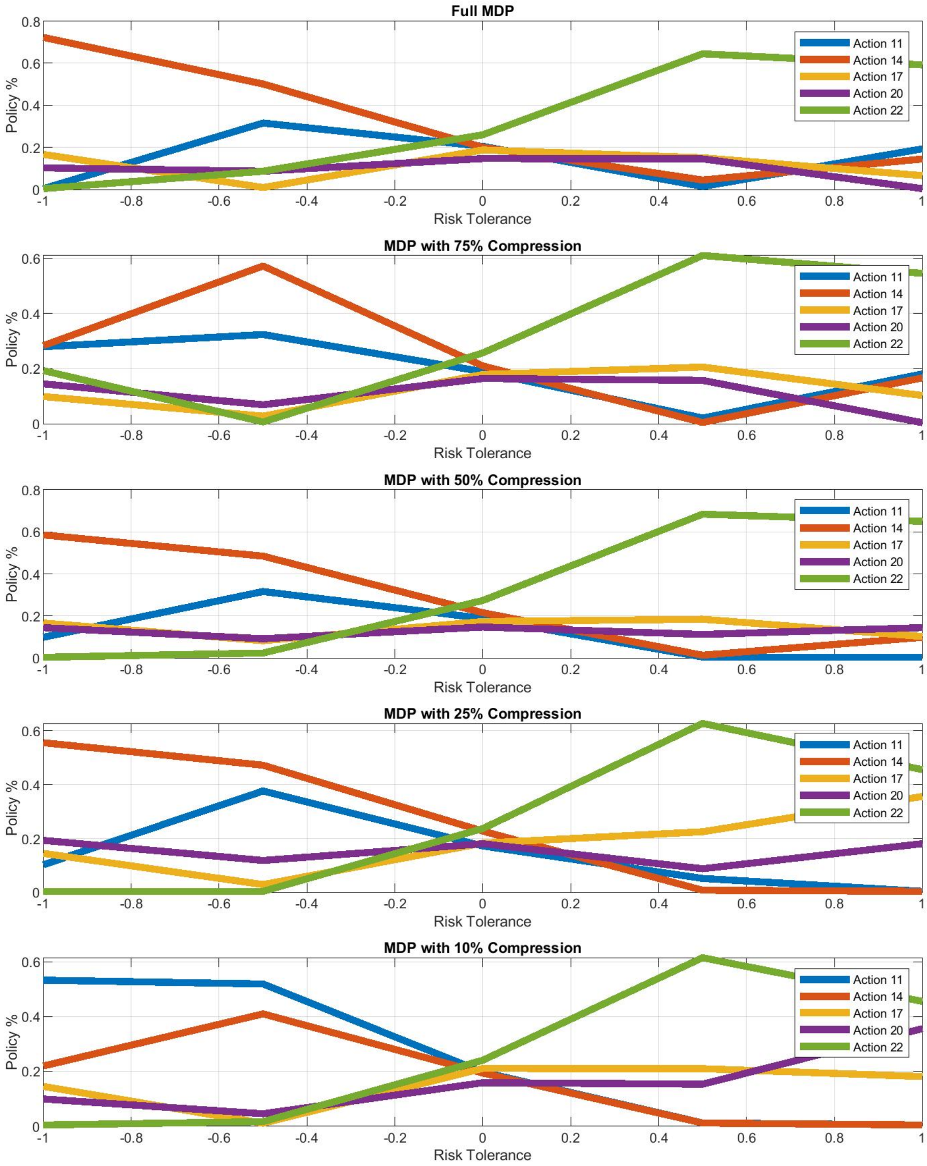


Figure 7.101: Experiment Set 2c: Stakeholder 1 Time Step $t = 2$ State of Interest RTSP versus State Compression Ratio

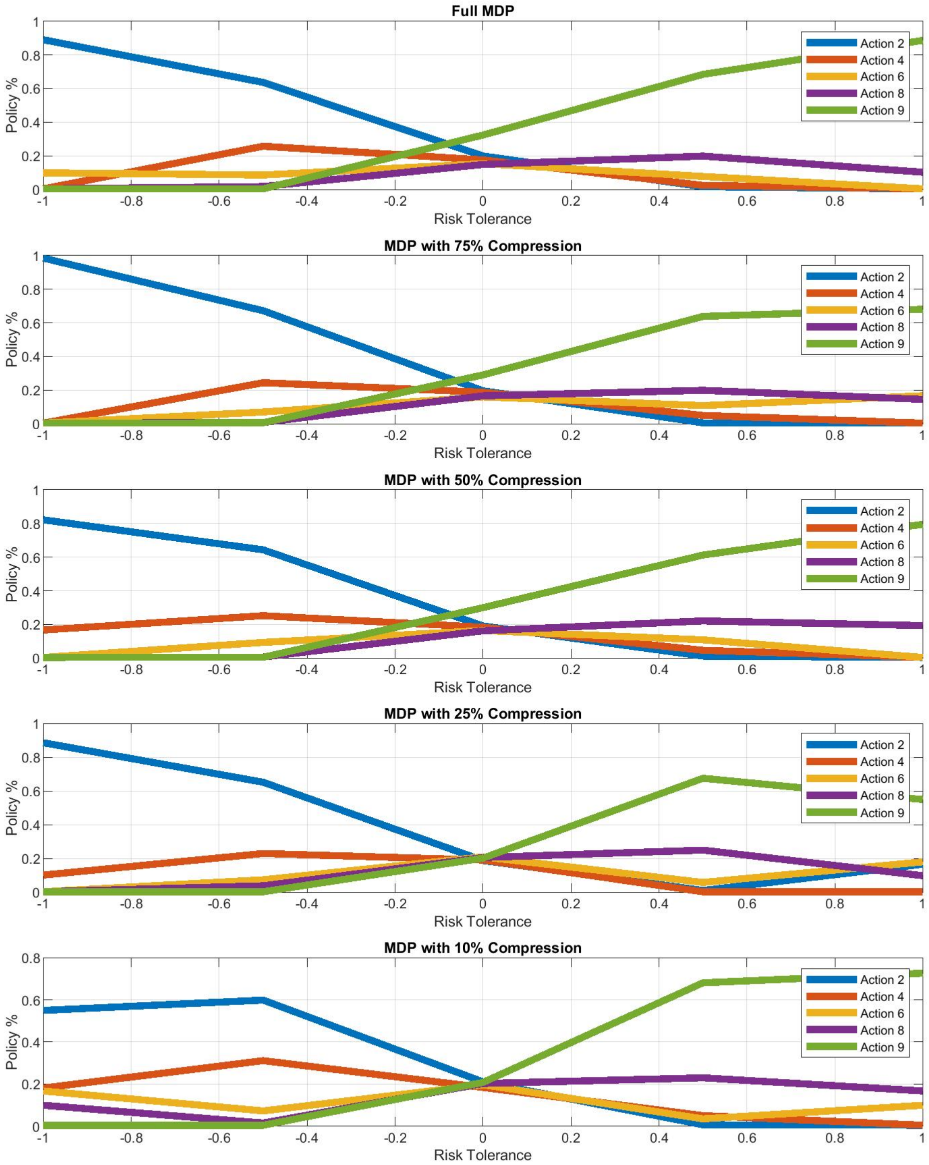


Figure 7.102: Experiment Set 2c: Stakeholder 1 Time Step $t = 3$ State of Interest RTSP versus State Compression Ratio

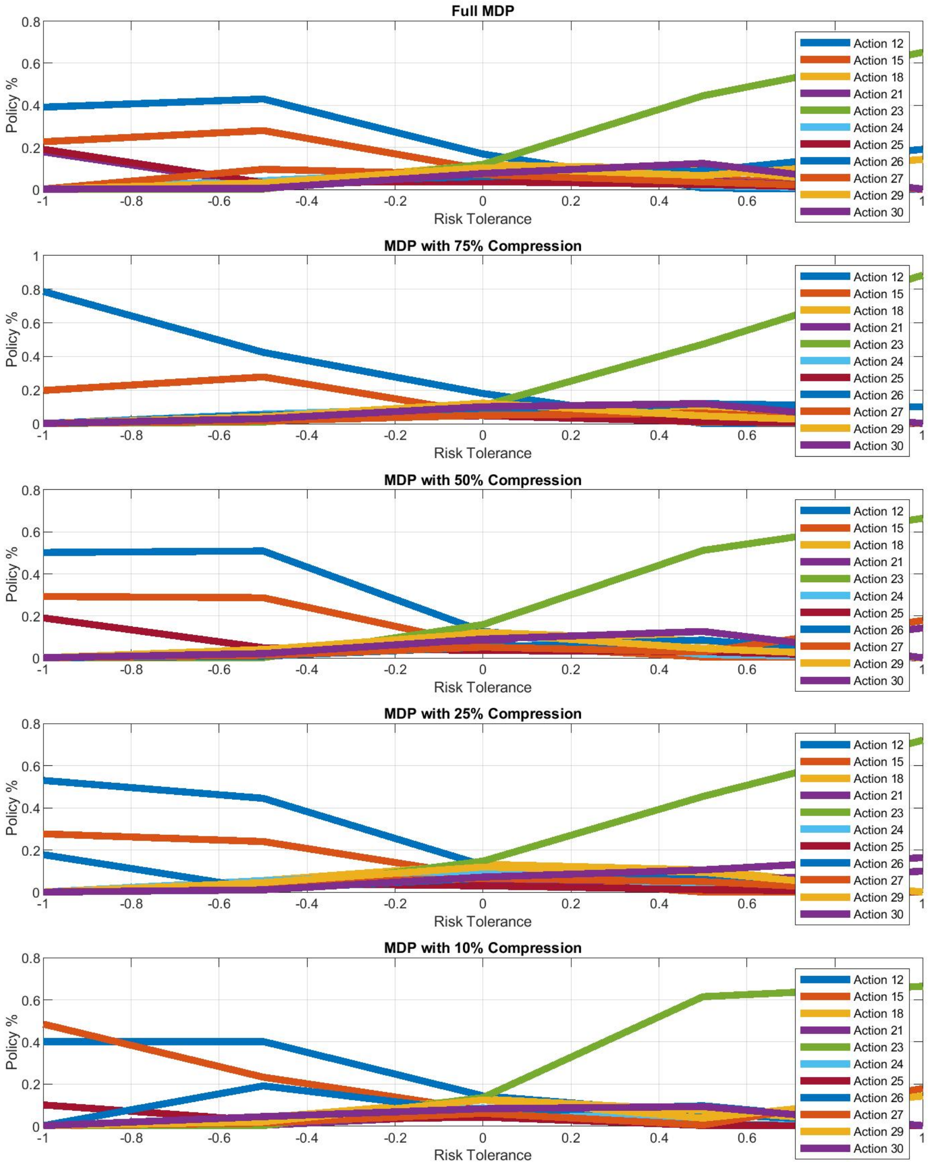


Figure 7.103: Experiment Set 2c: Stakeholder 1 Time Step $t = 4$ State of Interest RTSP versus State Compression Ratio

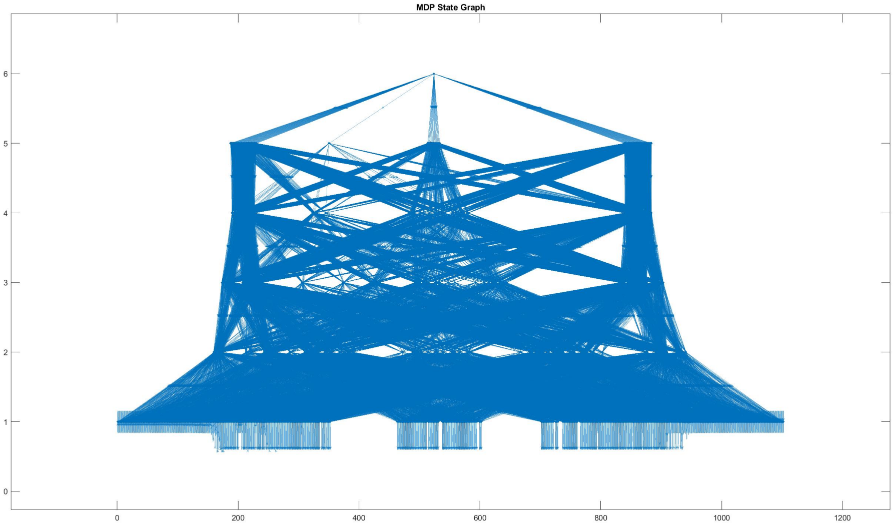


Figure 7.104: Experiment Set 2c: Stakeholder 2 Full MDP Graph with Highlighted States of Interest

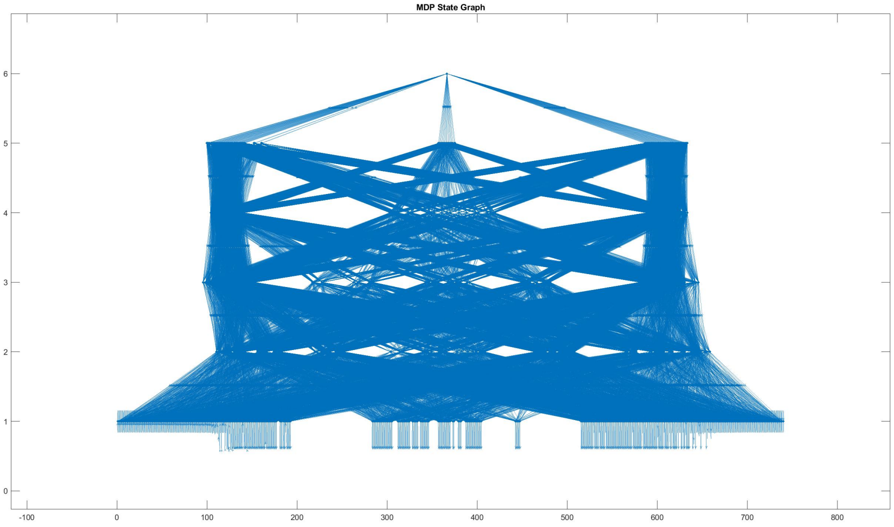


Figure 7.105: Experiment Set 2c: Stakeholder 2 75% State Compressed MDP Graph

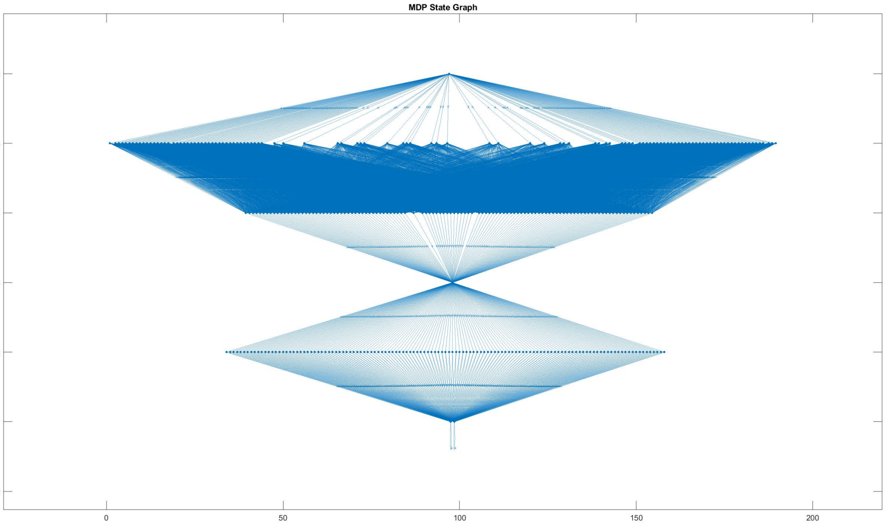


Figure 7.106: Experiment Set 2c: Stakeholder 2 25% State Compressed MDP Graph

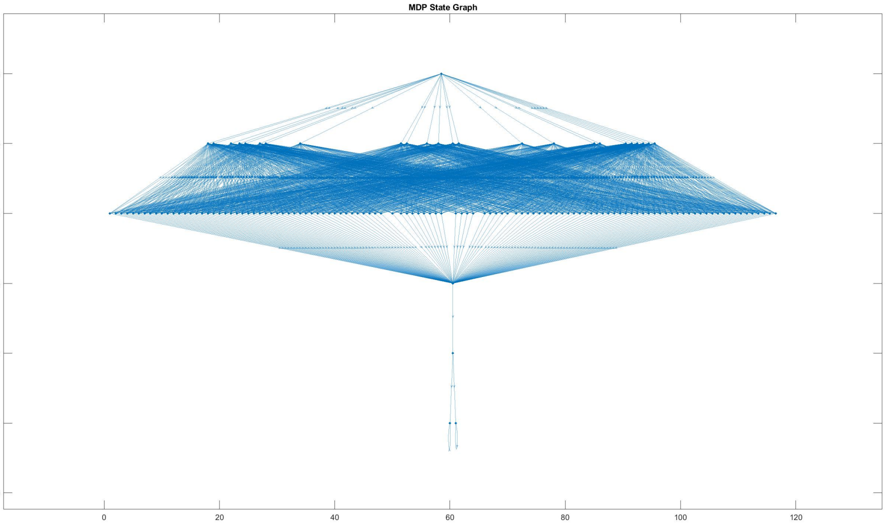


Figure 7.107: Experiment Set 2c: Stakeholder 2 10% State Compressed MDP Graph

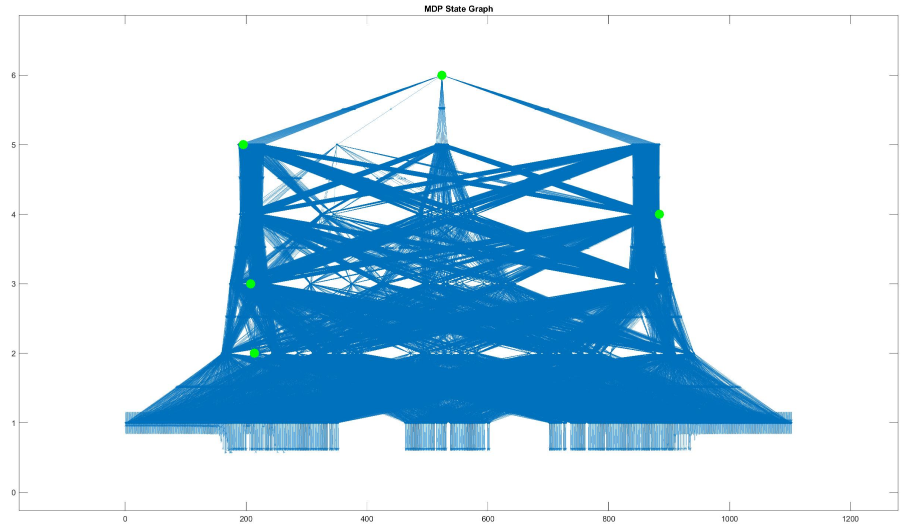


Figure 7.108: Experiment Set 2c: Stakeholder 2 Selected States for Evaluation

## 7.3 Experiment Set 3: Generating Insights from Derived Information

### 7.3.1 Experiment Set 3a: Lower Complexity Problems

Experiment Set 3a builds on the results of Experiment 1b. The evaluation using the risk-policy based algorithm and state-action metrics are analyzed to produce guidance for individual stakeholders. The selected results from Experiment 1b are used as exemplars to demonstrate the information provided via the optimal policy versus the methodology.

*Acquire Only Case*

Experiment 1b Case 1 examines the case where a stakeholder only has the option to acquire systems designed to provide a Pareto frontier decision from the Return mean-variance perspective. The optimal strategy that will develop to select the most risky action at every opportunity. For Scenario 1, the no inefficient action case, will result in always desiring to acquire System 3. The risk-based policy algorithm and the RTSPs allowed the potential actions to be categorized into action never to be taken, low risk actions, high risk actions, and

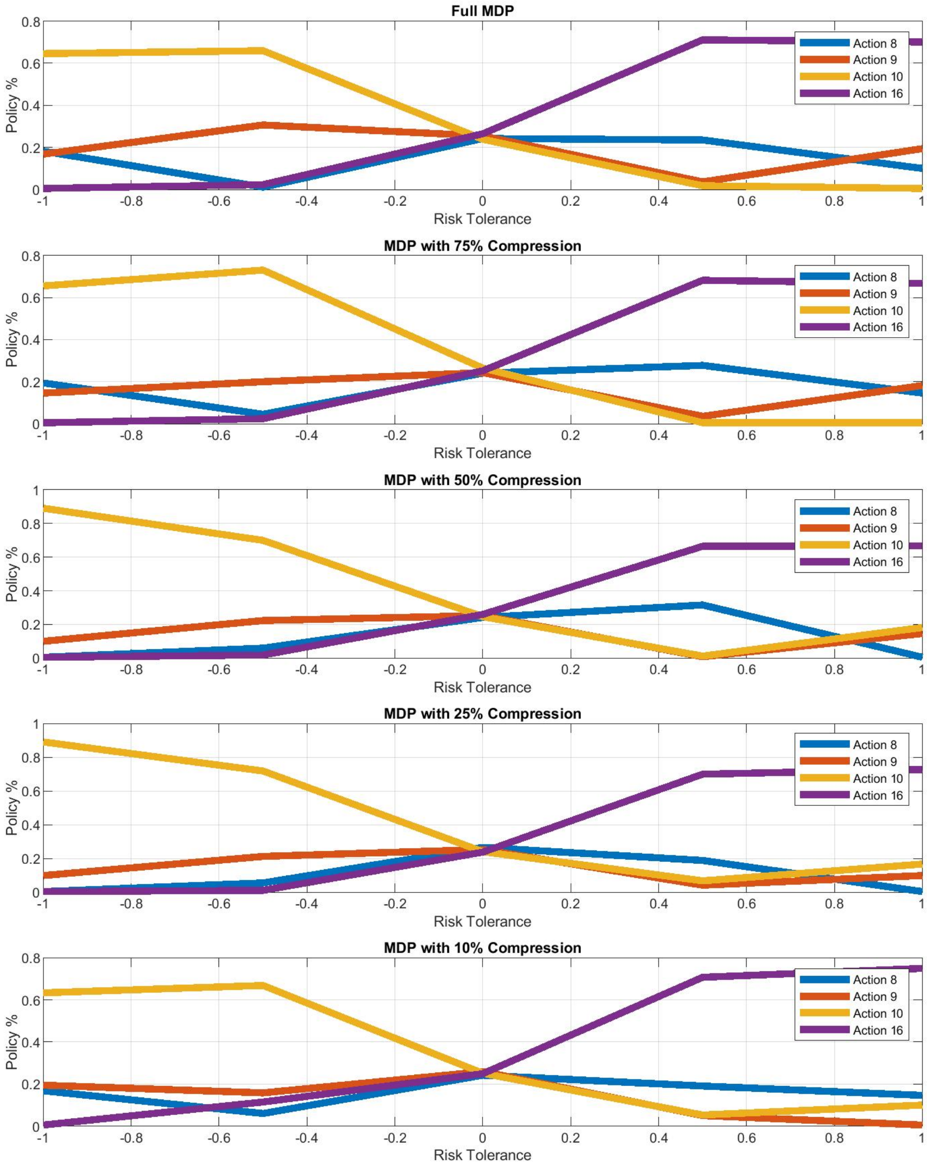


Figure 7.109: Experiment Set 2c: Stakeholder 2 Time Step $t = 0$ State of Interest RTSP versus State Compression Ratio

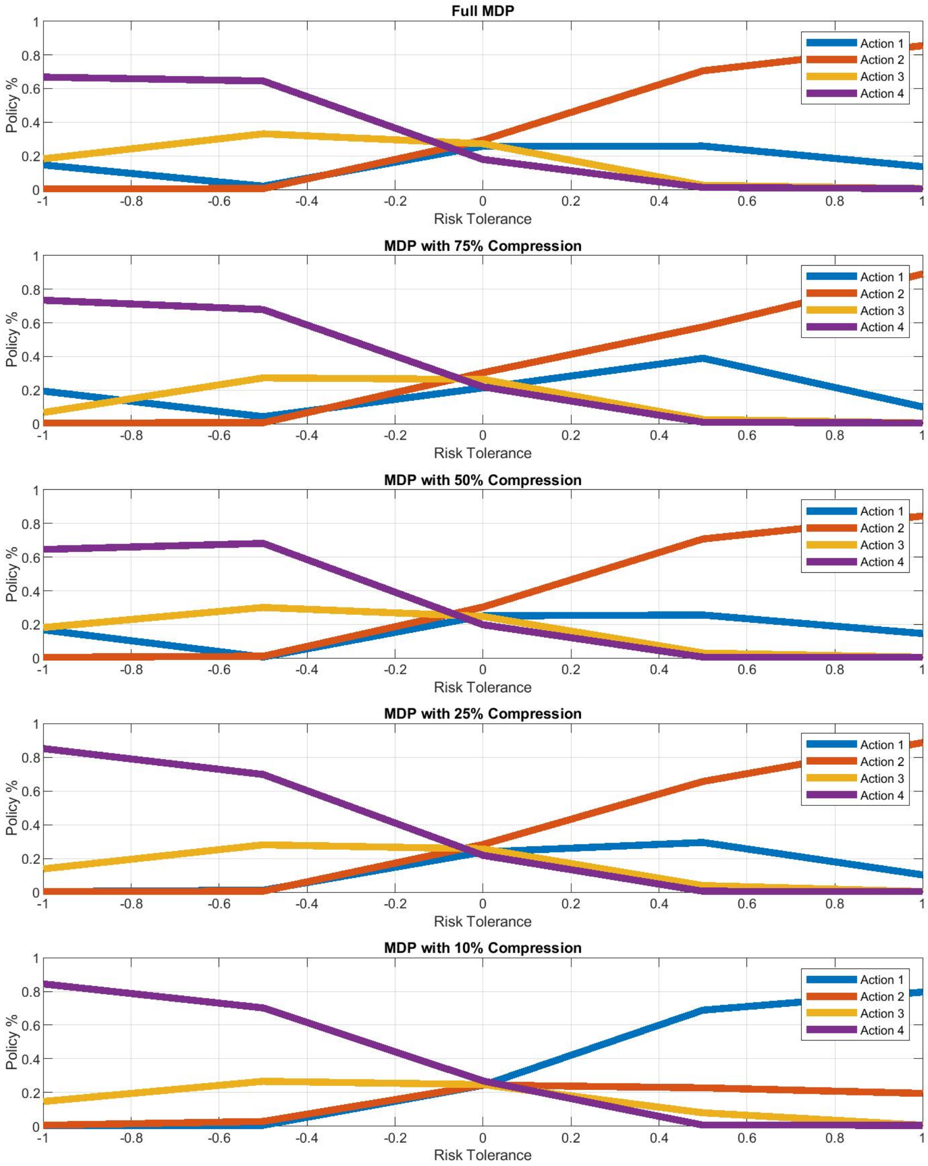


Figure 7.110: Experiment Set 2c: Stakeholder 2 Time Step $t = 1$ State of Interest RTSP versus State Compression Ratio

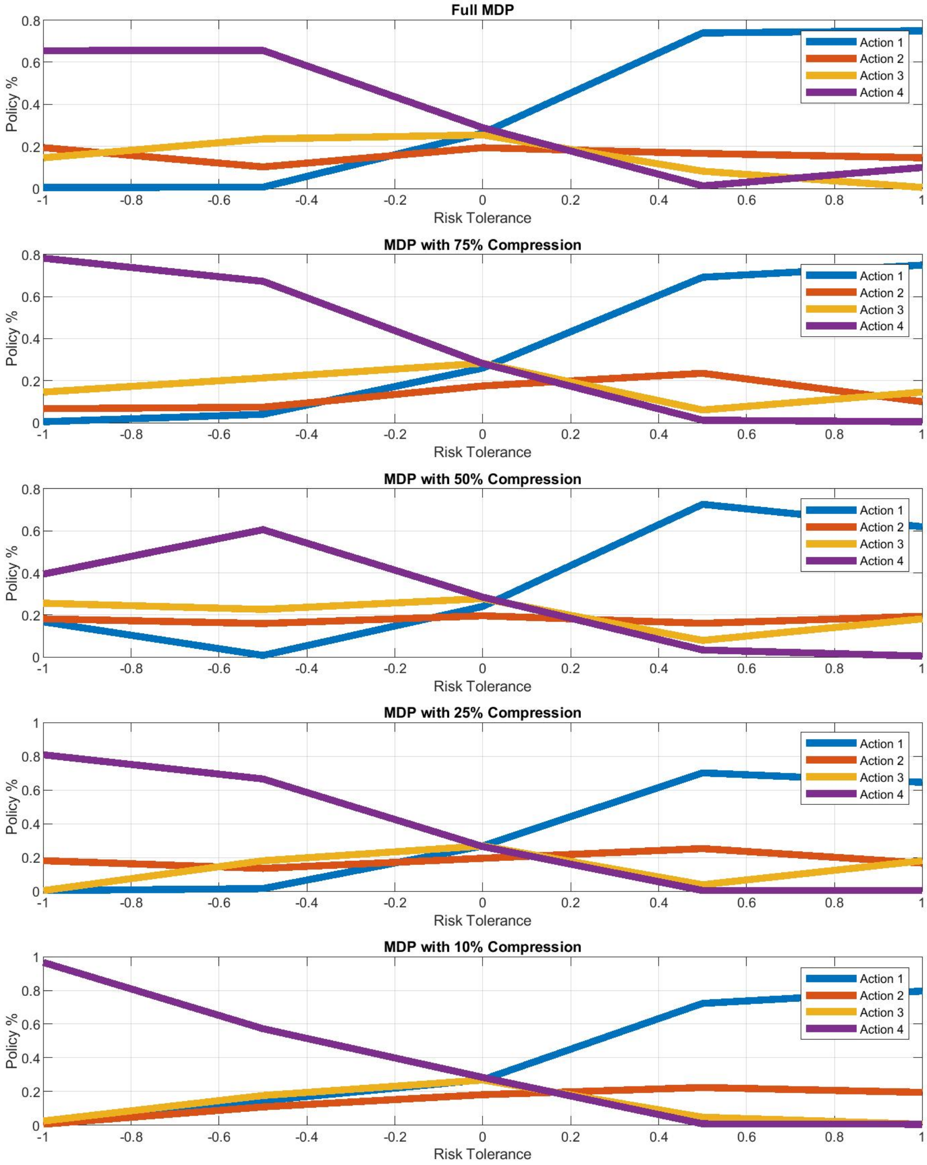


Figure 7.111: Experiment Set 2c: Stakeholder 2 Time Step $t = 2$ State of Interest RTSP versus State Compression Ratio

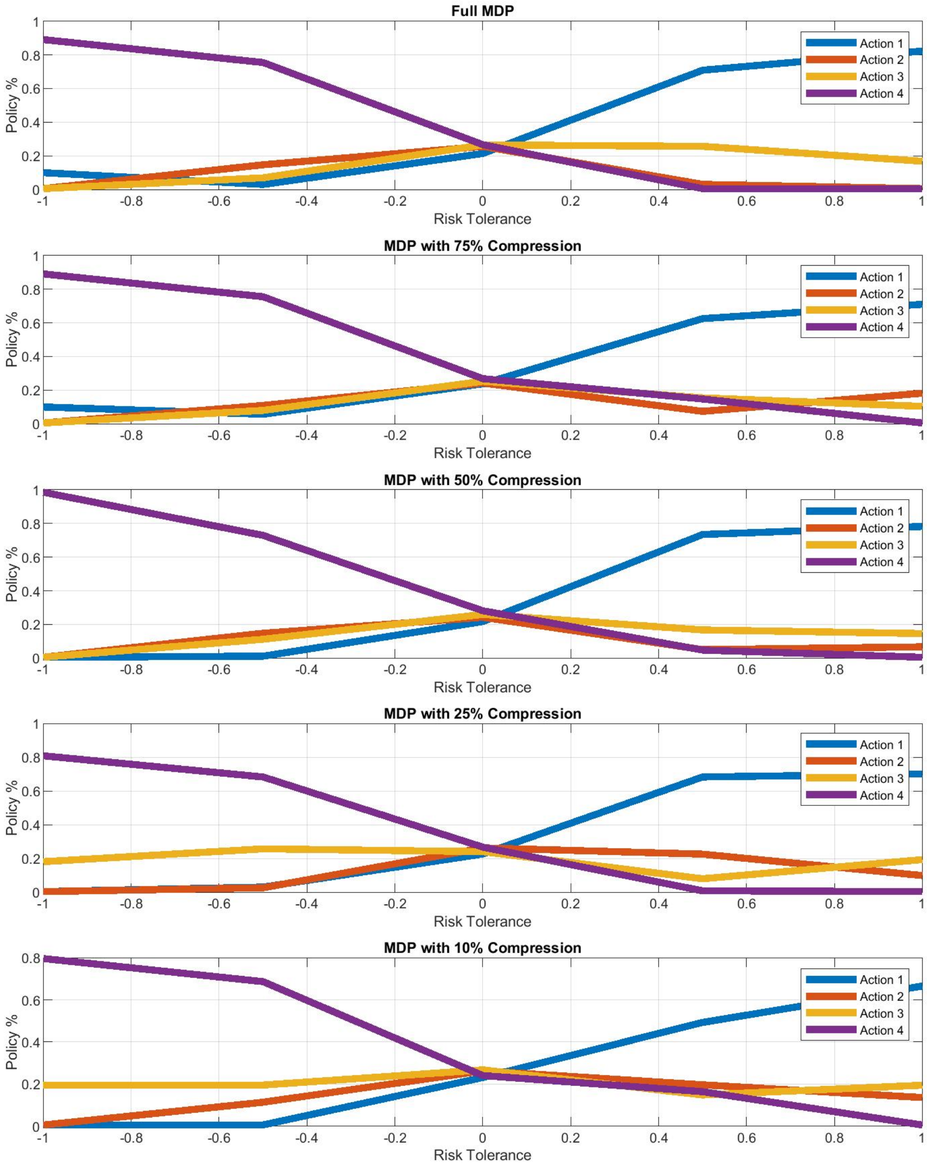


Figure 7.112: Experiment Set 2c: Stakeholder 2 Time Step $t = 3$ State of Interest RTSP versus State Compression Ratio

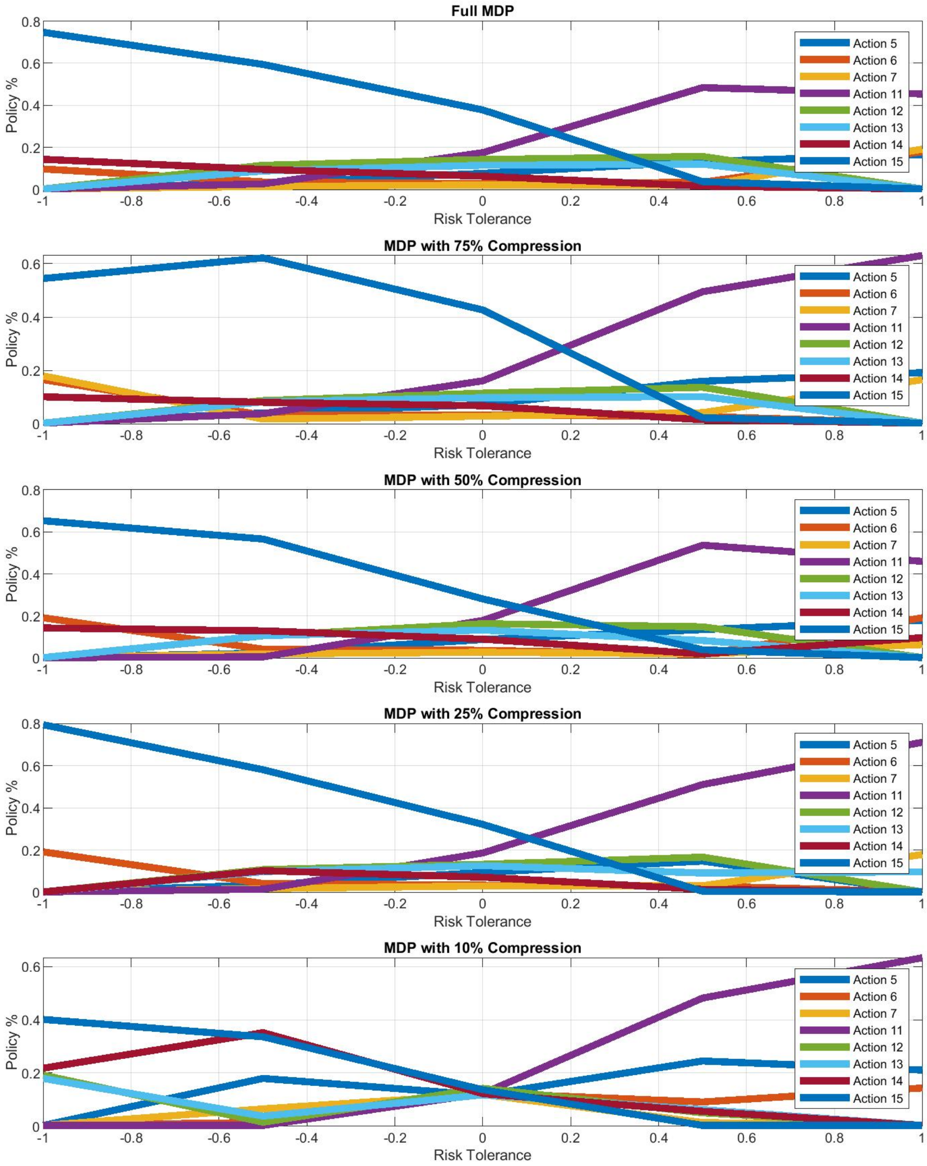


Figure 7.113: Experiment Set 2c: Stakeholder 2 Time Step $t = 4$ State of Interest RTSP versus State Compression Ratio

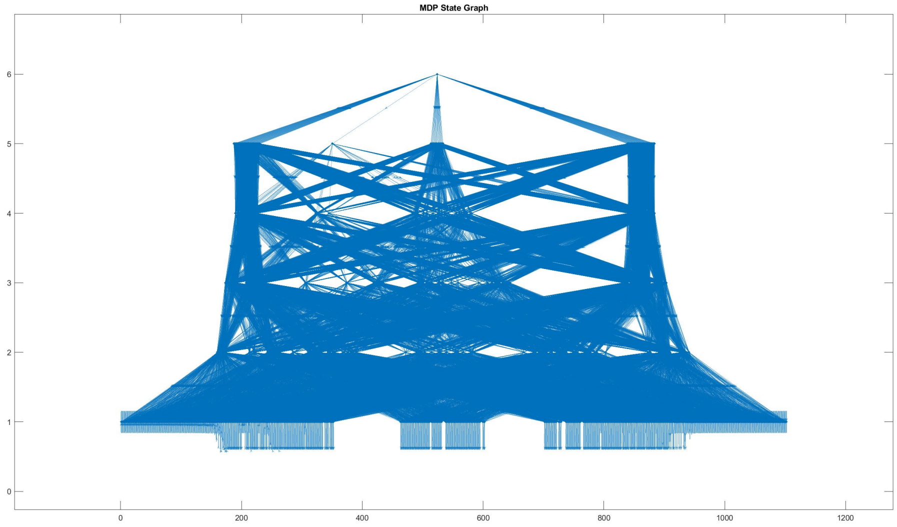


Figure 7.114: Experiment Set 2c: Stakeholder 3 Full MDP Graph with Highlighted States of Interest

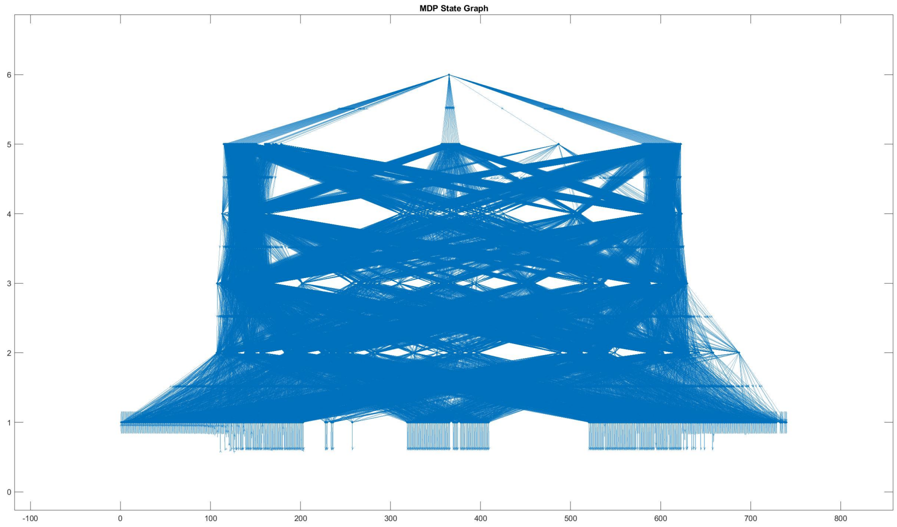


Figure 7.115: Experiment Set 2c: Stakeholder 3 75% State Compressed MDP Graph

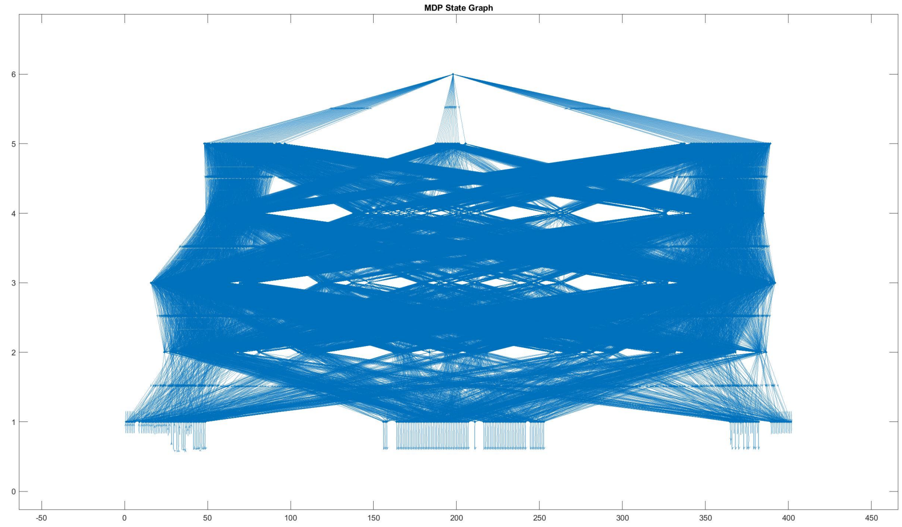


Figure 7.116: Experiment Set 2c: Stakeholder 3 50% State Compressed MDP Graph

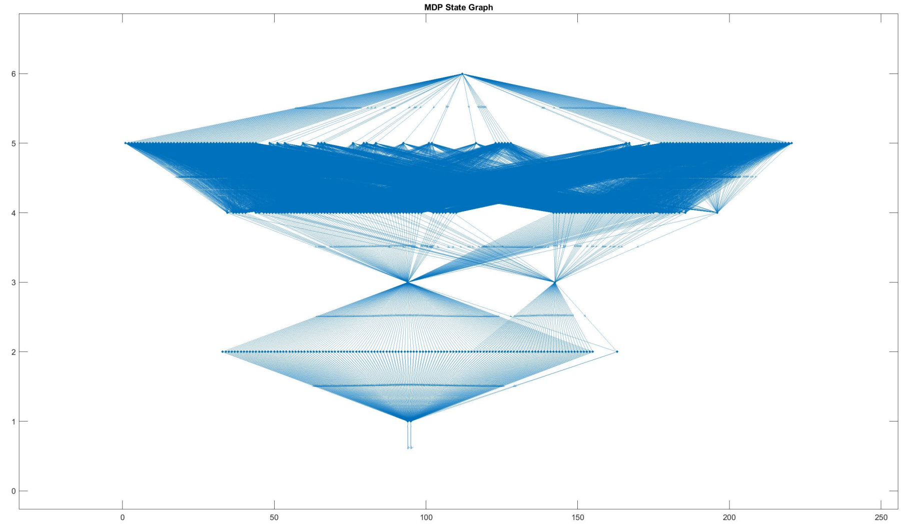


Figure 7.117: Experiment Set 2c: Stakeholder 3 25% State Compressed MDP Graph

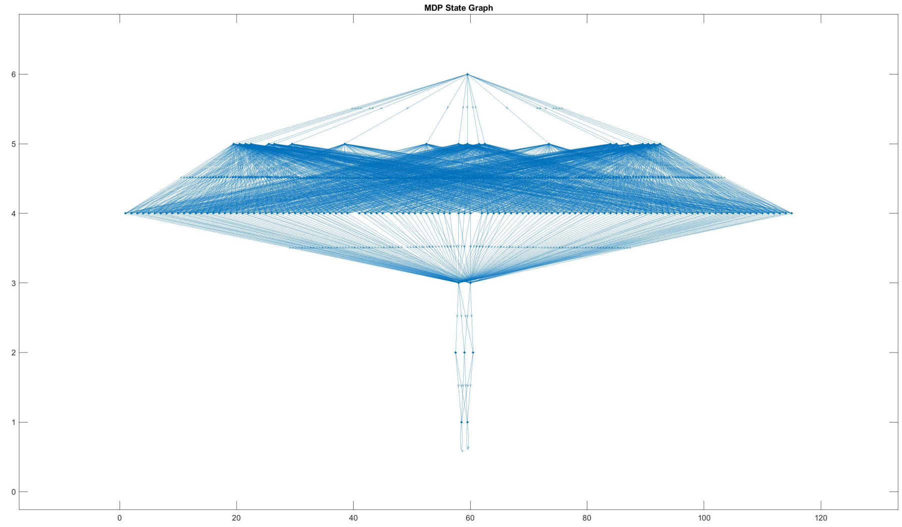


Figure 7.118: Experiment Set 2c: Stakeholder 3 10% State Compressed MDP Graph

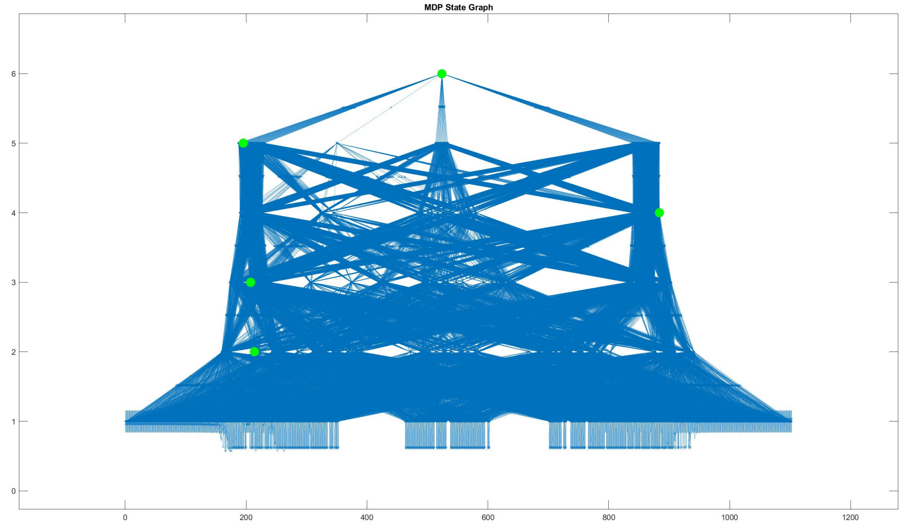


Figure 7.119: Experiment Set 2c: Stakeholder 3 Selected States for Evaluation

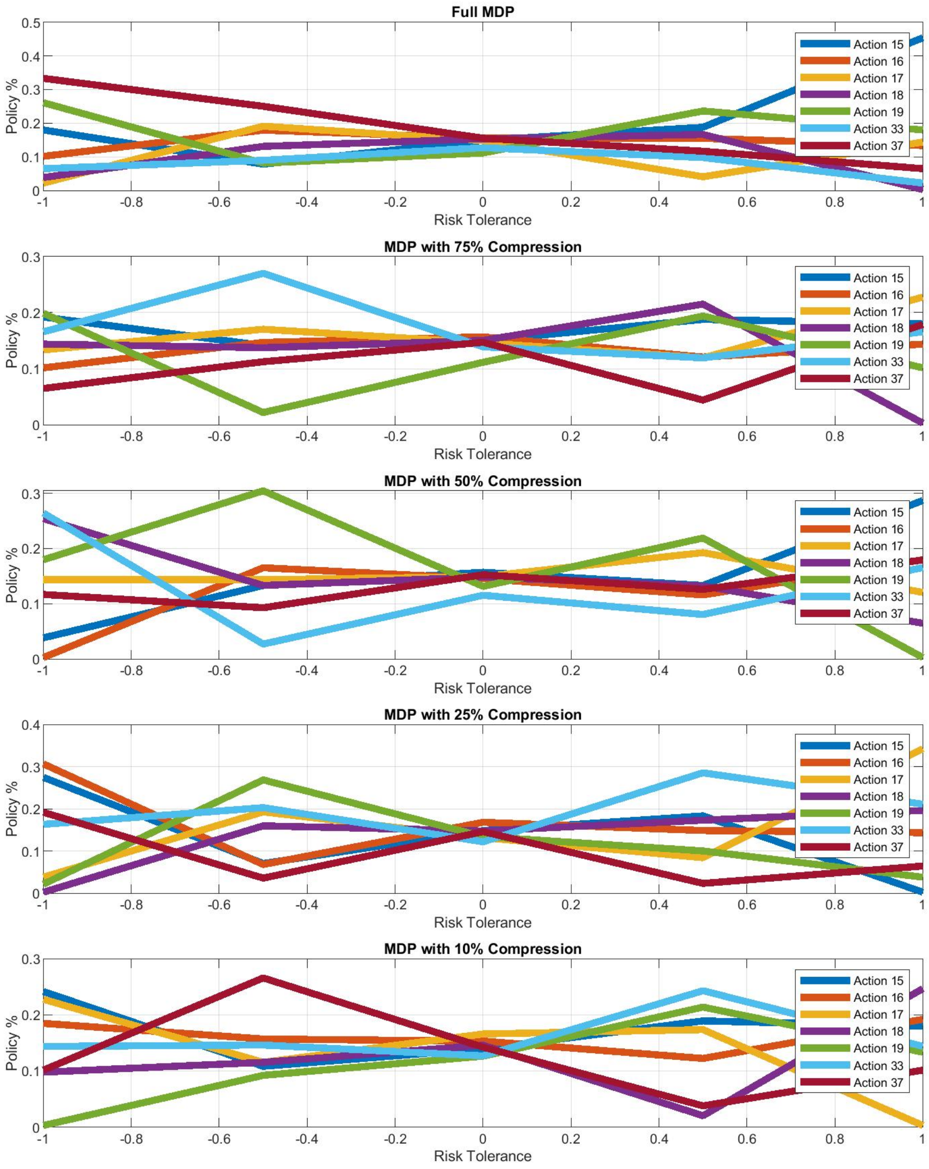


Figure 7.120: Experiment Set 2c: Stakeholder 3 Time Step $t = 0$ State of Interest RTSP versus State Compression Ratio

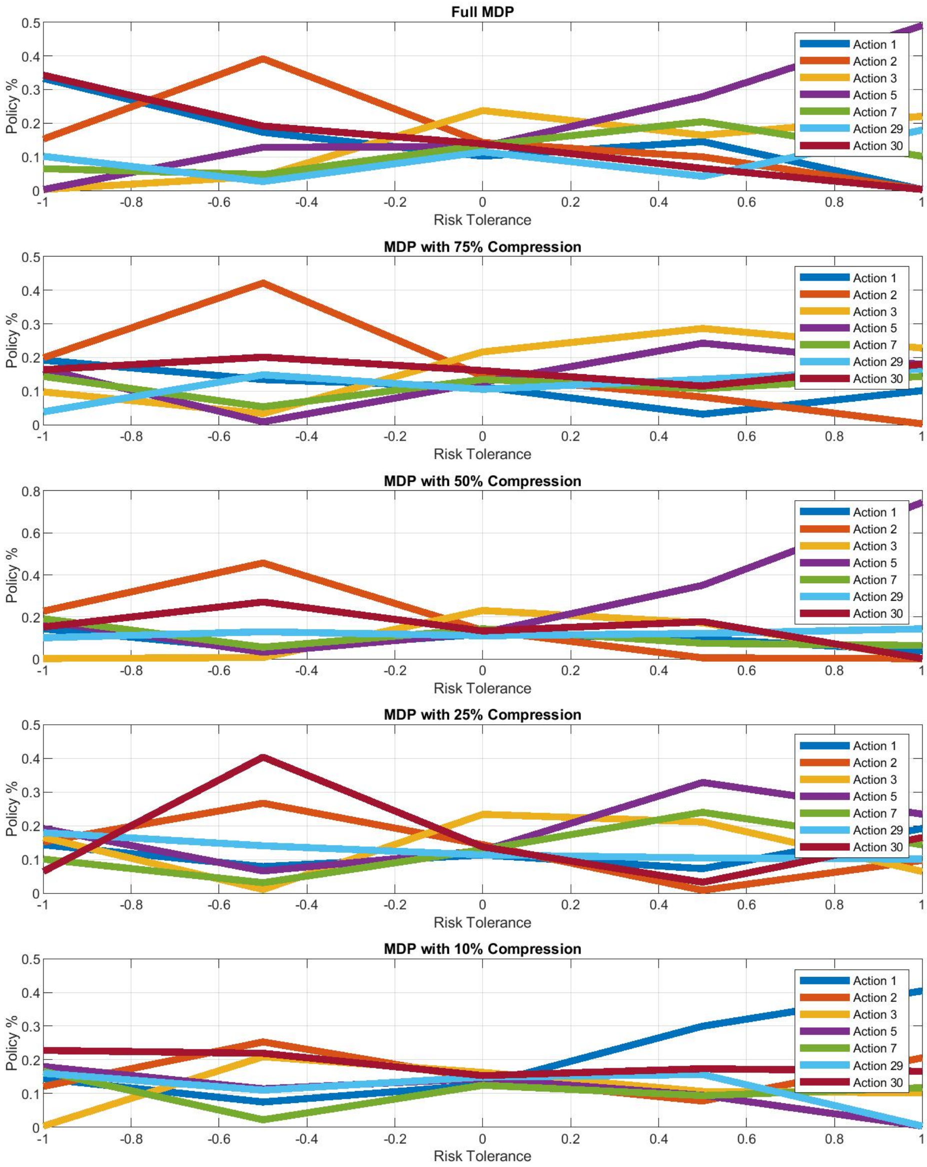


Figure 7.121: Experiment Set 2c: Stakeholder 3 Time Step $t = 1$ State of Interest RTSP versus State Compression Ratio

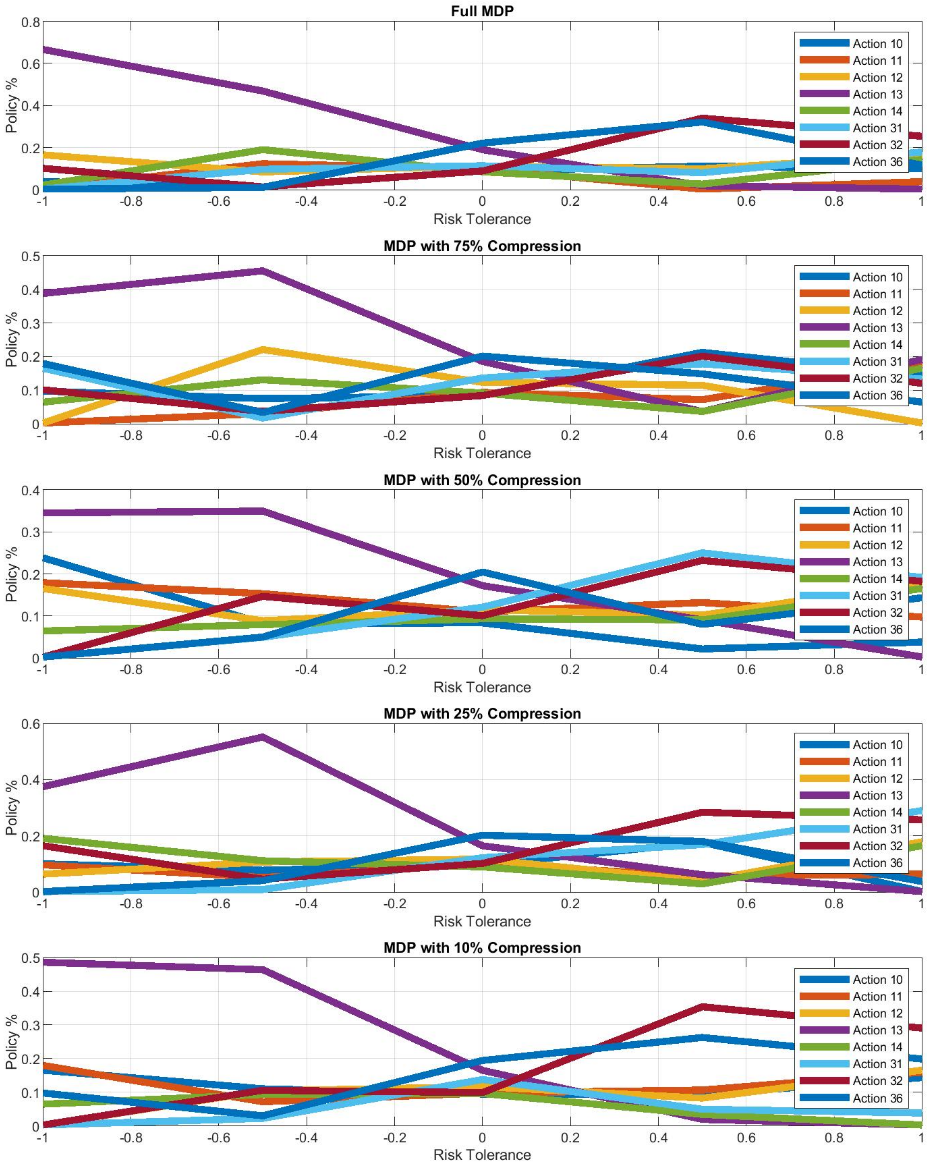


Figure 7.122: Experiment Set 2c: Stakeholder 3 Time Step $t = 2$ State of Interest RTSP versus State Compression Ratio

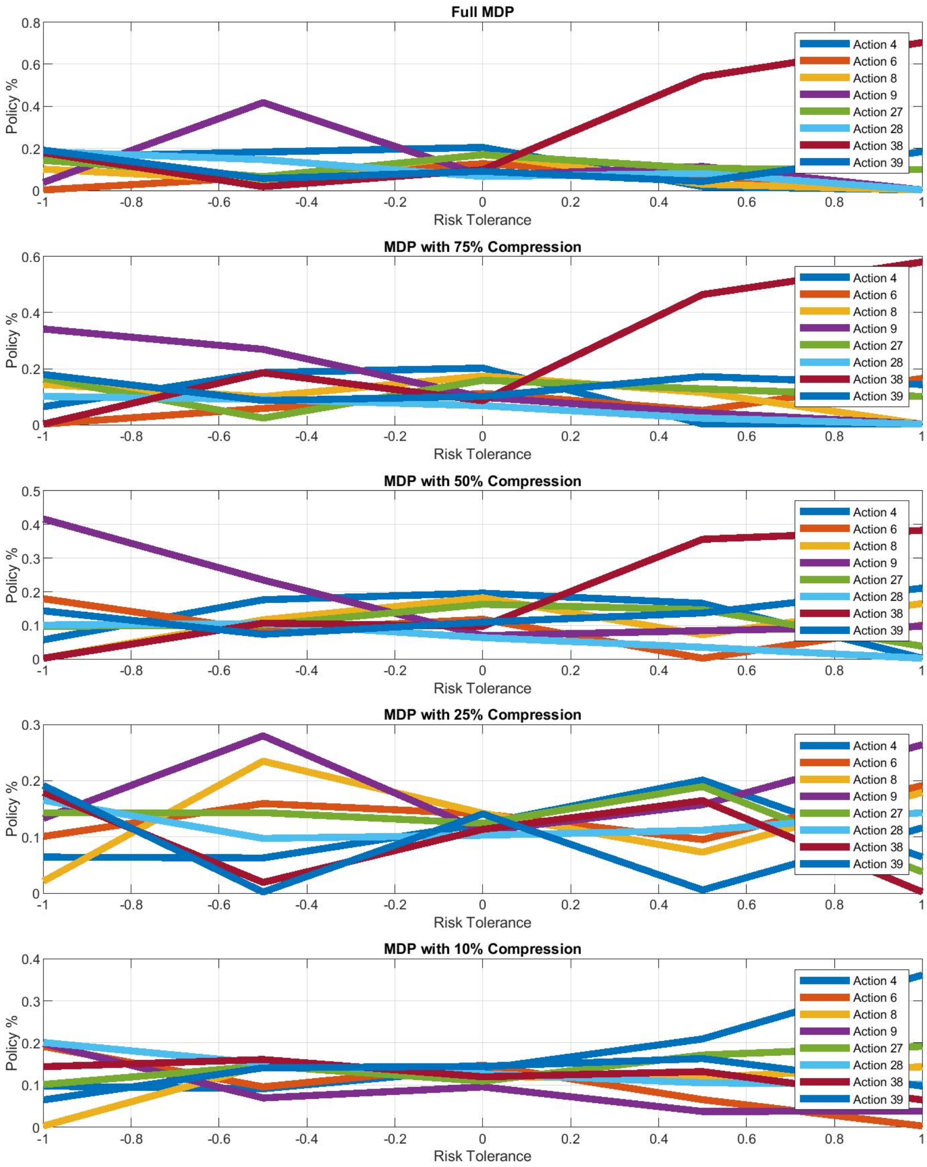


Figure 7.123: Experiment Set 2c: Stakeholder 3 Time Step $t = 3$ State of Interest RTSP versus State Compression Ratio

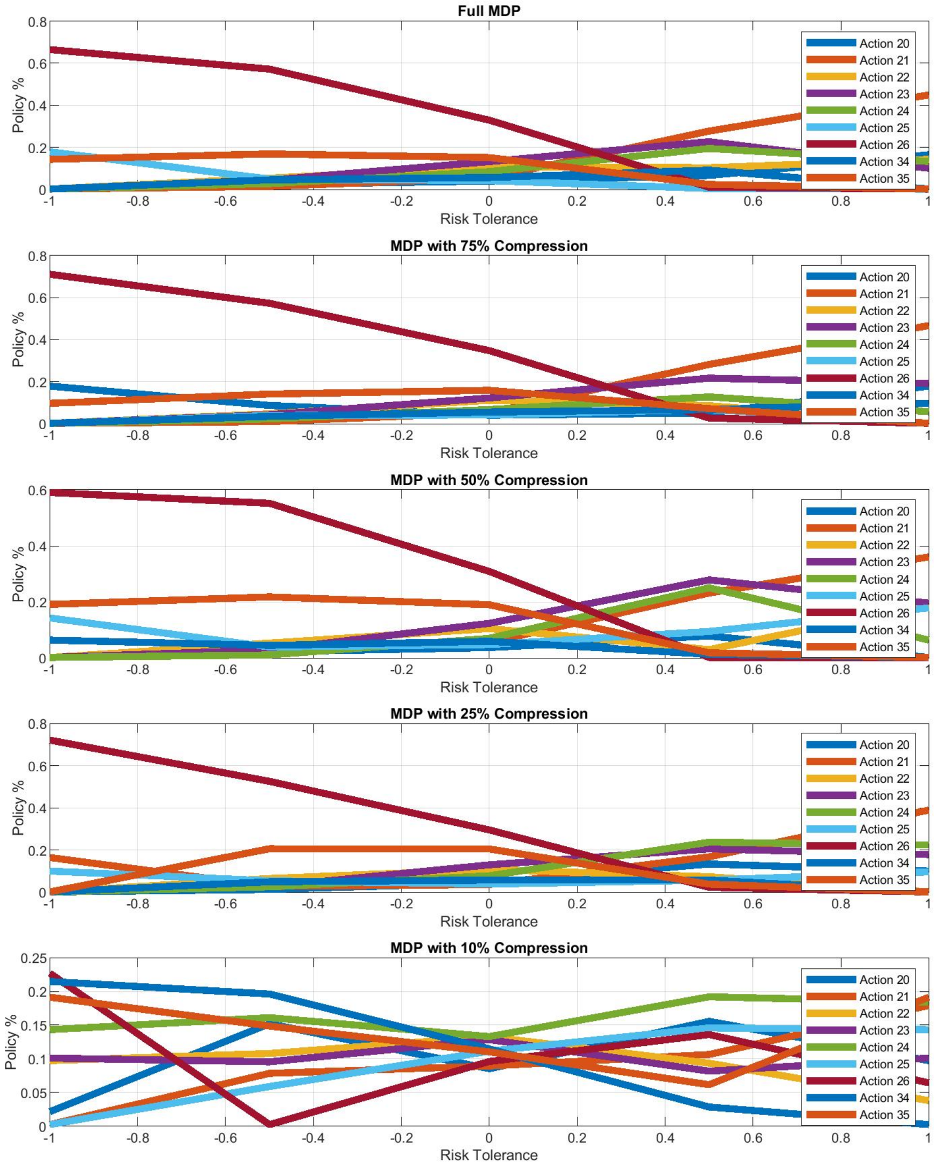
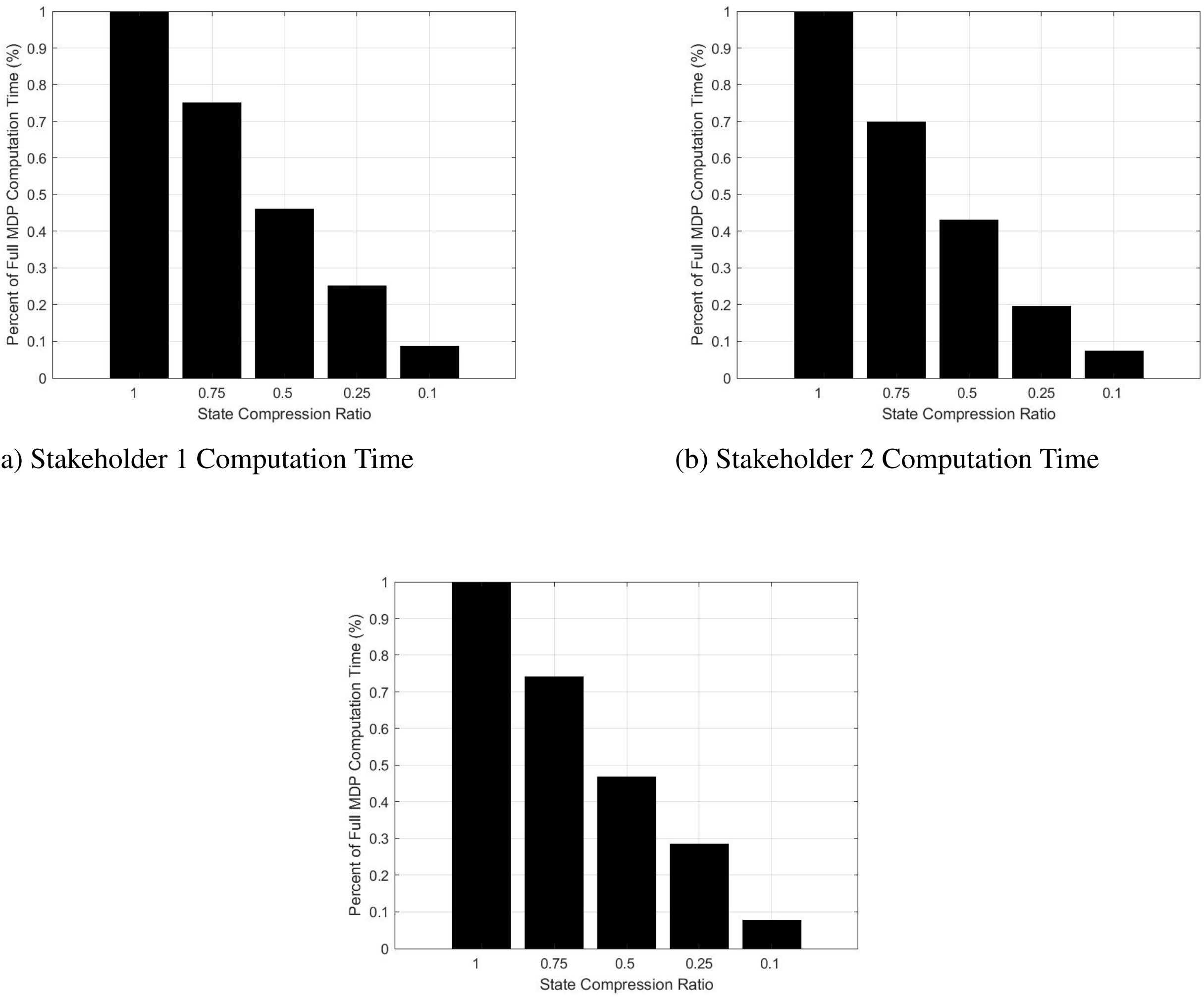


Figure 7.124: Experiment Set 2c: Stakeholder 3 Time Step $t = 4$ State of Interest RTSP versus State Compression Ratio



(a) Stakeholder 1 Computation Time

(b) Stakeholder 2 Computation Time

(c) Stakeholder 3 Computation Time

Figure 7.125: Experiment Set 2c: Computation Time

Pareto inefficient actions. This first and simple study case has demonstrated the ability of the methodology to produce more information than an optimal policy method. A more in-depth analysis under following more complex scenarios fully demonstrates the capabilities of the methodology.

### *Acquire or Develop Case*

The optimal strategy results in a single action selection per state. The recommendation derived from the optimal policy for the selected analysis states are:

- In State 1, choose to Acquire System 1.
- In State 13, choose to Acquire System 2.

A greater amount of information, more specific and nuanced, is available for the stakeholder when using the methodology. The information is derived from the methodology outputs, Risk-Tolerance Sensitivity profiles and the state-action metrics. The following information is derived from the Experiment 1b Case 2 results:

- In State 1, the stakeholder should select to Acquire System 1 over Developing System 2.
- In State 13, the stakeholder should never choose to Develop System 3.
- In State 13, the stakeholder should select Acquiring System 1 if they are risk adverse.
- In State 13, the stakeholder should select Acquiring System 2 if they are more risk tolerant.
- When selecting between Acquiring System 1 and Developing System 2, the stakeholder should always choose to Acquire System 1.
- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should never choose to Develop System 3.

- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should select Acquiring System 1 if they are risk adverse.
- If System 2 is developed and the stakeholder is selecting between Acquiring System 1, Acquiring System 2, and Developing System 3 the stakeholder should select Acquiring System 2 if they are more risk tolerant.

*Acquire, Develop, and/or Allocate Case*

Optimal strategies applied to the actions spaces analyzed in Experiment 1b Case 2 produce the following guidance:

- When Stakeholder 1 is allocating three System 1 only, allocate all available System 1 to Mission 1.
- When Stakeholder 1 is allocating four System 1 only, allocate all available System 1 to Mission 1.
- When Stakeholder 2 is allocating five System 1 and Developing System 2 or Acquiring System 1 the optimal selection varies based on state.
- When Stakeholder 3 is selecting allocations of System 9, the preferred action is random.

The methodology produces the following guidance to Stakeholders:

- When Stakeholder 1 is allocating three System 1 only,
  - never allocate all System 1 to Mission 2
  - unadvised to allocate more than one System 1 to Mission 2
  - allocating all System 1 to Mission 1 to is high risk and high reward
  - allocating a single System 1 to Mission 2 and two to Mission 1 is lower risk

- When Stakeholder 1 is allocating four System 1 only,
    - never allocate all System 1 to Mission 2
    - unadvised to allocate more than two System 1 to Mission 2
    - allocating all System 1 to Mission 1 to is high risk and high reward
    - allocating a single System 1 to Mission 2 and three to Mission 1 is lower risk
- When Stakeholder 1 is allocating five System 1, Acquire System 1 or Develop System 2,
    - unadvised to allocate nearly all System 1 to Mission 2
    - allocating nearly all System 1 to Mission 1 to is high risk and high reward
    - allocating a some System 1 to Mission 2 and more to Mission 1 is lower risk
- When Stakeholder 2 is allocating five System 5, Acquire System 7 or (Acquire System 5 and Develop System 6)
    - unadvised to allocate nearly all System 1 to Mission 2
    - allocating nearly all System 1 to Mission 1 to is high risk and high reward
    - allocating a some System 1 to Mission 2 and more to Mission 1 is lower risk
    - the time horizon provided does not allow the impacts of acquisition and development
- When Stakeholder 3 is selecting allocations of System 9 there is no preference as a function of risk-tolerance due to the equal contribution of Mission 1 and Mission 2 to Stakeholder 3 utility.

### 7.3.2 Experiment Set 3b: Full Complexity Problem

The full complexity test problem is designed to evaluate Hypothesis 3 while fully exercising and benchmarking the methodology. The Truth Model setup which characterizes the experiment is described in Section 6.3.2. Each step of the methodology is executed against the described setup and addressed in order below.

*Step 1: Generating the Meta-Model*

The first step of the methodology is generating the meta-model for evaluation. The Truth model must be sampled and the meta-model constructed. One thousand MC samples of the Truth Model were made to construct the meta-model. With a $20$ time step maximum, the sampling resulted in $21,000$ discrete state-action-reward-state ($s-a-r-s$) samples across time steps. The unique number of states and actions found as a function of MC samples is depicted in Figure 7.126.

Note that the number of states has not yet plateaued while the number of action has begun to plateau. Previous sampling has worked to ensure that the total number of states and action samples has asymptotically reached a constant value and remain constant for additional sampling. Repeated samples for each $s-a-r-s$ can be ensured by not stopping sampling until well after a steady state number of unique states and actions is produced. Additionally, the total sample size of each identified $s - a - r - s$ could be specified and overall MC sampling continued until it is met. This would ensure that every state-action pair has a sufficient representation statistically. The cost is a significant oversampling of early states and actions. The alternative is to evaluate the sampled states by time and ensure that sampling at states of interest are of a sufficient number. Figure 7.127 represents the number of unique samples by time step. Sufficient sampling for a given time step by state can be ensured as long as the number of unique state samples is significantly less than the number of total samples. Any time steps with the number of unique states close to the number of samples shows a time step that is under sampled. These time steps that are under sampled contribute to earlier time step metrics but will not provide sufficient information for future time steps. For the Experiment 3b, any time step greater than 3 will be regarded as supporting states of earlier time steps only.

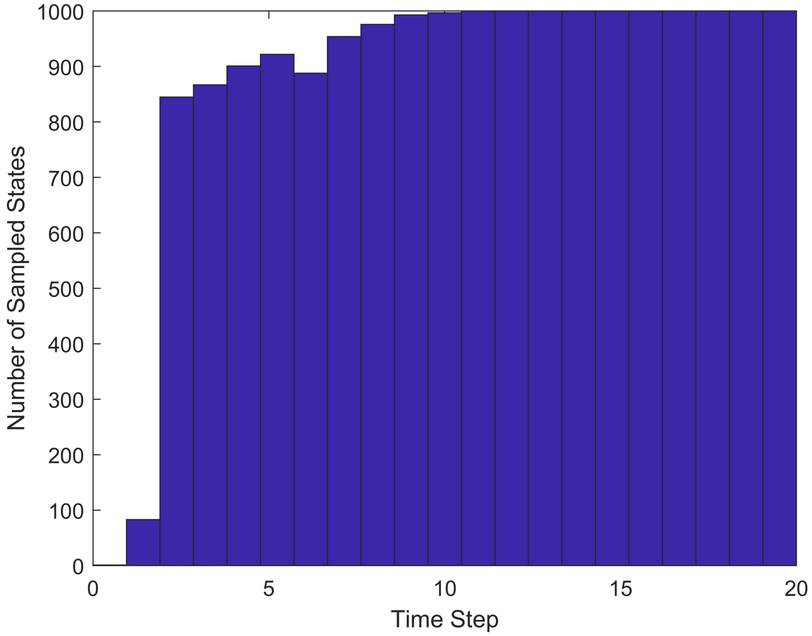


Figure 7.126: Experiment 3b: Monte Carlo Truth Model Sample Metrics

The sampling resulted in a full structure as depicted in Figure 7.128. Each node represents a single unique state and each edge represents a transition between states. The actions and rewards are not depicted as part of the structure diagram. A compression ratio of 35% was used to compress states at each time step with a bias for a minimum number of states per time step. The resulting structure is depicted in Figure 7.129. The total number of Actions is $1,026$. The total number of unique full states is $18,427$. The total number of unique meta-model states is $6,505$. The final component of sampling the meta-model is the resulting reward that the stakeholder received for each action made. The raw data for reward versus time for each MC sample is depicted in Figure 7.130. Not the difficulty in identifying patters and bifurcation points due to the large amount of uncertainty. The reward can also be viewed as a distribution as a function of time. The PDF as a function of time step yield a little more nuanced look at stakeholder reward but provide little more discernible information. A significant amount of information can be derived from evaluating the meta-model.

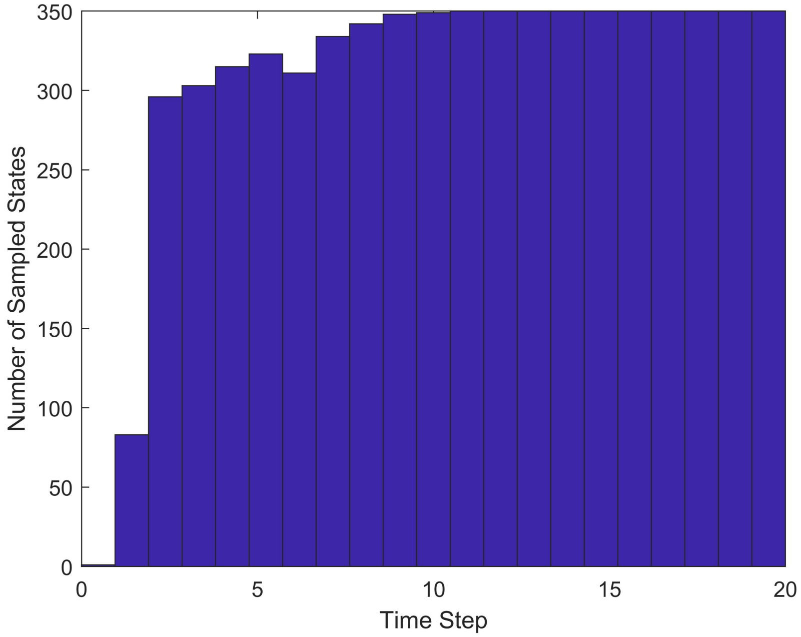


Figure 7.127: Experiment 3b: Time Step based Unique State Sample Metrics

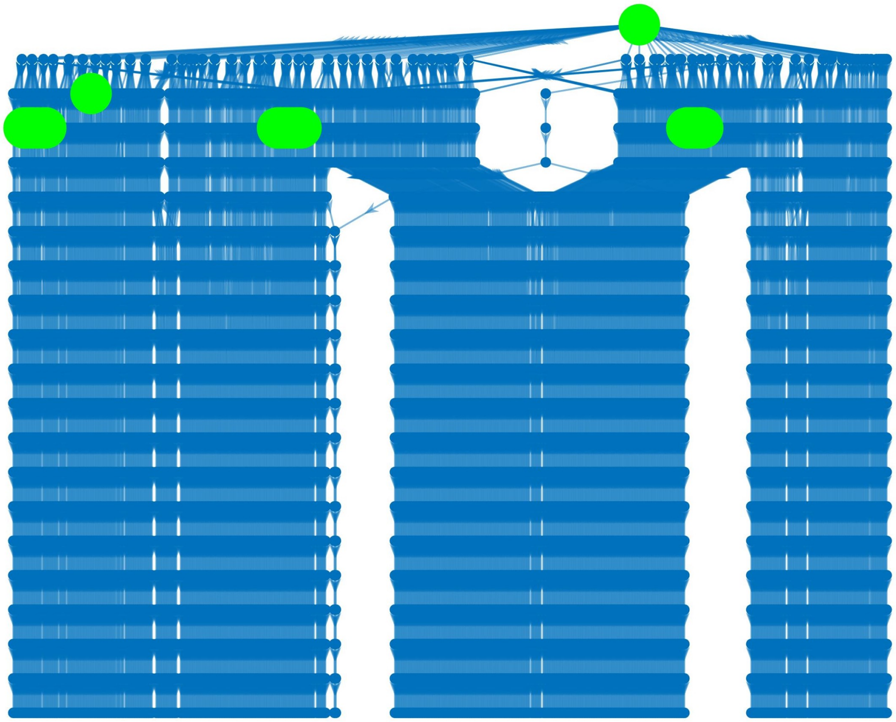


Figure 7.128: Experiment 3b: Full Structure

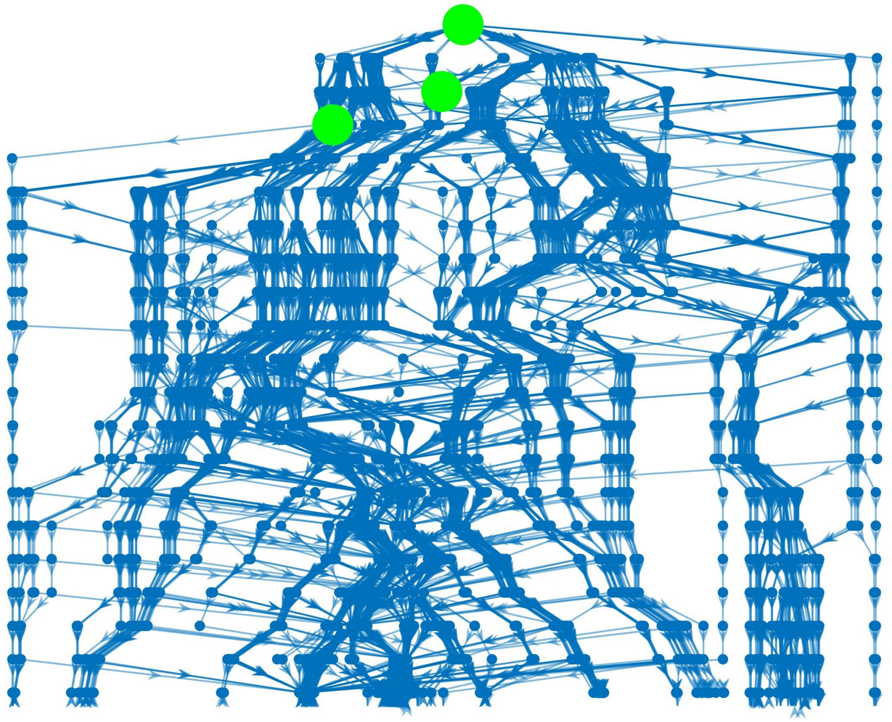


Figure 7.129: Experiment 3b: Meta-Model Structure

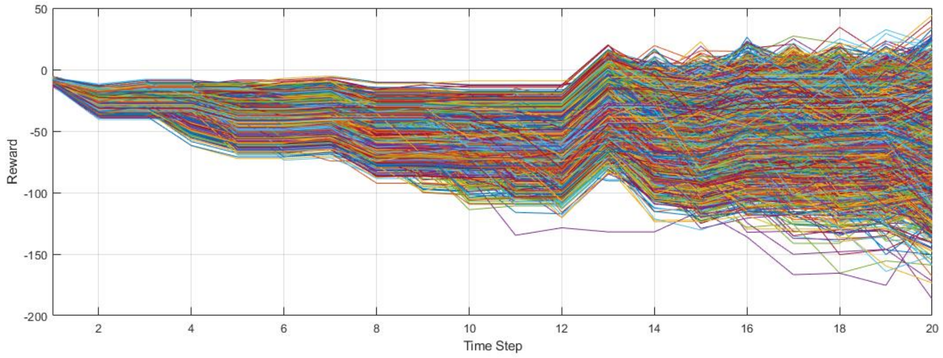


Figure 7.130: Experiment 3b: Stakeholder 3 Reward versus Time

*Step 2: Evaluating the Meta-Model*

The meta-model depicted in Section 7.3.2 was used to produce both risk-based policy and state-action metrics. Five risk-tolerance levels $(-1, -0.5, 0, .5, 1)$ were used to produce risk-based policies. A time horizon of 20 steps was used for each risk-tolerance level and state in the meta-model. More that $60,000,000$ of $s-a-r-s$ samples of the meta-model were gathered for each of the five risk-tolerance based policy. For each of $6,000+$ unique states, $500$ samples consisting of $20$ time steps were run.

Five risk-based policies were generated and used to develop RTSPs, entropy measurements, and Return maps. Three meta-model states and their corresponding full model states were identified as States of Interest (SOI). The meta-model SOIs collected for further evaluation meet a threshold of requirements of being a decision point (available actions greater than one) and having sufficient MC samples. The selected MDP states are highlighted in Figure 7.129. Similarly, the full MDP states are a subset of the full meta-model SOIs that continue to carry multiple actions after the decompression. The full state space SOIs are highlighted in Figure 7.128.

Three specific use cases have been selected for evaluation and investigation. The first use case is the most significant for Stakeholder 1, Germany. The first use case is the initial state for Germany (State 1). The initial state is when Germany is presented with the largest selection of potential actions and most complex future outcome. The second use case examines a future decision point branching from a specific initial state action selection, State 987. The third analyzes a specific time step as a whole. A single meta-model state is examined (meta-model State 626) that encompasses a decision space in the full MDP model.

Germany is faced with a large number of development decisions at the initial time step. The allocation options are constrained by the availability of a single system attributable to a single mission. The full action space at the initial state is captured in Table 7.7.

Germany has the option to begin to invest in the new NGF. It will have the ability to

Table 7.7: Experiment 3b: Initial State Available Actions

| **Action** | **Develop Next Gen Fighter** | **Develop Combat Cloud (legacy)** | **Develop Tornado Refresh** | **Develop Tornado ECR** | **Develop F/A-18 (Germany)** | **Develop F-18G (Germany)** | **$n$ Tornado to Traditional SEAD Mission** |
|---|---|---|---|---|---|---|---|
| Action 38 | | | | | | ✓ | 5 |
| Action 41 | | | | | ✓ | | 5 |
| Action 48 | | | ✓ | | | | 5 |
| Action 53 | | ✓ | | | | | 5 |
| Action 57 | ✓ | | | | | | 5 |
| Action 122 | | | | ✓ | | | 5 |

be allocated to any of the three desired missions. The resulting performance is less certain but anticipated to be higher than alternative options. Germany can invest in a communications upgrade for legacy systems with developing the Cloud Combat focused on integrating legacy platforms. Germany can refresh existing Tornado platforms at a reduced cost and shorter schedule. The draw back is the expected system performance. Germany can also invest in the development of Tornado ECR to help increase the SEAD/DEAD capabilities needed for traditional adversaries. Lastly, Germany can invest in F/A-18 or F-18G platforms. There is a short turn around given historic development and the F/A-18 is nuclear capable. Similar to the Tornado ECR variant, the F-18G variant helps with the traditional adversary via SEAD/DEAD mission supporting capabilities. It is unclear given Germany's mission preferences, the projected system performance, the projected system timelines, and adversary decisions what is the best path forward for Germany.

The resulting Risk-Tolerance Sensitivity Profile (RTSP) for the initial state for Germany is captured in Figure 7.131. All previously described actions are represented. Action 38 and Action 41 peak near $\xi = -1$ and therefore correspond to 'worst decisions'. Action 38 is to invest in developing the F-18G and Action 41 is to invest in the development of the F/A-18. Action 53 and Action 47 peak near $\xi = 0$ and represent 'low risk decisions' relative to alternative actions. Action 48, investment in refreshing existing Tornado platforms, represents the 'high risk' option of those initially available based on the peak near $\xi = 1$. The final action, development of the Tornado ECR variant, results in a dominated solution for all of $\xi$.

Examination of the State 1 Return map (Figure 7.132) helps gain more insight in addition to the State 1 RTSP. The graph represents the change in Return mean and variance as a function of $\xi$ for each available action. For State 1, the Return map clearly shows very little change in relative mean and variance in Return for each state based on the $\xi$ value. Each action mean and variance remains near the same. There is not crossing of mean-variance lines between difference available actions. There is no significant relative shift as the risk-

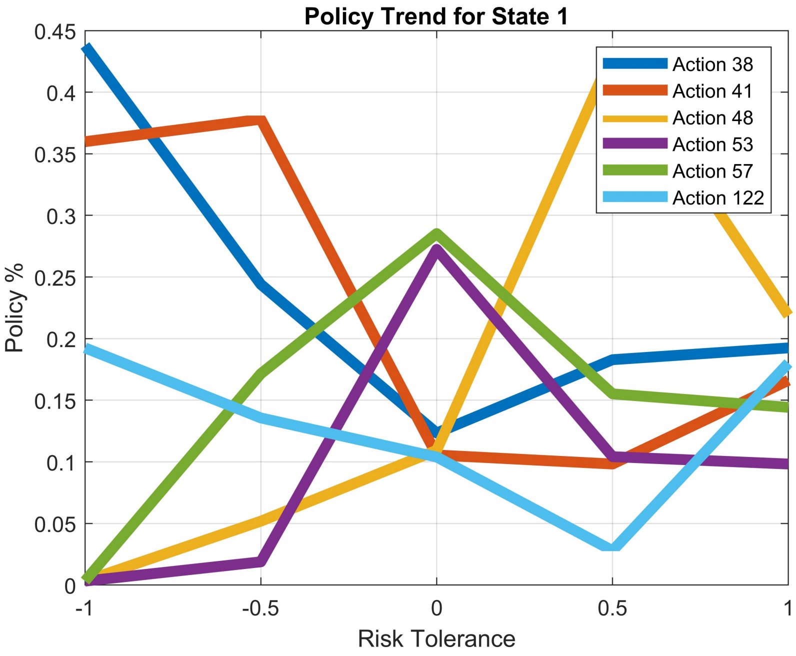


Figure 7.131: Experiment 3b: Initial State RTSP

tolerance level is varied. Action 38 and 41 remain near the low mean and high risk position at the bottom right. Actions 53 and 57 remain in the low risk position in the center left. Action 48 is in the high risk position in the upper right. and Action 122 is fully dominated by all other actions indicated by its position between the maximum risk and worst positions. These observations are consistent and supportive of the RTSP results above.

The last plot of interest for State 1 is the entropy plot (Figure 7.133). Similar to the RTSP, the future Return entropy of each action at State 1 is plotted as a function of risk-tolerance level ($\xi$). Two observations can be made about the entropy sensitivity. First, the near constant entropy values for each state as the risk-tolerance is varied. The second is the grouping of actions in a low and high entropy bin. The correlation between the high Return variance actions and the high entropy actions. These two observations align to the consistency seen in the Return map above.

The second SOI selected for examination is State 987 and represents a state that occurs after an initial decision to develop the F/A-18 is made. The state occurs after the develop-

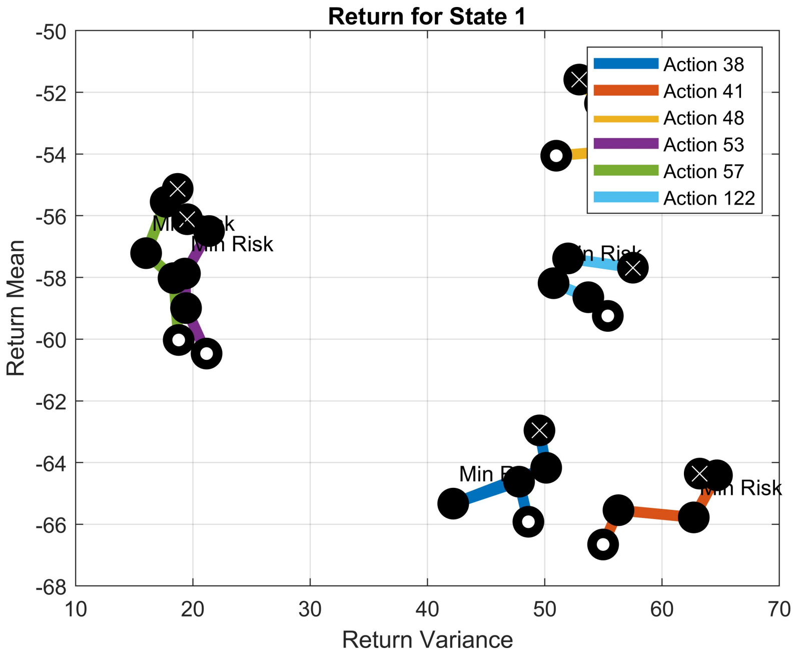


Figure 7.132: Experiment 3b: Initial State Mean-Variance Map

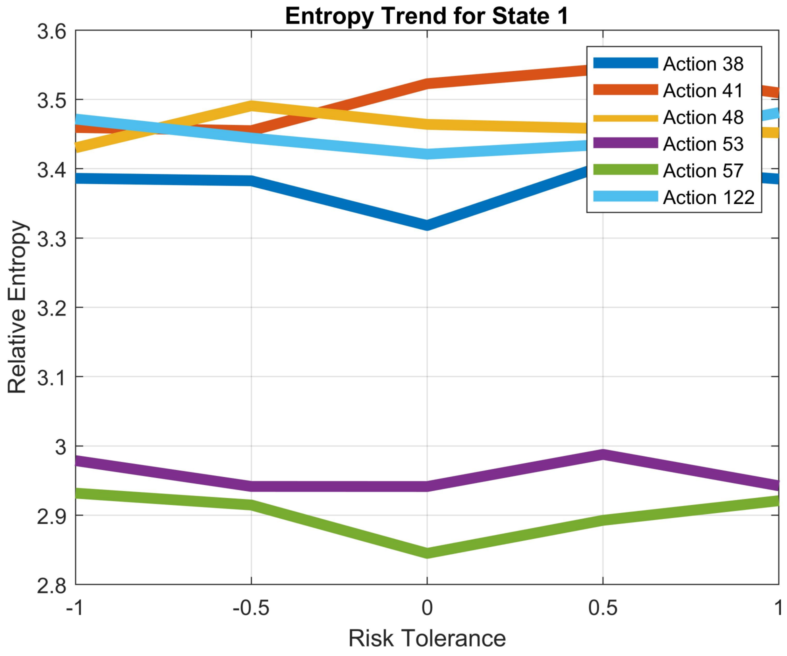


Figure 7.133: Experiment 3b: Initial State Entropy

Table 7.8: Experiment 3b: State 987 Available Actions

| **Action** | **Develop Next Gen Fighter** | **Develop Combat Cloud (legacy)** | **Acquire F/A-18 (Germany)** | **$n$ Tornado to Traditional SEAD Mission** |
|---|---|---|---|---|
| Action 53 | | ✓ | | 5 |
| Action 55 | | ✓ | ✓ | 5 |
| Action 57 | ✓ | | | 5 |
| Action 58 | ✓ | | ✓ | 5 |

ment is complete and, with freed budget, more decision opportunities are present. In this State there are four available actions which are depicted in Table 7.8.

Two actions were also available in the initial state, Action 53 and Action 57. Both actions involve the development of a new system and allocation of available Tornado platforms. Action 53 represents the development of the Cloud Combat with legacy platforms and Action 57 represents the development of the NGF. Developing only a single platform and not acquiring new air wings (e.g. the newly developed F/A-18) would result in potential available resources in the near future. Alternatively, Germany could not only develop the legacy focused Cloud Combat or NGF but could also acquire an F/A-18 platform just developed. This action could restrict future resources and sets a fixed path for a reasonable time forward.

The RTSP for State 987 is captured in Figure 7.134. The result initially appears less complex than the initial state. Action 53 can be categorized in the 'worst' category as

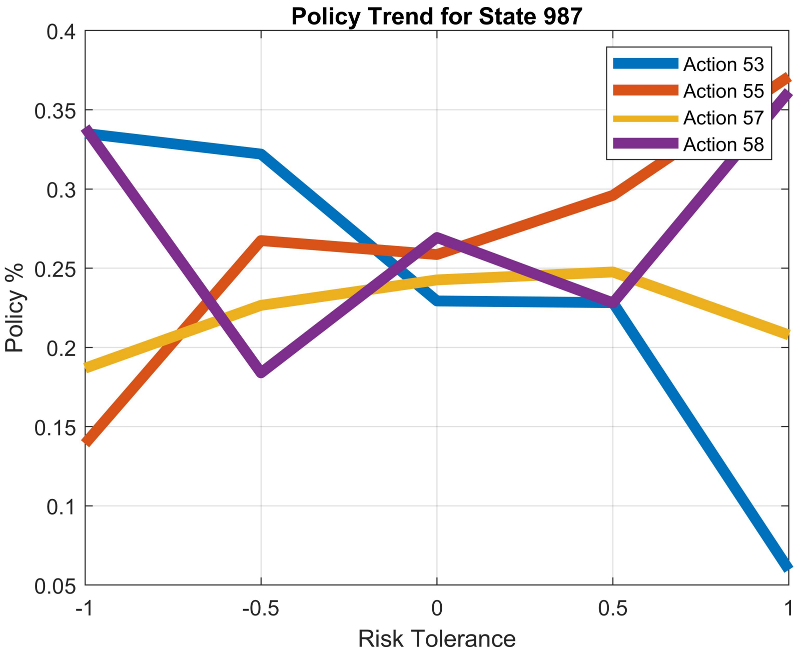


Figure 7.134: Experiment 3b: State 987 RTSP

it peaks at low $\xi$ and decreases to near zero at high $\xi$. Action 53 is the Cloud Combat investment. Action 55 falls into the 'high risk' category for a low to high influence trend as $\xi$ is varied from $-1$ to $1$. Action 55 corresponds to the development of legacy Cloud Combat and the acquisition of an F/A-18. Action 58, development of the NGF and F/A-18 acquisition, shows signs of being dominated by other available actions with a near constant and erratic response as $\xi$ is varied. Action 57, investing in NGF only, can be categorized as 'low risk' due to the shape of the response and peak near $\xi = 0$.

The Return map for State 987, Figure 7.135, captures a more complex interaction as $\xi$ is varied than what was seen in State 1. The black markers with a centered white dot represent the $\xi = -1$ points. The black markers with a centered white 'x' represent the $\xi = 1$ points. Each dot in between represents discrete $\xi$ steps in between.

State 987 has a number of crossing paths where State 1 had none. The relative action Return mean and variance was not sensitive to the changes in $\xi$. There exists more noise in State 987 results due to the number of available future samples than exists in State 1. Given

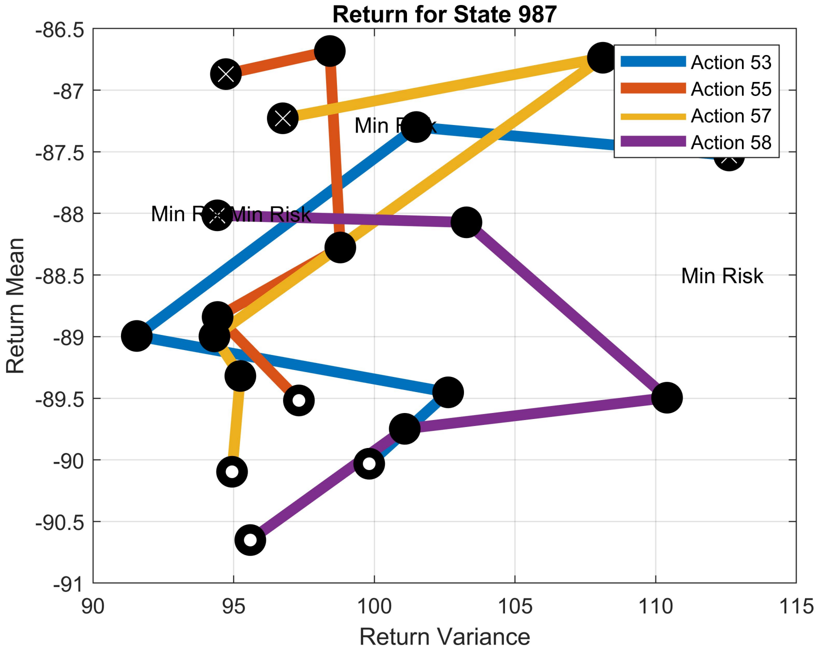


Figure 7.135: Experiment 3b: State 987 Mean-Variance Map

the lower level of sampling, there are still trends and sensitivities that can be evaluated. State 987 shows a much higher sensitivity to the risk-tolerance of the German stakeholder. A clear trend can be seen with Action 53 (develop legacy Cloud Combat). The action stays in a worst case location relative to the other action positions for each $\xi$. Similarly, Action 55 (develop legacy Combat Cloud and acquire F/A-18s) has a high risk relative to the other actions for all of $\xi$. These two observations support the RTSP result of the first action being a worst case scenario and the second being the highest risk. The most erratic options in the Return map is Action 58 (develop NGF and acquire F/A-18s). The inconsistency aligns to the non-dominated attribution derived from the RTSP. The least risk option, Action 57, can be observed to be consistently near the minimum variance point as $\xi$ is varied.

The observations on the Return map support the conclusions drawn from the RTSP for State 987. The last metric is the entropy of each action (Figure 7.136). The observed entropy follows the trends seen in the Return variance of the Return map in general. All actions reduce in overall entropy near the minimum risk point where $\xi$ equals zero except

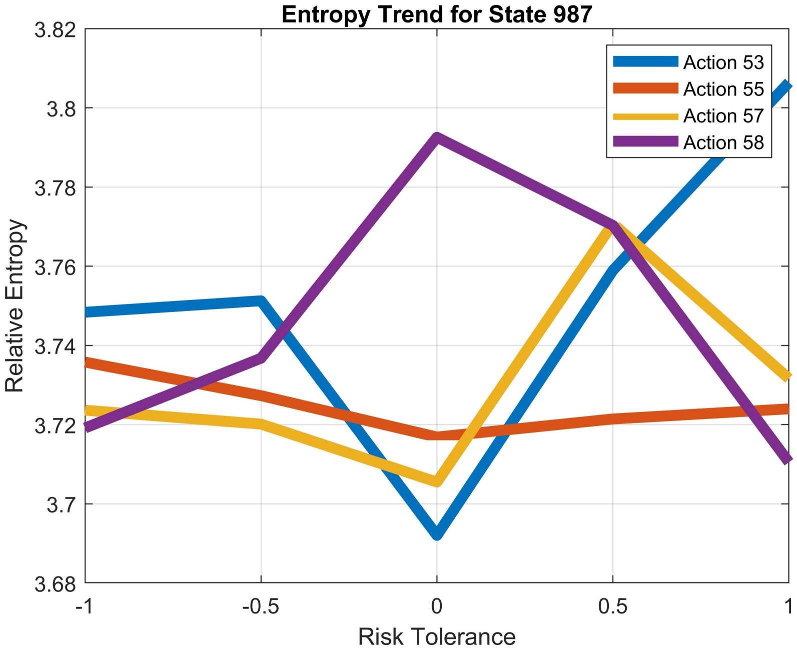


Figure 7.136: Experiment 3b: State 987 Entropy

for the fully dominated action (Action 58).

The last selected state was chosen from the meta-model and not from the reconstituted states. The meta-model selection allows the analysis of a decision space before it is applied back to the full problem. When applied back to the full problem, the relative impact of specific actions can be lost as they are dis-aggregated. The selected state is meta-model State 626. There are five available actions in the state (Table 7.9). Actions 53 and 57 have been previously introduced. Two new actions exist, Action 131 (develop NGF and acquire F-18G) and Action 198 (develop legacy Cloud Combat and acquire F-18G). All full states represented by this composition state (State 626) have actions that are a subset of those presented in the table. All the states represent results when the F-18G was developed first during the initial state. The option here, similar to state 987, is whether to continue to invest in the F-18G (moving to acquisition) or not.

The RTSP is captured in Figure 7.137 shows the relative weight of each action using the risk-based derived policies. A few clear trends are present. First, the actions with and

Table 7.9: Experiment 3b: Composite State 626 Available Actions

| **Action** | **Develop Next Gen Fighter** | **Develop Combat Cloud (legacy)** | **Acquire F-18G (Germany)** | $n$ **Tornado to Traditional SEAD Mission** |
|---|---|---|---|---|
| Action 53 | | ✓ | | 5 |
| Action 57 | ✓ | | | 5 |
| Action 131 | ✓ | | ✓ | 5 |
| Action 198 | | ✓ | ✓ | 5 |

without the acquisition of F-18Gs can be made. Action 57 (develop NGF alone) and Action 53 (develop legacy Cloud Combat alone) both appear to be nearly all ways dominated by the other actions, meaning those that also acquire F-18Gs. This is apparent in the lack of trend relative to the risk-tolerance level for both of the actions. Action 198 appears to be the worst action to select and Action 131 appears to be the most risky though not nearly as relatively risking as previous actions evaluated. The lack of significantly higher risk is derived from the deviation of Action 131 from the others as $\xi$ approaches $1$.

The Return map for meta-model State 626 is shown in Figure 7.138. The map shows a near constant domination of Action 53 as it remains on the right hand side of the plot. Action 57 shows a similar patter except at high risk-tolerance levels where Action 131 remains dominant. Action 131 consistently dominates the other actions with less of a margin at high $\xi$.

The entropy graph for State 626 also presents additional useful information (Figure

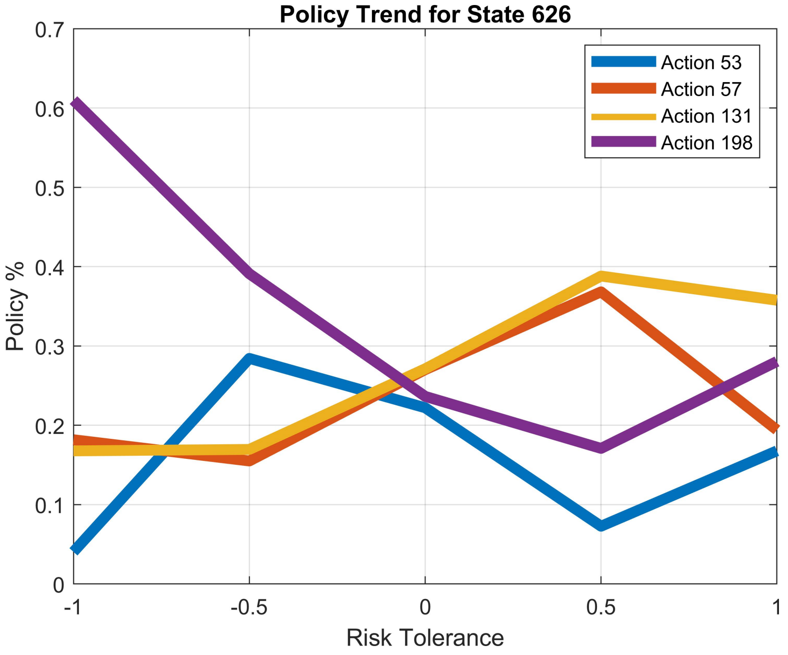


Figure 7.137: Experiment 3b: Composite State 626 RTSP

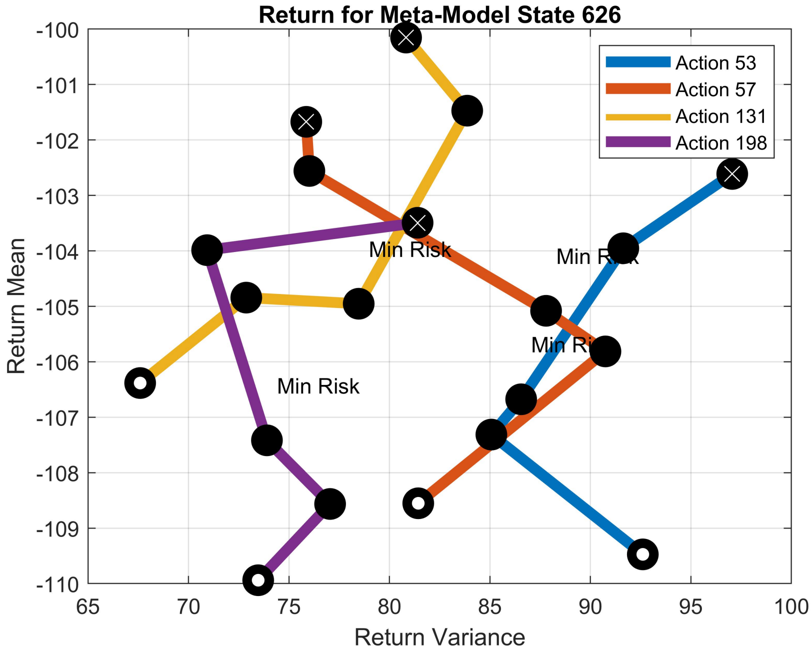


Figure 7.138: Experiment 3b: Composite State 626 Mean-Variance Map

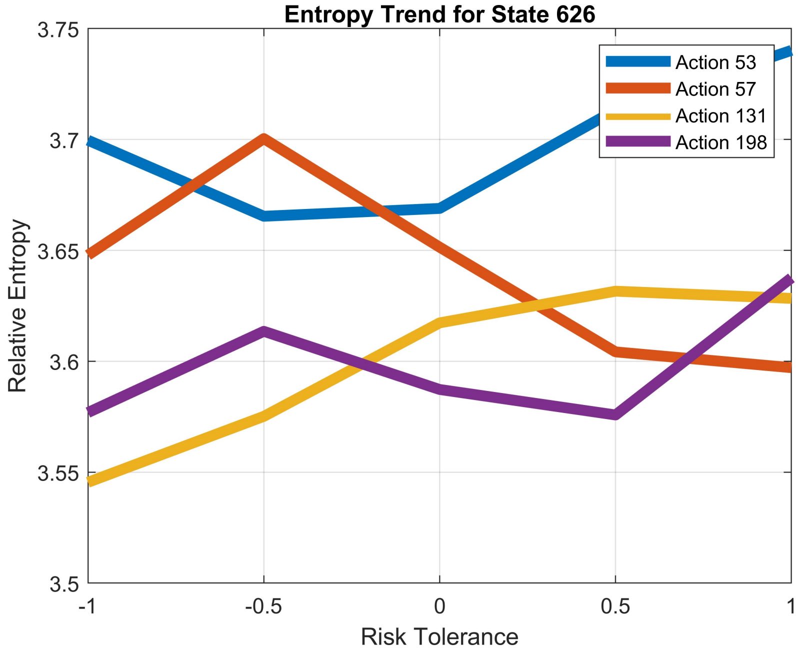


Figure 7.139: Experiment 3b: Composite State 626 Entropy

7.139). Here, the entropy diverges from some expectations given the RTSP and the Return map. Action 53 (develop legacy Cloud Combat alone) proves to be the most volatile for all $\xi$ represented by the consistently higher entropy. This shows the most variation in future outcome and shows that this selection restricts the future less than others. It should be noted that this is done with no necessary gain in mean out come as represented by the action's dominated state derived from the RTSP and the Return map. Action 57 is the other higher volatility state though only at lower $\xi$. This gain aligns to a less restricted future. Both actions allow resources to be decided to be used at future states where deciding to acquire a given system (e.g. F-18G) would restrict future outcomes. Again, it should be noted that the mean Return is negatively impacted by Action 131 and positively impact by action 198. The increase and decrease in mean Return given the lower volatility shows the impact of the F-18G acquisition.

*Step 3: Generate Stakeholder Insights*

The final step of the methodology process is to derive insights to provide to the stakeholder of interest. For experiment 3, the stakeholder of interest is Germany with all other stakeholders, and their decisions, acting as context for analyzing Germany's decision space. Specific rules sets can be developed based on the Step 2 analysis in the previous section. The most important insights come to the initial decision space Germany is faced with and the main purpose of the exercise.

Germany is initially faced with the decision of how to replace their aging Tornado air wings and how to expand current air based capabilities. There is a next generation development path that involves many stakeholders (developing the NGF and the initial iterations of the Cloud Combat). This growth path would entail working with other stakeholders and developing new systems for insertion into the existing SoS. There is a path to refresh the current fleet with new technology and invest in expanding the Tornado platforms role (develop Tornado Refresh and develop Tornado ECR). This refresh path has less technology and system development. The refresh path also is a fully solo development. There is a path of acquiring higher TRL platforms (F/A-18 development and F-18G development). The higher TRL plat form involves acquiring existing developed systems and is reliant on an external partner country, the United States. The descriptions of the varying initial state decisions are present in the representative truth model. The cost sharing, cost uncertainty, performance, and schedule uncertainty are captured for the joint development of the low TRL solutions (NGF and Cloud Combat). The higher reliability of performance with some schedule savings is captured for the refresh path. Lastly, the short timeline of acquiring previously developed systems and the relatively high certainty of their performance are captured. This provides qualitative context to overlay the results produced when the metamodel was evaluated and metrics were produced.

The first item of note is that developing EW platforms to support the SEAD/DEAD mission first aligns to a worst decision scenario. With out the support of more centrally acting assets (NGF, refreshed Tornado, or F/A-18) the resulting SoS will produce a very risky and low performing outcome. The second observation deals with which base platform that Germany should initially invest in. The subset of all options are to develop the Next Gen Fighter (NGF), refresh the existing Tornado platforms, or acquire F/A-18s from the United States. Of these options, the analysis indicates that acquiring the F/A-18 aligns to a worst decision, developing the NGF aligns to a low-risk tolerance, and refreshing the Tornado aircraft aligns to a high-risk tolerance. The recommendation would be to never invest in the acquisition of the F/A-18s and select your current action based on your risk-tolerance profile. The Tornado refresh may be advantageous if adversaries take more time to develop new systems and less performance is needed to combat a near peer threat. The Tornado refresh outcome takes the full brunt of adversary uncertainty. The development of the NGF may take longer but ultimately produces a more dependable outcome in the face of similar uncertainties. The F/A-18 acquisition may close a short term gap but ultimately leaves less resources to develop long term solutions. The opportunity to refresh current Tornado platforms will pass by as they are retired and no resource are available. The NGF will not see IOC until after the time horizon considered due to resource constraints applied to the short term F/A-18 solution. The last opportunity for investment by Germany is to begin the development of the Cloud Combat solution with the integration of legacy platforms. The full impact of follow programs (e.g. extending Cloud Combat to future platforms) is not impact in the time horizon evaluated. Legacy platforms will exist throughout time horizon considered and the SoS performance sees an immediate impact of it's development. The Cloud Combat, though maybe considered a support capability, should be considered as part of a low-risk portfolio going forward.

The second state based information is derived from the analysis fro State 987. This state represents a successor state to the initial state that is present after the development of the F/A-18 is selected as the initial action. The quick acquisition time of the existing platform allows for a very near term decision to emerge within a few years. Note that there is not an opportunity for Tornado refresh and there is no opportunity to develop EW capabilities. The action space encompasses two decisions for Germany:

1. Acquire a fleet of F/A-18s or cancel the program after development

2. Develop the NGF or develop the legacy platform based Cloud Combat

Looking at each of the two decisions independently can be done by looking at the policies without the effect of the other decision variable. The acquisition of the F/A-18 fleet emerges as the higher risk option with the cancellation decision emerging as the worst-to-lower-risk option. This decision dominates the sensitivity to risk. The decision between NGF and Cloud Combat development is eclipsed by the decision to acquire the F/A-18 platform or not. This decision domination is scene in the less response when the policy is controlled for the F/A-18 decision. Due to the decision to invest in the long term programs being identified later than the initial state, the time horizon of the investment results is beyond the planning horizon used. The results of the time horizon cut off is a shift for decisions with out a Return toward the worst decision category and those with a Return toward the higher-risk category. This decision space is close to the point of seeing impacts of the time horizon cut-off.

If the decisions are viewed in combination then additional trends can be derived. Developing the NGF and acquiring the F/A-18 results in outcomes that are sub-optimal compared to others for all risk tolerances. An investment in Cloud Combat alone yields a future with a well integrated legacy SoS but does not address the changing threat environment directly with new capabilities and is the worst option to select. The acquisition of F/A-18 platforms and the development of the Cloud Combat closes the mission need in the time horizon considered, including the nuclear delivery mission) but does come with higher risk. The lowest risk path forward is to cancel the F/A-18 acquisition and develop the NGF. This path allows future resources to remain available and looks to develop the NGF.

To examine this decision space relative to predecessors can provide context as well. To have reached the state of being able to acquire F/A-18 platforms, the development decision would have been made in the initial state. That initial state decision is considered a poor initial decision. If Germany were to select the F/A-18 acquisition today then more information would be know when they actually reach the second decision point of whether to continue and acquire the platform or cancel the program. The epistemic uncertainty would have faded as the future became reality. The actual results state would one of the many single draws of future outcome made in creating the analysis. To examine this viewpoint, the Truth Model could be initialized at the second decision point and a full examination made giving varying initial conditions.

The final analysis conducted was the evaluation of a decision space encompassing the decisions to:

1. Acquire a fleet of F-18Gs or cancel the program after development

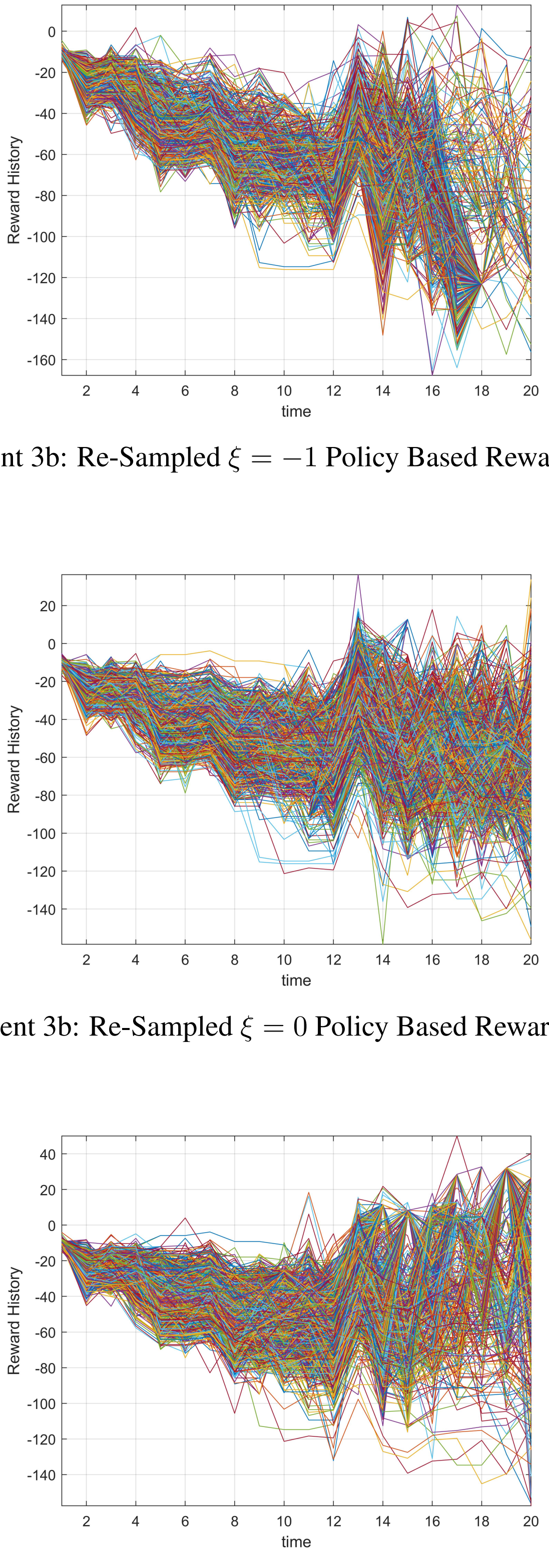
2. Develop the NGF or develop the legacy platform based Cloud Combat

Once again, a non-recommended decisions was made at the initial state to develop F-18Gs. This could be made to help cover the need to counter enemy air defense and support existing platforms across the mission space. Acquiring F-18G platforms and developing the legacy Cloud Combat capability decision results in great long term support but little acting platforms to execute the needed missions and does not directly support the nuclear delivery mission in the future. Just investing in the Cloud Combat capability results in net outcomes that all other decisions surpass, providing only some support and no new acting capabilities. Developing both the F-18G and the NGF has some merit but comes with increased risk relative to only developing the NGF. Developing only the NGF allows for non-committed future resources to go toward other potential long term solutions.

The results reviewed in this section are subject to the assumptions made during the definition of the problem via the construction of the Truth Model. It is possible to return to the problem definition and play what-if games utilizing the methodology presented here.

Above, specific states and decision-spaces were analyzed from the full state space and the compressed state space to produce specific rule sets. These rules sets are derived from the composite of varying risk-based policies. A demonstration of the overall impact of a selected policy can also be measured by rerunning the policy back through the meta-model or full Truth Sim. The results of the replay through either model demonstrate the relative outcomes if different rule sets are followed. For the full example problem the policies were used to re-sample the MDP using the derived policies for three varying risk-tolerance levels ($\xi = 1$, $\xi = 0$, $\xi = -1$). The resulting reward versus time set for each are captured in Figure 7.140a, Figure 7.140b, and Figure 7.140c. Each individual plot appears chaotic and without specific trends. The comparison of the final reward states near the ending time steps provides the relative impact of the risk-based policies. As $\xi$ is increased, the mean of the reward cloud at high time steps begins to shift higher and higher, corresponding to the anticipated higher mean result. The spread of values around the mean at higher time steps if lower for the $\xi = 0$ case and higher for the $\xi = -1$ and $\xi = 1$ case. Little else quantitative information can be directly inferred from the comparison of the reward versus time plots.

The Reward versus time plots display relative significant trends between the result from using each policy. Six additional comparisons sliced by time and risk-tolerance level demonstrate the difference in mean and variance of reward for each of the representative policies. Each comparison of policy return is done with a Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of all Returns seen as a function of time step. One distribution is represented in blue and the other in green. On the PDF plots, the mean at each time step is represented by a vertical line matching the corresponding color. The three sigma limits are marked with a vertical dashed line above and below the mean marker in the corresponding color. This allows the direct comparison of mean and variation as a function of time step for the results of two different risk-tolerance levels ($\xi$). Three cases are examined:



(a) Experiment 3b: Re-Sampled $\xi = -1$ Policy Based Reward versus Time

(b) Experiment 3b: Re-Sampled $\xi = 0$ Policy Based Reward versus Time

(c) Experiment 3b: Re-Sampled $\xi = 1$ Policy Based Reward versus Time

Figure 7.140: Experiment 3b: Re-Sampled Policy Based Reward versus Time

- Case 1: $\xi = -1$ vs. $\xi = 0$ (Figure 7.141 and Figure 7.142)
- Case 2: $\xi = 0$ vs. $\xi = 1$ (Figure 7.143 and Figure 7.144)
- Case 3: $\xi = -1$ vs. $\xi = 1$ (Figure 7.145 and Figure 7.146).

The first case compares the worst risk-tolerance value risk-based policy results to the lowest-risk risk-tolerance value risk-based policy results. The PDF and CDF plots compare the distribution of reward across each time set step. Near the initial state, time step $t = 0$, there is little difference between the two Reward profiles. The reward profiles begin to diverge as the time steps increase and as decisions are made using the different risk-based policies. The mean of the worst case decreases as time steps increase (mean shown by the solid blue vertical line at each time step) while the low-risk case remains higher (Figure 7.141). The variance of each distribution is visually represented by the $3\sigma$ values depicted in the corresponding vertical dashed lines. The relative variance of the worst distribution at later time steps is much greater than the variation of the low-risk distribution. The CDF of the same data provides a slightly different view point (Figure 7.142). A significant translation difference demonstrates the better mean performance of the low-risk policy and the shorter rise time represents the lower variance of the low-risk outcomes. These observations demonstrate that the the worst case having a lower mean outcome and higher variance in outcome than the lowest risk policy.

The second case compares the lowest-risk risk-tolerance value risk-based policy results to the highest-risk risk-tolerance value risk-based policy results. A similar trend to the first case can be observed . The PDF comparing the reward outcomes (Figure 7.143) shows the lower mean of the lowest-risk policy outcomes and the higher mean of the highest-risk policy outcomes. Additionally, the variance trend is visible via the $3\sigma$ markers. As the time steps increase, the variance grows more substantially for the highest-risk policy reward profile than it does for the lowest-risk policy reward profile. The Reward CDF comparison (Figure 7.144) shows a similar relative mean and variance trend seen in case one between the worst policy and lowest-risk policy but in this case between the lowest-risk policy and the highest-risk policy.

The third case compares the reward profiles generated by using the worst case policy to the highest-risk policy. The first two cases compared each extreme to a common baseline seen in the lowest-risk policy. The third case compares the two extreme cases. A clear much more significant divergence in mean Reward is evident as the time steps increase (Figure 7.145). This divergence represents the impact of worst versus highest-risk policies on potential future outcomes. The divergence is clearly depicted by the relative shift between the two distributions shown in the CDF comparison (Figure 7.146).

*Benchmark*

The standard approach today to solving the applied problem is to determine the optimal policy and use it to provide input to decision makers. This results in a less stable and informative solution on which to base decisions. The optimal policy was determined using Q-learning and policy-value iterations as described in Section 6.3.2. The data presented here was selected for comparison to the Experiment 3b results. The policy convergence results are used to explore the stability and the resulting policies are compared corresponding to the risk-based policies to demonstrate the difference in derived information.

State 1 policy convergence is presented in Figure 7.147. The convergence to a steady state can be observed over the course of the iterations. A note can be made on the lack of variation and contrast in the resulting policy. There is little individual contrast between the optimum and the sub-optimum action solutions. Each action is very near an equal policy of 16.67%. The policy is ultimately derived from the value of each state-action sample. Action rewards that have a high variance can create variances in the final policy based on the sampling used. These variances in the final policy can be great enough to provide variation in optimum actions selection from policy evaluation to policy evaluation. Actions that are close in Q-value, represented by the final policy values, can result in a flip or switch

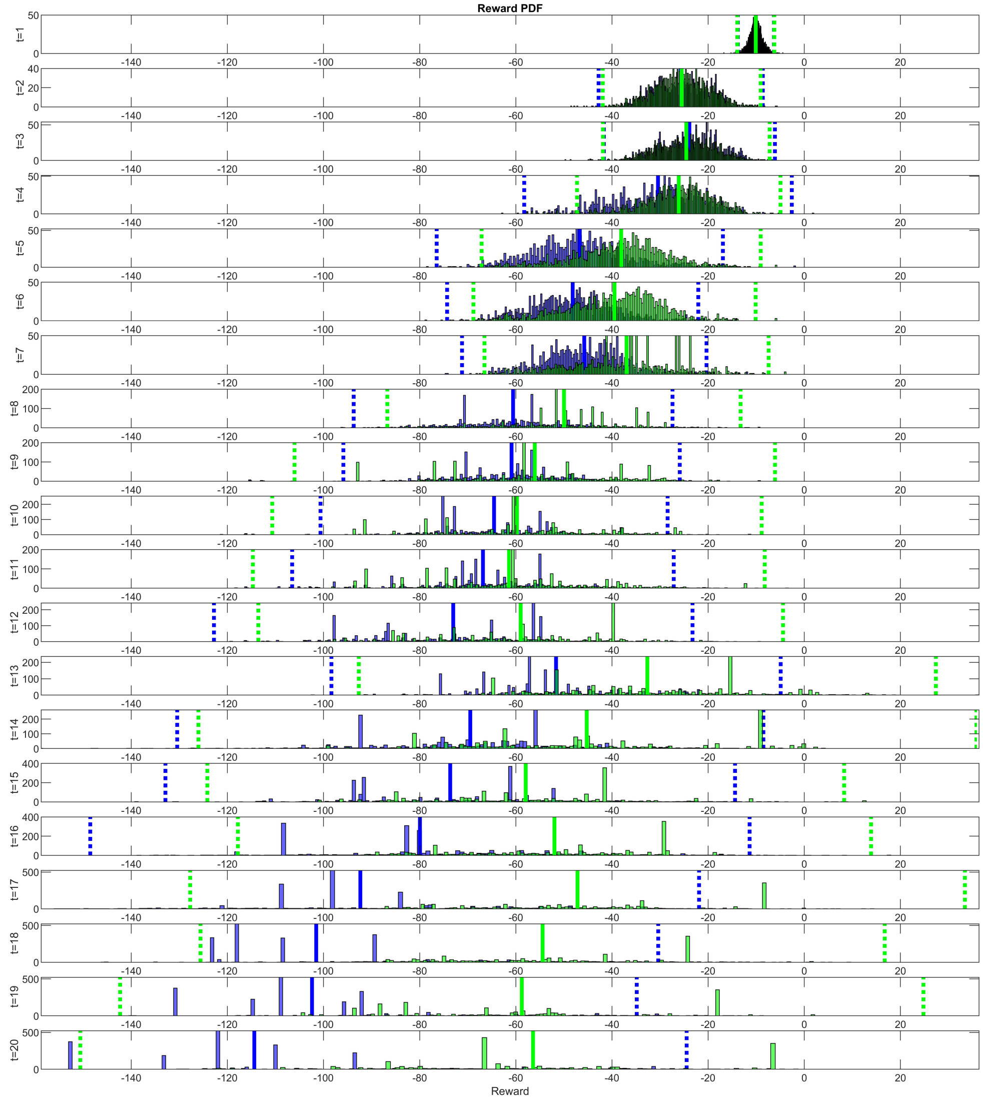


Figure 7.141: Experiment 3b: $\xi = -1$ (blue) vs. $\xi = 0$ (green) Policy Based Reward PDF

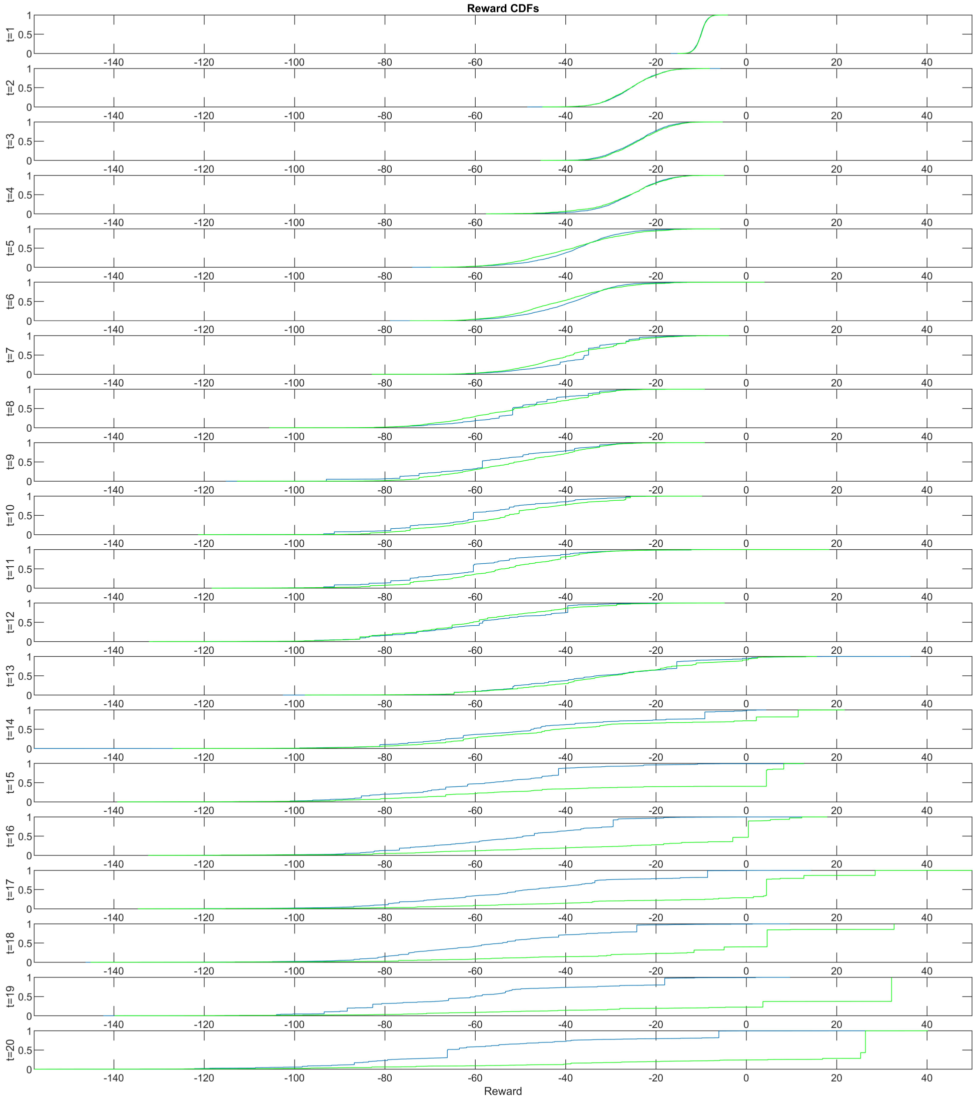


Figure 7.142: Experiment 3b: $\xi = -1$ (blue) vs. $\xi = 0$ (green) Policy Based Reward CDF

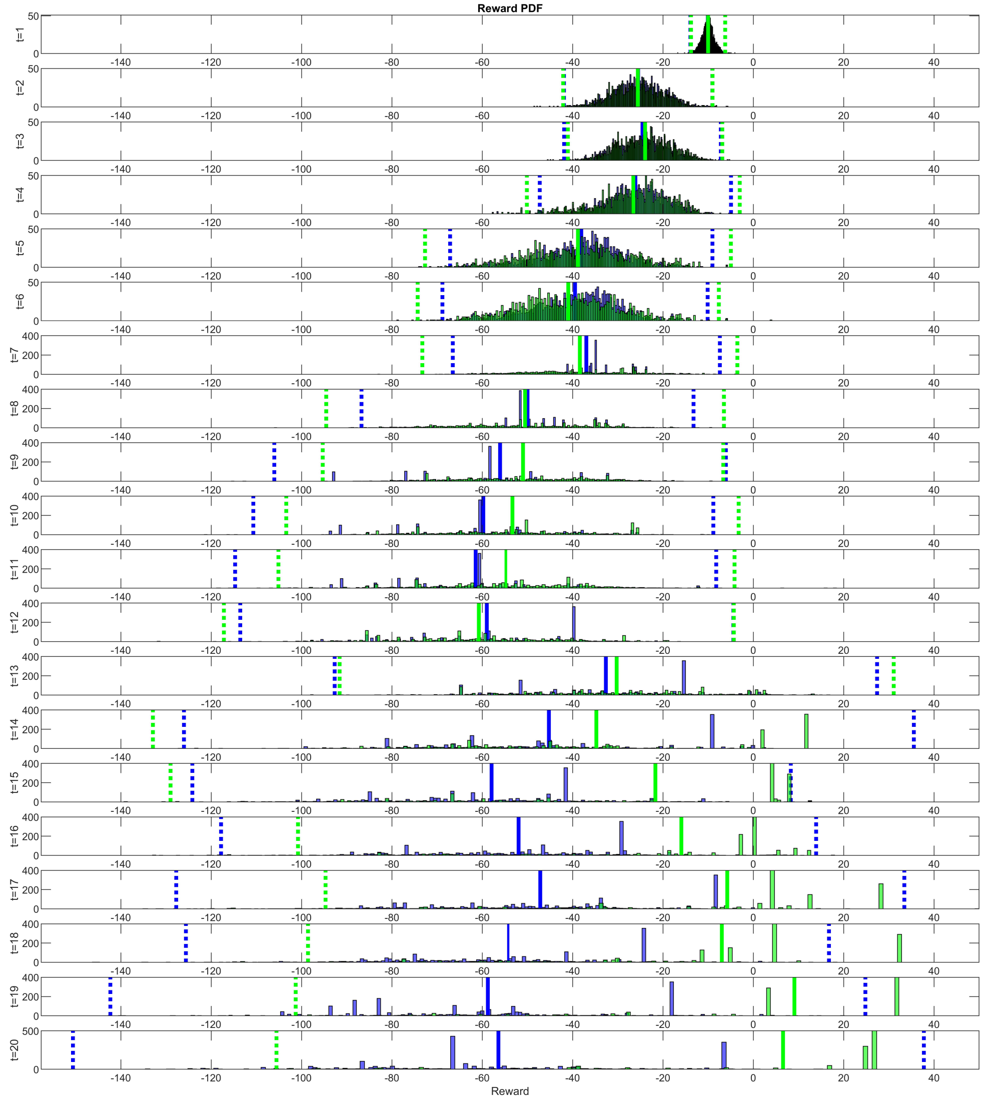


Figure 7.143: Experiment 3b: $\xi = 0$ (blue) vs. $\xi = 1$ (green) Policy Based Reward PDF

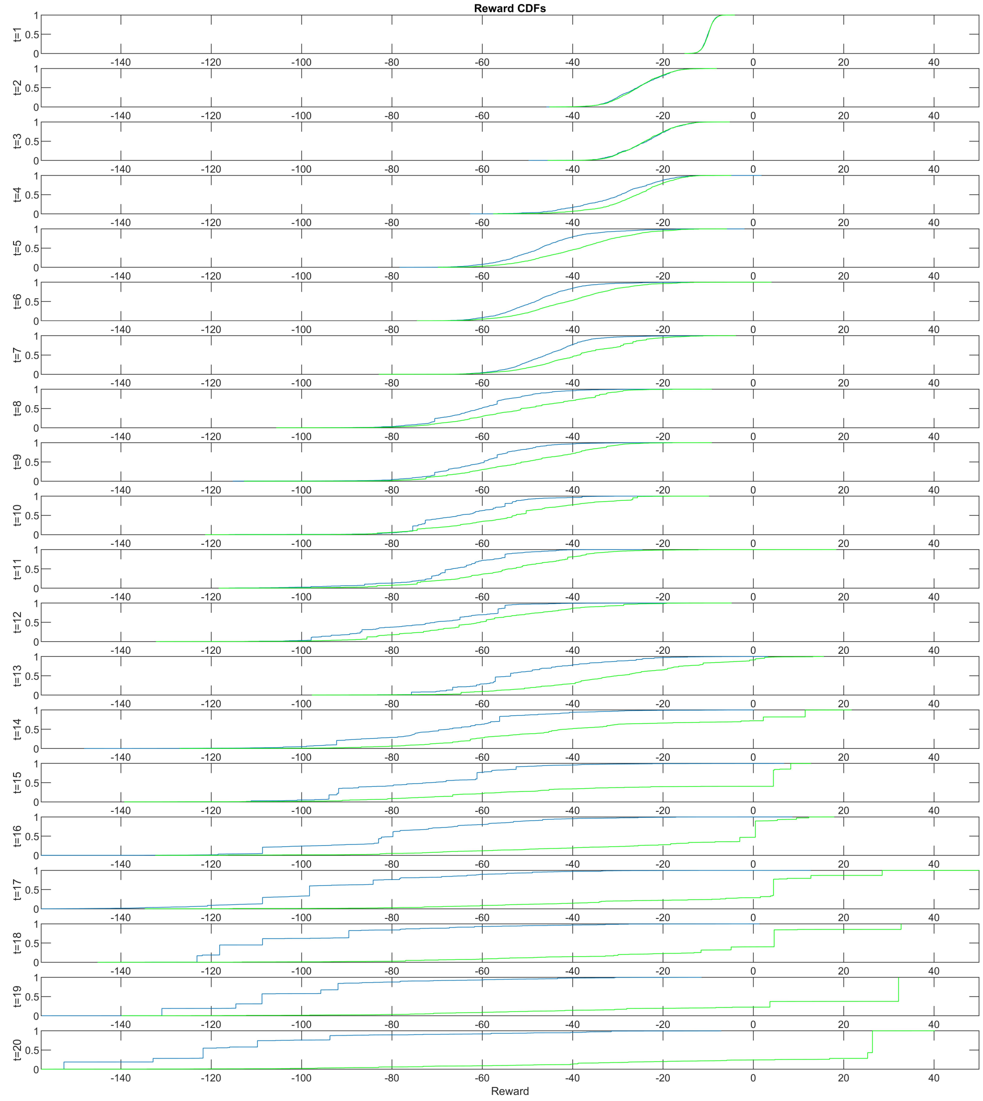


Figure 7.144: Experiment 3b: $\xi = 0$ (blue) vs. $\xi = 1$ (green) Policy Based Reward CDF

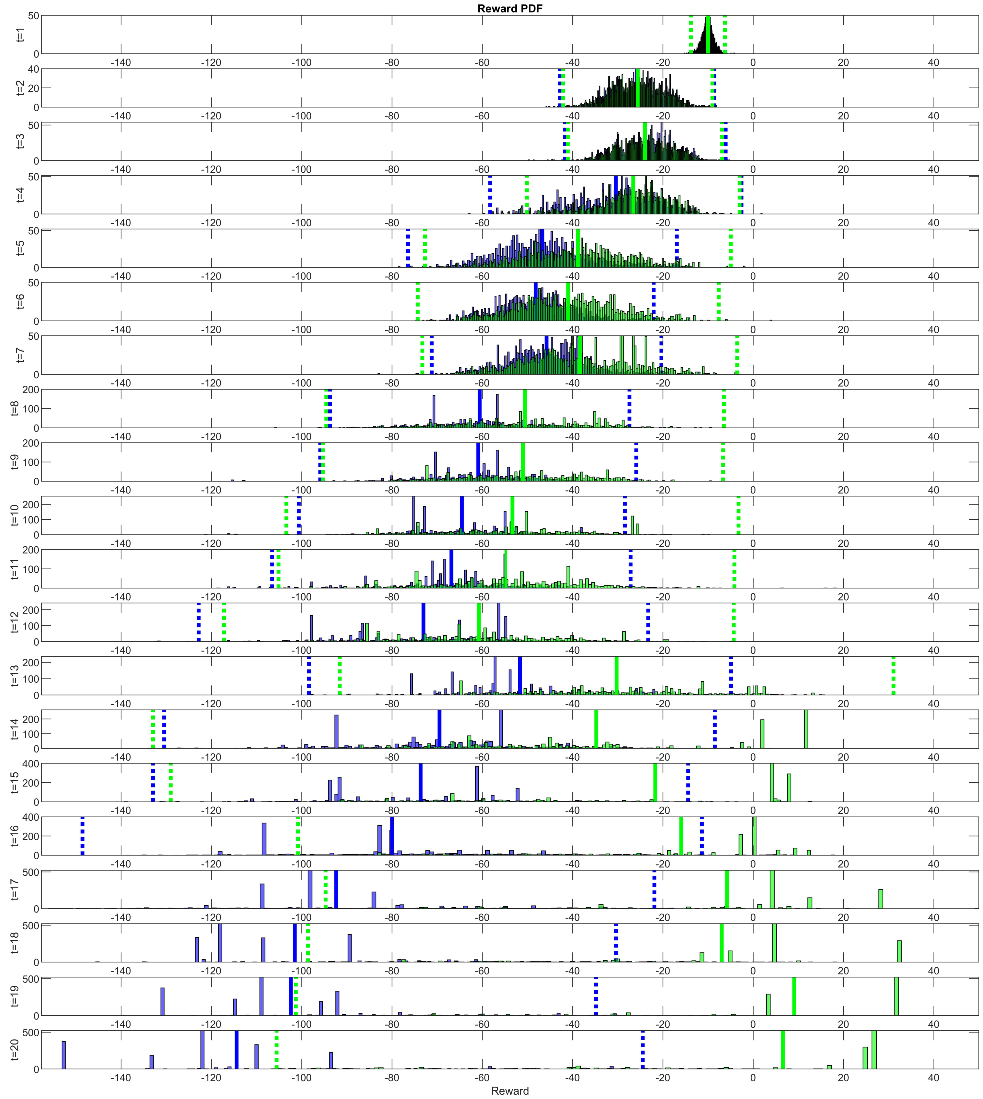


Figure 7.145: Experiment 3b: $\xi = -1$ (blue) vs. $\xi = 1$ (green) Policy Based Reward PDF

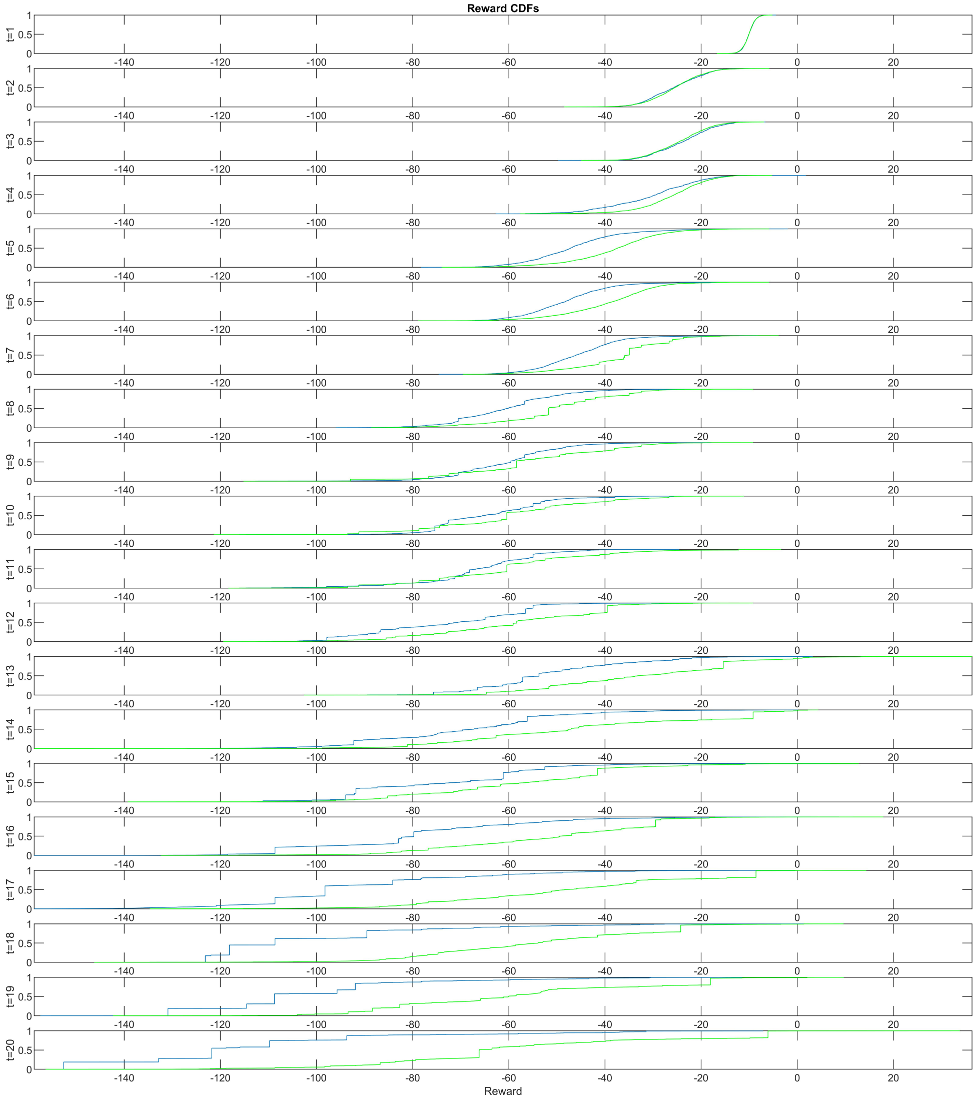


Figure 7.146: Experiment 3b: $\xi = -1$ (blue) vs. $\xi = 1$ (green) Policy Based Reward CDF

of the optimum action.

For the policy evaluation shown in Figure 7.147, the optimal policy can be identified as the highest exploration based policy. The optimum action for State 1 is the highest value at the final iteration, Action 48. The optimum State 1 action corresponds to the most risky option identified previously. The optimal solution yields the recommendation to choose to refresh the Tornado platforms that Germany currently has. There is one additional insight available than just the optimum selection. There is a relative scoring. The relative score roughly matches the mean Return order as expected. The mean Return of each action from taken Figure 7.132 can be compared to the preferred order based on the exploratory policy shown in the optimum policy convergence plot. Both methods rely on similar long term Return (not just Reward). The optimal policy derives it's solution from the mean of the Return. The direct order similarity between the optimal action preferred order and the mean Return of each action demonstrates the reliance.

The risk-based policy method expands the evaluation based on a second dimension. The first dimension is the mean Return and the second dimension is the Return variances. Optimum policy methods do not take into account the Return variance of actions. The risk-based policy method includes the use of a foundation to calculate all derived information previously presented. This inclusion allows for the more nuanced and informative information to be derived from the risk-based policy algorithm, RTSPs, and mean-variance maps.

The second state explored above was State 987. The corresponding policy convergence for the optimal policy solution for State 987 is shown in Figure 7.148. Note that unlike State 1, previous analysis State 987 has a higher degree of chance as the risk-tolerance level is varied. This is depicted in the corresponding risk-based policy mean-variance map presented in Figure 7.135. The optimal policy method identifies Action 55 as the optimal action for State 987. Action 55 desires to both develop the Combat Cloud for legacy platforms and acquire the newly developed F/A-18. The selected action corresponds to one of

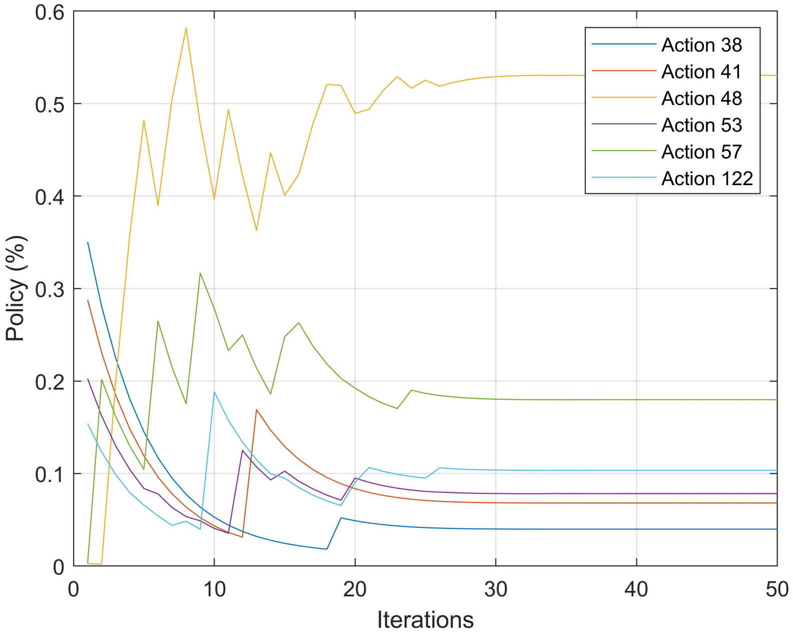


Figure 7.147: Experiment 3b: State 1 Optimal Policy Iterations

the riskier options presented in the risk-based analysis.

The decisions space explored through meta-model State 626 also has a corresponding optimal policy. The convergence graph is shown in Figure 7.149. Note that Action 131 just barely passes Action 57 for optimal action selection. This slight differences corresponds to a sensitivity to variance in outcome. If sampling is changed, there could easily be a switch in optimal action selection. Both actions correspond to higher risk selections outlined in previous analysis. This further demonstrates the lack of information gathered relative to the risk-based approach, the tendency of the to select the highest risk options, and the sensitivity in optimal action selection in the face of high variation in outcome.

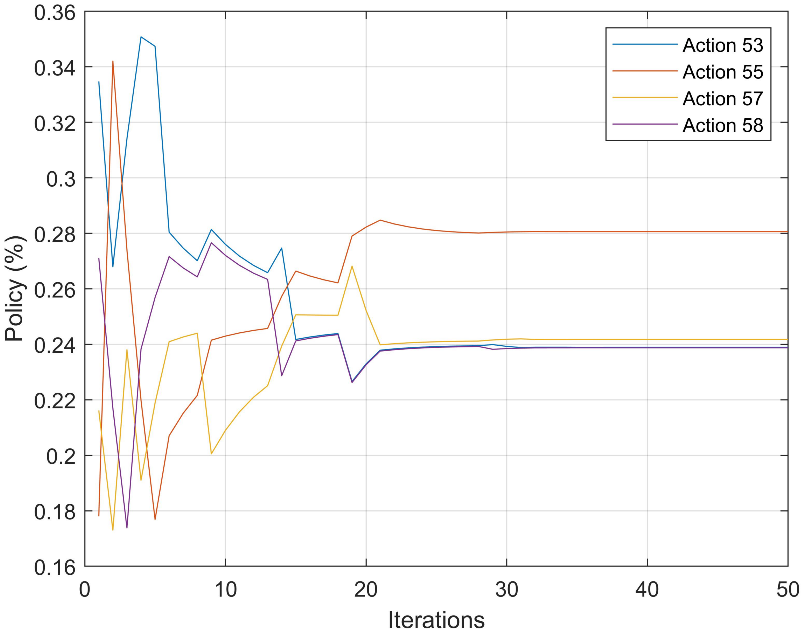


Figure 7.148: Experiment 3b: State 987 Optimal Policy Iterations

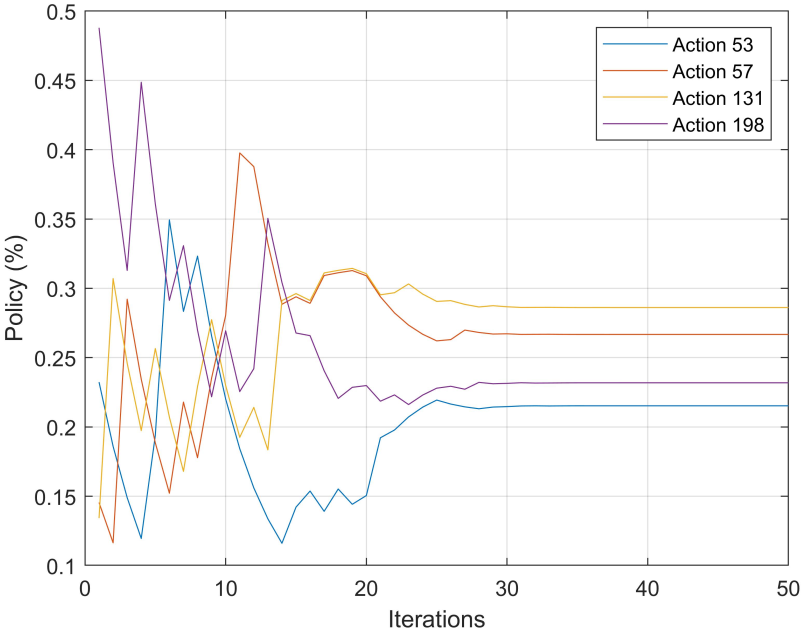


Figure 7.149: Experiment 3b: Meta-Model State 626 Optimal Policy Iterations

# CHAPTER 8
# CONCLUSIONS

Today, no single system is design without addressing impacts of the larger System of System it integrates with. This dissertation set forth to address an identified need to assist stakeholders who are faced with strategic planning in complex and uncertain environments. This dissertation worked to address the associated complexity and uncertainty where routine design methods and decision support methods have lacked focus.

## 8.1 Summary of Methodology Application

The methodology developed as part of this work is designed to be used by an analyst conducting evaluations of future scenarios to help provide immediate decision evaluation and future decision evaluation. Provided here is a summary of the necessary inputs, the expected outputs, and application opportunities.

### 8.1.1 Required Inputs

Through out this work, the concept of an existing Truth Model was used to define both the primary input needs to the methodology and to help bound the simulation test bed used to exercise the methodology. The functions of the test bed was to emulate a full and complete Truth Model. The functions and design of the test bed are described in Appendix refAppendixB. The development of the functions of the described Truth Model (Section 5.2) and the test bed represent the functions that need to be provided in order for an analyst to use the methodology.

The main function of the Truth Model is to provide the development of the decision space. There must be a heuristic or evolving mechanism that can produce a decision space for each stakeholder and play out a selected action. An example can be found in a Agent

Based Modeling where heuristics can be used to develop and select decisions. The heuristics can include technology road maps, system development progression, and resource constraints.

Additionally, evaluating the results of each decision is just as important. The capability to evaluate mission level metrics and overall stakeholder utility is needed to provide the necessary state-action rewards. At the most basic, a transfer function between the current stakeholder allocations and individual stakeholder utility is needed. The transfer function would traditionally be an engagement or mission level analysis produced using frameworks such as FLAMES, AFSIM, STORM, BRAWLER, etc.

The last input of significant note is defining all sources of uncertainty. The methodology relies on the analyst to develop stochastic inputs or models that represent low level uncertainty. An example of this for technology and system definition can be found in the Section 3.2 where uncertainty is used in technology planning. The application of the methodology relies on the bottoms up definition of individual uncertainty elements. For example, a program office for an acquisition can provide estimates and uncertainties on holding a CDR date or a initial delivery date. A system performance analysis team can provide uncertainty related to system performance. The uncertainties can be defined at a low level and more easily justified than they can be directly justified at a macro level. This methodology relies on the this bottoms up approach to address the uncertainty at a macro level.

### 8.1.2 Expected Outputs

This methodology produces insights that can be used to develop a risk-based playbook for a stakeholder. There are two basic types of insights that are provided. State-based risk-tolerance-dependent rules and action-based risk-tolerance-dependent rules. Both sets of outputs are generated as a function of stakeholder risk-tolerance. This risk-tolerance dependency allows actions to be categorized and provided to a stakeholder as actions never to be taken, actions that should be selected for a low-risk-moderate-reward outcome, actions that should be selected for a high-risk-high-reward, and actions that are always Pareto sub-optimal where there is always a better choice independent of stakeholder risk-tolerance.

State based rule sets provide 'if in State $X$, decisions $y$ should be considered given $\xi$ stakeholder risk-tolerance' style rules. Action based rule sets provide 'if a stakeholder sees decision space $Y$ and the state has aspects $x$, decisions $y$ should be considered given $\xi$ stakeholder risk-tolerance' style rules. Each type of resulting output rule set is explored in Section 7.3.

### 8.1.3 Application Opportunities

An analyst is seen as the final end user of this methodology. The analysts goal is, no matter the application, to provide insights to a System of Systems stakeholder facing decision making under high uncertainty. The methodology is designed to look across multiple System of Systems. In this work, a single SoS is often represented as an allocation of assets to a specified mission. The ability to judge development, acquisition, and allocation decision across multiple SoS enables an analyst to evaluate the impact of these decisions in a broad sense. Trade-offs can be made between decisions despite a high degree of uncertainty.

The example used as the full complexity and demonstration problem represents conducting a broad AoA for future fighter investment in the context of the System of Systems the fighter will be integrated with. Similarly, the methodology can be applied at any critical decision period for single or multiple acquisition decisions. The methodology can also be used for long term strategic planning. The previous example primarily looked at the initial decision state. All future states can additionally be evaluated and conditional rules created and applied. This enables the generation of a rule book. A rule book can provide not just immediate guidance but ongoing guidance over time. The evaluation can be re-examined as future possibilities are culled with time.

## 8.2 Hypothesis Resolution

Each hypothesis asserted in Chapter 4 was tested with experiment sets described in Chapter 6 based on the developed methodology depicted in Chapter 5. The goal of each experiment set is to test the validity of the corresponding hypothesis using the developed methodology. The results of each experiment are presented and described in Chapter 7. The results and their analysis confirm that, under specified ground rules and assumptions, the hypothesis hold true. The validity of each hypothesis is explored below.

### 8.2.1 Hypothesis 1: Policy Generation

Hypothesis 1 asserts that the use of the risk-based policy methods on a representative MDP will allow Pareto efficient decisions to be identified. Experiment Set 1 was constructed to test Hypothesis 1. Experiment Set 1 was broken into two separate sets based on the complexity of the set up. Experiment Set 1a used repeated decisions with an abstracted state-space. The simplified set up allowed only the isolation very specific actions and their results to be evaluated. Experiment Set 1a demonstrated that Pareto efficient actions can be identified and separated from inefficient actions using the Risk-Tolerance Sensitivity Profiles (RTSP).

The second experiment set, Experiment Set 1b, was built on running a representative Truth Model setup. Cases were run across a spectrum from fully intuitive to those requiring more retrospection to evaluate. Each case added complexity to the decision space:

- Experiment 1b Case 1: Acquisition Only
- Experiment 1b Case 2: Acquiring Develop Systems and Developing New Systems
- Experiment 1b Case 3: Acquiring, Developing, and Allocation of Systems

The results demonstrated the continued Pareto efficient action identification based on long-term stakeholder utility. The variation between short term and long term utility was specifically explored along with the sensitivity to variations in system capability. The ability for the risk-based policy algorithm to produce relevant RTSPs was demonstrated across each case of increasing complexity. Each RTSP allowed more than just the Pareto efficient and inefficient actions to be identified. The profiles allowed the Pareto efficient actions to be categorized into those that produced the 'worst', the 'low risk', and the 'high risk' results for a given stakeholder.

Hypothesis 1 was tested using Step 2 of the methodology and it was demonstrated that under ideal (Experiment Set 1a) and realistic (Experiment Set 1b) conditions to be true.

### 8.2.2 Hypothesis 2: State Compression

Hypothesis 2 asserts that the state space can be compressed to produce a lower order MDP, or meta-model, and that the meta-model risk-based policy solution will not dilute the resulting RTSPs. Additionally, the computation time would decrease while the resulting RTSPs would not drop in relevant fidelity. Experiment Set 2 was designed to demonstrate the validity of Hypothesis 2 using Step 1 and Step 2 of the methodology. The full Truth Model experiment cases developed to test the risk-based policy generation algorithm (Experiment Set 1b) were used to evaluate the state space compression feasibility.

Experiment Set 2 results and analysis demonstrated that the RTSP can be considered valid down to a state compression ratio of near 25% for well sampled states. For the same conditions, the policy computation time was reduced by 70%-80% under higher complexity cases and even more for less complex cases. Experiment Set 2 showed that Hypothesis 2 held using the methodology Step 1 for specified compression ratios.

### 8.2.3 Hypothesis 3: Derived Information

The final hypothesis of this work asserts that the information gathered from using the outputs of the risk-based policy algorithm, the RTSPs, allow more information to be gathered above and beyond what is produced using optimal policy methods. The optimal policy methods represent the current state of the art approach to solving architecture evolution problems. Proving Hypothesis 3 results in proving the methodology developed has produced utility above and beyond current approaches.

Experiment Set 3 was broken into two sub-sets: Experiment Set 3a and Experiment Set 3b. Experiment Set 3a explores the information gathered from evaluating the increasingly complex Truth-Model-based scenarios of Experiment Set 1b and Experiment Set 2. The experiment allowed the intuitive and digestible problems to be evaluated for information and compared against optimal policy solutions.

Experiment Set 3b represents the full complexity test case. This test case was not evaluated in Experiment 1b due to the computational requirements to solve it. The full complexity test case was then not available for state compression analysis in Experiment Set 2. The full complexity problem required the full methodology to be applied in order to fully test the hypothesis using a stressing case.

Additionally, the full complexity test problem was based on a representative operational case (FCAS SoS development). The operational case was selected to demonstrate the utility of the methodology and provide context for it's application to real world problems. Experiment Set 3 assists in demonstrating the final utility of the methodology beyond just testing Hypothesis 3. The methodology is compared to an optimal policy method demonstrating the additional utility of using the risk-based evaluation approach. The application also gives direct context for the application of the methodology above and beyond the test scenarios used in Experiment Set 1 and Experiment Set 2.

## 8.3 Reflection on Research Objective

The goal of the research conducted for this dissertation was *to develop a new methodology that will instantiate the evolution of a System of Systems specifically with regard to the decision making of the stakeholders accounting for the influence of the external environment, the morphing of the requirements, and the availability of resources over the lifetime of a*

*SoS to enable robust individual stakeholder decision making*.

The methodology described in Chapter 5 represents the culmination of this body of research. The methodology represents the resolution of the Research Objective. The methodology itself is *new*. It is based on a novel risk-based policy algorithm and applied state space reduction methods.

The methodology uses a Truth Model to fully explore a multi-stakeholder decision space and represent it as an MDP. The decision space can be fully characterized or represented using a reduced MDP, or meta-model. The sampling method, time based model representation, and Return calculation methods enable the impact of time based metrics to be captured. Capturing time based metrics allows the future impact of decisions to be accounted for along with aspects changing in the temporal dimension. Such aspects include changes in external environment, changes in mission preference, and changes in resources over time.

The last aspect of the Research Objective addresses the idea of helping stakeholders make decisions under high uncertainty. The use of RTSPs allows information to be gathered on the relationship between the mean and variance of long-term stakeholder utility. The additional information provided to stakeholder by using the RTSPs allows stakeholders to see through the fog of uncertainty. It allows the quantification of relative risk and reward for individual decisions while accounting for a large number of degrees of uncertainty. The key algorithm used to produce RTSPs is the risk-based policy algorithm and is at the heart of the methodology. The methodology ultimately enables to production of the needed products and the final development of stakeholder recommendations based on the produced products.

The methodology developed through the research presented in this body of work has satisfied the original Research Objective. A number of contributions have been made to the field of Aerospace Engineering and System of Systems Engineering. Additionally, a number of future paths for continuation of this research have been identified.

## 8.4 Revisiting the Motivation

The developed methodology was shown to meet the original Research Objective set forth in the beginning of this work. The Research Objective was developed in response to the described motivation. This work was motivated by the need to supply a decision maker within the United States military a method to plan future investments and developments in a high uncertain environment. An environment where other cooperative and non-cooperative stakeholders will impact the desired outcome. An environment that requires balancing many competing objectives.

The full complexity example problem demonstrates the ability to use the designed methodology to address the needs of a single stakeholder. The selected stakeholder of interest, Germany, was able to be provided a risk-based rule set under the uncertainty associated with developmental timelines, predicted system performance, resource constraints, multi-objective priorities, cooperative stakeholder decision making, and non-cooperative stakeholder decision making. The methodology will help a single United States defense stakeholder plan without relying on another level of unified and centralized control.

## 8.5 Summary of Contributions

This work has resulted in a number of contributions that culminate in impacts on the field of System of Systems Engineering with specific impacts on strategic military force structure planning. The highlighted impacts below are those that enabled the construction of the methodology that addresses the over arching Research Objective and provides advances toward fully addressing the motivating problem.

**Risk-Based Policy Algorithm** The risk-based policy algorithm was created to directly address the need to evaluate the future uncertainty and it's impact on the value of making as specific decision. The concept of portfolio risk and reward from the field of Investment Science was selected and merged with Reinforcement Learning (RL) practices to produce the risk-based policy algorithm.

The future variance in outcome for a stakeholder represents the future risk and the mean outcome represents the reward. A Pareto frontier of action portfolios is established using the risk-reward of a stakeholder for a specified decision point. During the evaluation of a decision point a risk-tolerance is selected which yields a single action portfolio selection.

The RL practice of policy evaluation and iterations are used to develop a policy based on the risk-tolerance and action portfolio selection. The algorithm iterates over all states and updates the selected action portfolio (or policy) at each iteration. The result is a action portfolio characterization across all future states for the given stakeholder.

This algorithm is at the heart of the uncertainty analysis present in this work. It fully enables the development of the RTSP concept. This novel algorithm ultimately allows a cloud of uncertainty to be evaluated in concrete, quantified, and absolution terms.

**Risk-Tolerance Sensitivity Analysis** The impact of the risk-tolerance sensitivity analysis builds on the contributions of the risk-based policy algorithm. The analysis method turns the novel risk-based policy algorithm into a decision evaluation tool. No current method is equipped to analyze the resulting policy space generated by the algorithm. The RTSP based analysis enables a broad characterization of individual decision points of stakeholder.

The RTSP analysis enables a stakeholder to fully characterize the impact of uncertainty via the risk-tolerance variable. It enables the direct identification of Pareto efficient decisions (both positive and negative) relative to long-term reward. Stakeholder can then reject non-efficient and non-productive decisions. The remaining selectable actions result in a low-risk to high-risk decision frontier. The analysis method provides stakeholders with the capability to make risk-based decisions under extreme complexity and extreme uncertainty.

**Decision Space Analysis** The impact of the decision space analysis builds on the contributions of the risk-tolerance sensitivity analysis. The analysis of RTSPs addresses a single stakeholder decision point. The decision space analysis identifies similar decision points and assists in the development of rules for decision spaces. Decision spaces are characterized by the available actions a stakeholder can take at varying states. In other words, a decision space has constant actions but variable states.

Rules based on the available actions can be directly developed but other results are available as well. RTSPs can be grouped by similarity with variation between groups correlate to state information. This yields conditional action profiles based on the state seen by the stakeholder.

The development of blanket decision space recommendations and conditional decision space recommendations based on the risk-tolerance of a stakeholder provides the basis of the development of a stakeholder risk-based playbook.

**Evaluation of Complex and High-Uncertainty Decision Spaces** The most significant contribution of this work is the final products delivered to stakeholders which is dependent on the previously highlighted contributions as well as the full developed methodology. The final products are insightful observations for strategic decision makers of directed System of Systems (e.g. military SoS) accounting for a highly complex future (multi-stakeholder and multi-objective) and a highly uncertain future. The hierarchical composition of analysis (risk-based policy development → RTSP analysis → decision space evaluation) is what enables the final contribution and the final utility of this work.

## 8.6 Future Work

The methodology presented in the dissertation is a solid foundation for SoS stakeholder decision making under a large amount of uncertainty considering multiple stakeholders and multiple objectives. Given the foundation set by this work, there are a number of areas for future work to explore, building upon the work presented in this dissertation.

**Increased Scalability** There are three areas to explore to directly increase the scalability of the methodology presented in this work. The first is to reduce the complexity of the problem representation. Second is to reduce the state space used for evaluation. Third is to reduce the action space used for evaluation.

The methodology presented in this work used a direct sample of a Truth Model which was assumed to only yield Monte Carlo samples (such as a time based simulation) and relied on storing the full sample sets nearing the creation of a full MDP. Potential options exist to reduce the sampling and representation burden. First is to develop and use a Truth Model that allows arbitrary state sampling and off-policy action evaluation. The added features open the door to Reinforcement Learning techniques that were not directly considered in this work due to applicability. Off-policy action evaluation would allow for more information to be gathered from the truth model for any given sampling. The assumption of a time-based simulation as the Truth Model and MC sampling leads to unbalanced sampling (over sampling near the initial state and sparse sampling near leaf states). Arbitrary state sampling would allow for even sampling across time steps.

Online and direct development of the meta-model should be considered in the future. A large amount of information is both stored and required to be processed to develop and evaluate the meta-model. Partial online records of sampling statistics were used. A direct online, episode by episode creation of the relevant metrics should be explored further. In doing so, the maintenance of the mean and variance metrics should be taken into account. In addition to online evaluation, direct function development should be explored. Direct function approximation can be used to continue to reduce the memory and processing necessary for the current method. This can include Constitutional Neural Networks for state-space reduction, Deep Neural Networks for value functions, Recurrent Neural Network (RNN) for time-based evaluations.

Action space decomposition and reduction can also be explored further. Action spaces are held constant in the methodology from the full sampled action space to the meta-model action space. Action space decoupling (similar to the typical state space factorization explored in Chapter 4) can be applied to reduce the overall action space. This method would essentially create multiple action spaces present at each state. An example of two potentially decoupled action spaces for the problem addressed in this work would be allocation of systems and the creation of systems.

**Expanding State and Action Metrics** This work concentrated on the exploration of stakeholder risk and reward based metrics (e.g. RTSP). Two concepts were presented to evaluate the decision outcome volatility and the decision opportunity cost of stakeholder. Full evaluation and development of the additional decision evaluation metrics can be further explored. There exist other qualitative decision metrics that can be quantified and integrated into the existing methodology.

**Meta-Methodology Analysis** The final step to process or function development is to develop interfaces and enable use as a black box. The idea of encapsulation enables more simple and easier development of complex systems. Methodologies can be encapsulated just as processes and functions can be. This methodology can be selected and used in a larger body of work.

The methodology can be used as a black box to evaluate long term priorities, varied future scenarios, and full sets of future system development timelines. The necessary inputs of this methodology can be varied to evaluate more than just a single Truth Model setup and a single stakeholder. Meta-methodology analysis may require additional scalability considerations depending on the complexity of the problem under consideration.

**Automatic Generation of Stakeholder Risk-Based Playbook** The output of the methodology is the information which enables the development of rule sets to build a risk-based stakeholder playbook. Future work can concentrate on automating the development of the risk-based playbook. The automatic development should consider meta-methodology analysis and options to increase the scalability as outlined above.

# Appendices

# APPENDIX A

# DISSERTATION SUMMARY

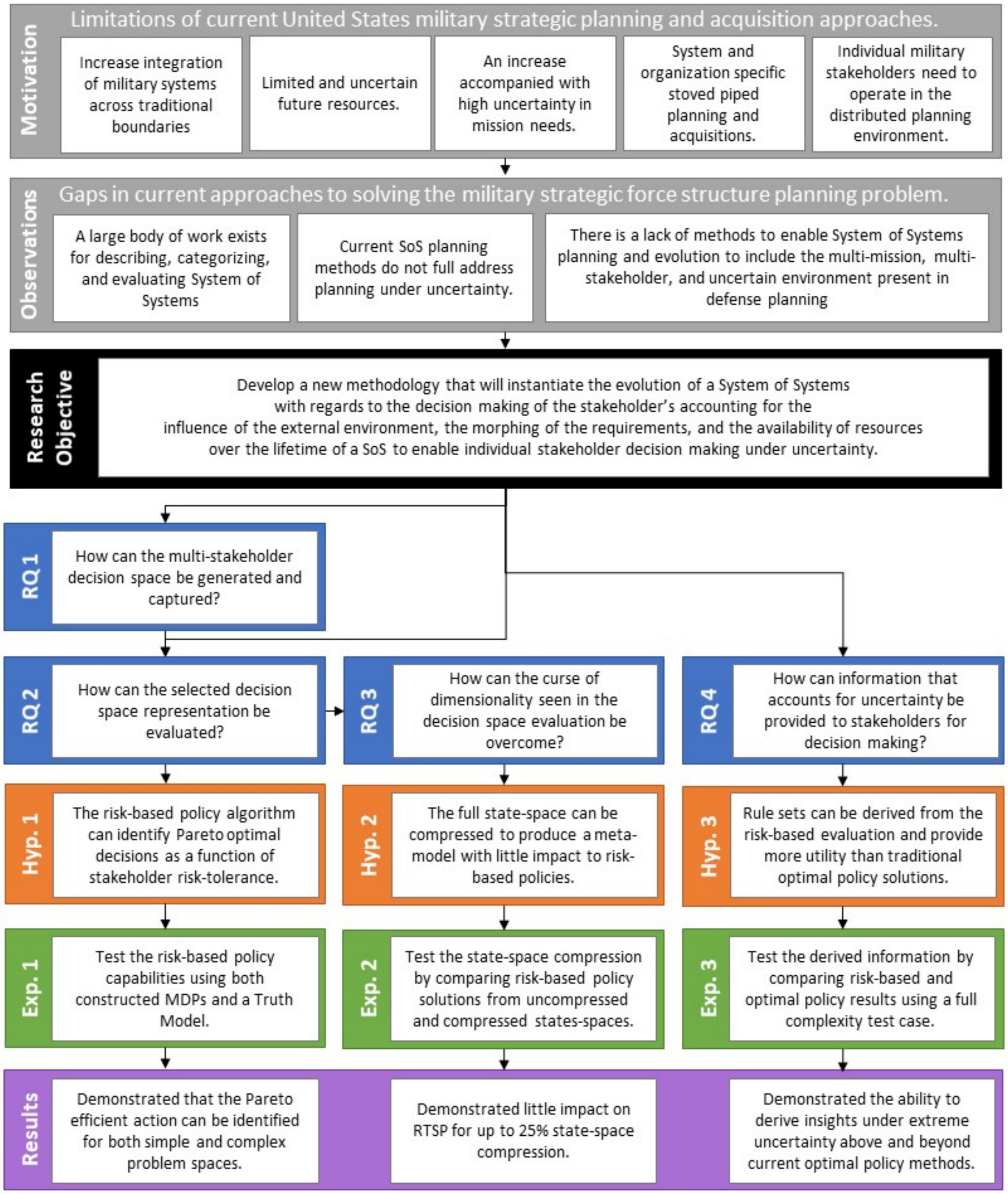

Motivation
Limitations of current United States military strategic planning and acquisition approaches.
Increase integration of military systems across traditional boundaries
Limited and uncertain future resources.
An increase accompanied with high uncertainty in mission needs.
System and organization specific stoved piped planning and acquisitions.
Individual military stakeholders need to operate in the distributed planning environment.
Observations
Gaps in current approaches to solving the military strategic force structure planning problem.
A large body of work exists for describing, categorizing, and evaluating System of Systems
Current SoS planning methods do not full address planning under uncertainty.
There is a lack of methods to enable System of Systems planning and evolution to include the multi-mission, multi-stakeholder, and uncertain environment present in defense planning
Research Objective
Develop a new methodology that will instantiate the evolution of a System of Systems with regards to the decision making of the stakeholder's accounting for the influence of the external environment, the morphing of the requirements, and the availability of resources over the lifetime of a SoS to enable individual stakeholder decision making under uncertainty.
RQ 1
How can the multi-stakeholder decision space be generated and captured?
RQ 2
How can the selected decision space representation be evaluated?
RQ 3
How can the curse of dimensionality seen in the decision space evaluation be overcome?
RQ 4
How can information that accounts for uncertainty be provided to stakeholders for decision making?
Hyp. 1
The risk-based policy algorithm can identify Pareto optimal decisions as a function of stakeholder risk-tolerance.
Hyp. 2
The full state-space can be compressed to produce a meta-model with little impact to risk-based policies.
Hyp. 3
Rule sets can be derived from the risk-based evaluation and provide more utility than traditional optimal policy solutions.
Exp. 1
Test the risk-based policy capabilities using both constructed MDPs and a Truth Model.
Exp. 2
Test the state-space compression by comparing risk-based policy solutions from uncompressed and compressed states-spaces.
Exp. 3
Test the derived information by comparing risk-based and optimal policy results using a full complexity test case.
Results
Demonstrated that the Pareto efficient action can be identified for both simple and complex problem spaces.
Demonstrated little impact on RTSP for up to 25% state-space compression.
Demonstrated the ability to derive insights under extreme uncertainty above and beyond current optimal policy methods.

Figure A.1: Dissertation Summary

# APPENDIX B
# TRUTH MODEL DESCRIPTION

The Truth Model Description appendix describes the Truth Model set up used in the experimental set ups described throughout Chapter 6. The purpose of the appendix is to provide an overview of the Truth Model test bed used for experimentation. The specific set up for each experiment is described in Chapter 6.

## B.1 System to Metric Mapping

Often an individual system-to-metric mapping is done through multi-level Modeling, Simulation, and Analysis (MS&A). This work does not address the exploration or development of methods to evaluate current SoS against specific mission level metrics. Chapter 3 outlines such methods. For the purposes of experimentation, the Truth Model uses a mapping between the systems and the final metrics of interest for individual stakeholders.

Each stakeholder has a unique and personal prioritization of missions. Additionally, each stakeholder has control of the development, acquisition, operation, and retirement of specific systems. Systems owned by stakeholders comprise a SoS. The combined capability of the systems will result in a mission outcome measured by a mission level metric.

Each mission is outlined through the traditional kill chain (Find, Fix, Target, Track, Engage, and Assess or F2T2A). In Figure B.1, an example of a basic Integrated Air Defense System (IADS) is used to show the mapping of SoS systems to a mission. For a given snapshot in time, or a frozen SoS state (which may never exist), the type and number of systems is specified (Equation B.1).

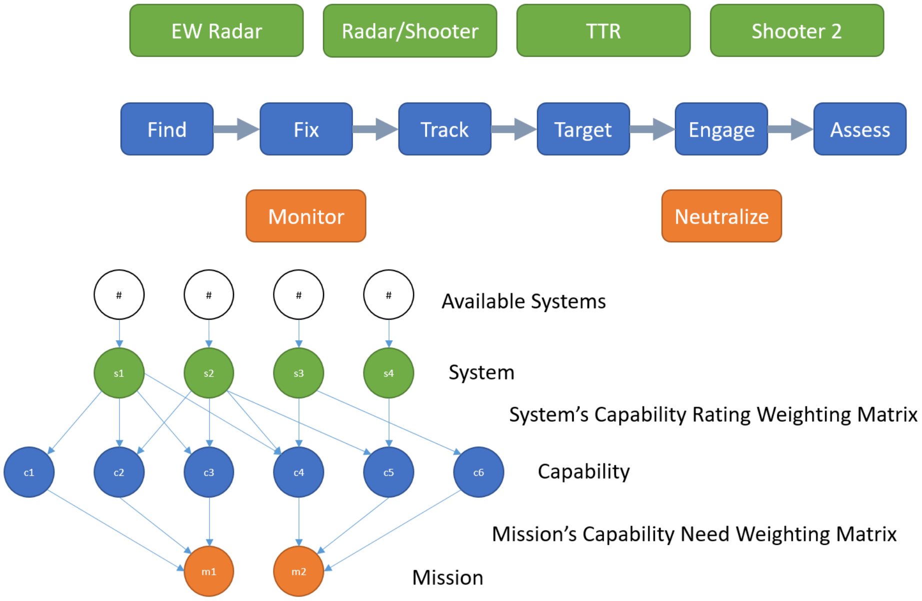


Figure B.1: Example Assessment Mapping

$$\mathbf{n}_{sys} = \begin{Bmatrix} n_1 \\ n_2 \\ \vdots \\ n_m \end{Bmatrix} \tag{B.1}$$

where $n_m$ is the number of system $m$ current available

Each system contributes to the ability to execute specific capabilities utilized to execute a mission. This is represented by matrix $C_{m,k}$ in Equation B.2.

$$C_{m,k} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,k} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,k} \end{bmatrix} \tag{B.2}$$

where $c_{m,k}$ is the system $m$ contribution to capability $k$

In this example the capabilities are defined by the kill chain. Ultimately, each of these capabilities contribute to an overall mission success (B.3). In this example, some stakeholders may have a preference for the 'monitoring' mission while others in the 'neutralize' mission. Stakeholders may control systems that contribute to all missions but have a preference for one mission over another.

$$V_{k,l} = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,l} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k,1} & v_{k,2} & \cdots & v_{k,l} \end{bmatrix} \tag{B.3}$$

$v_{k,l}$ = system $k$ contribution to mission $l$

For any given point in time the system level metrics can be calculated using Equation B.4. There is an additional constraint on the utility based on the number of systems. The utility can be scaled linearly or by non-linear means. Figure B.2 shows the relative impact of each additional system from $0$ systems to the max number of systems (normalized at $1$).

$$\mathbf{q} = \mathbf{n}_{sys}^T C_{m,k} V_{k,l} \tag{B.4}$$

where $\mathbf{q}$ is the vector of mission level metrics

The final metric for each mission is used to measure the ultimate over all utility (or reward) for each stakeholder (or player) over the course of the game. A graphic depicting the mapping relationships is depicted in Figure B.3.

### B.2 System Life Cycle and Decision Points

The representation of the system life cycle determines the acquisition, refresh, and development decision-space of stakeholders have. A generic system life cycle is modeled using a finite state machine. A basic diagram of the generic state machine can be found in Figure B.4. Each system type moves through Asset States (depicted in blue). An asset type

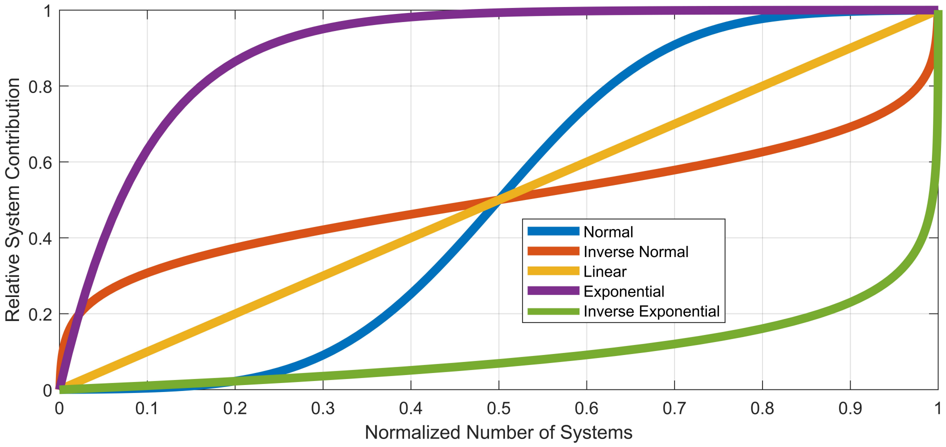


Figure B.2: System Performance Impact Curves

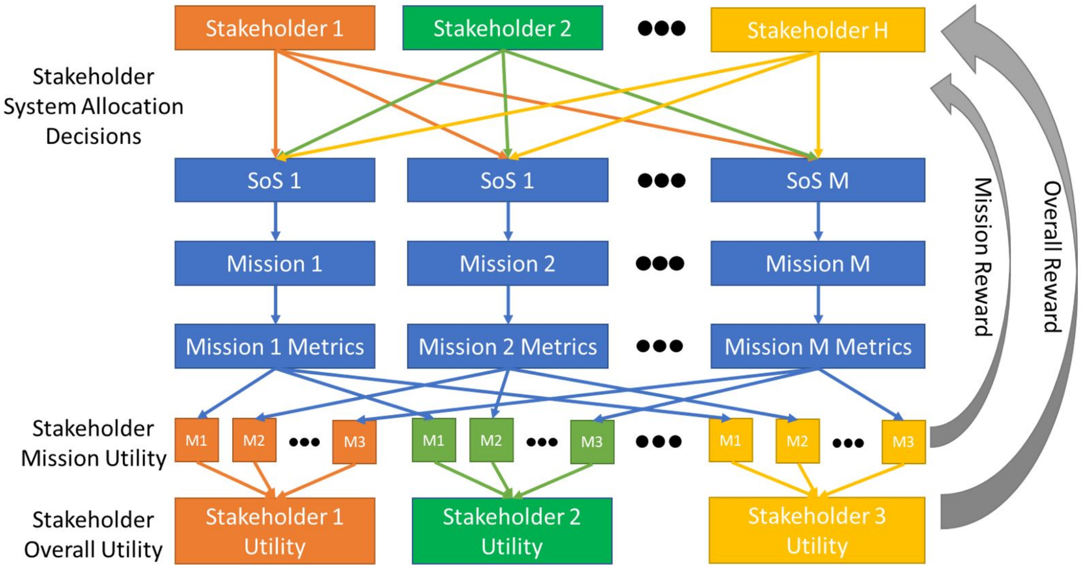


Figure B.3: Metric Mapping Overview

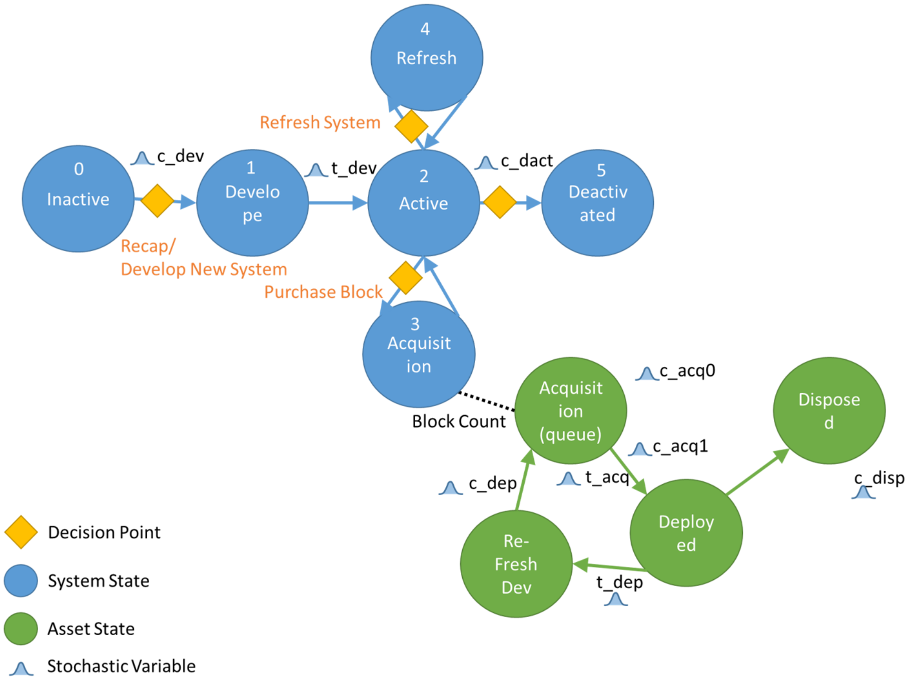


Figure B.4: System Life-Cycle State Diagram

represents a class of asset, such as a Global Hawk. Each individual system, or Block 30 Aircraft 2, moves through the System States. The stakeholder(s) responsible for the Asset Type have decision points that transition individual systems through the state machine.

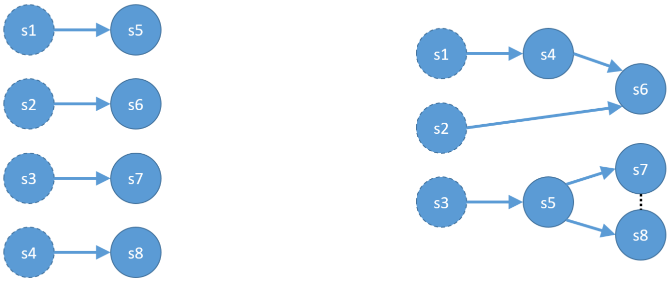
The decision options open to stakeholders are:

**Technology Refresh:** Invest the NRE to insert new technology in the existing baseline. New technology and the performance impacts will be added to the existing system and the deployment time increased.

**System Development:** Development of a new system with fully associated NRE cost.

**Block Acquisition:** Manufacturing of a new block based on a previously developed system.

**Dispose of System:** Retire system from use.



$$\begin{bmatrix}0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$$

Figure B.5: System Development Tree and Compatibility Example

## B.3 Capturing Decision Uncertainty

Uncertainty comes in many forms as does quantifying the impacts of uncertainty. Uncertainty that is related to decisions is captured throughout the model of the system life cycle in several ways. In Figure B.4 the points of uncertainty are identified with a probability distribution icon. Uncertainty that is related to the cost and time for development is captured using pre-defined probability distributions. For technology development, similar cost and time distributions are attributed for each stage of TRL advancement to capture uncertainty (Figure B.6). Additional uncertainty is accounted for in the final performance and capability provided by individual systems at the block level. The two main types of uncertainty captured are temporal and performance uncertainty. Resource (e.g. budget) and requirement (e.g mission preference) are also represented.

## B.4 Technology Development

Additional stakeholder decisions revolve around the development of key technologies to be utilized during the development of a system type or a technology refresh of a current system type. The action to develop, acquire, refresh, or dispose of systems represents a tactical decision. The action to invest in technology is a strategic decision. Both decisions

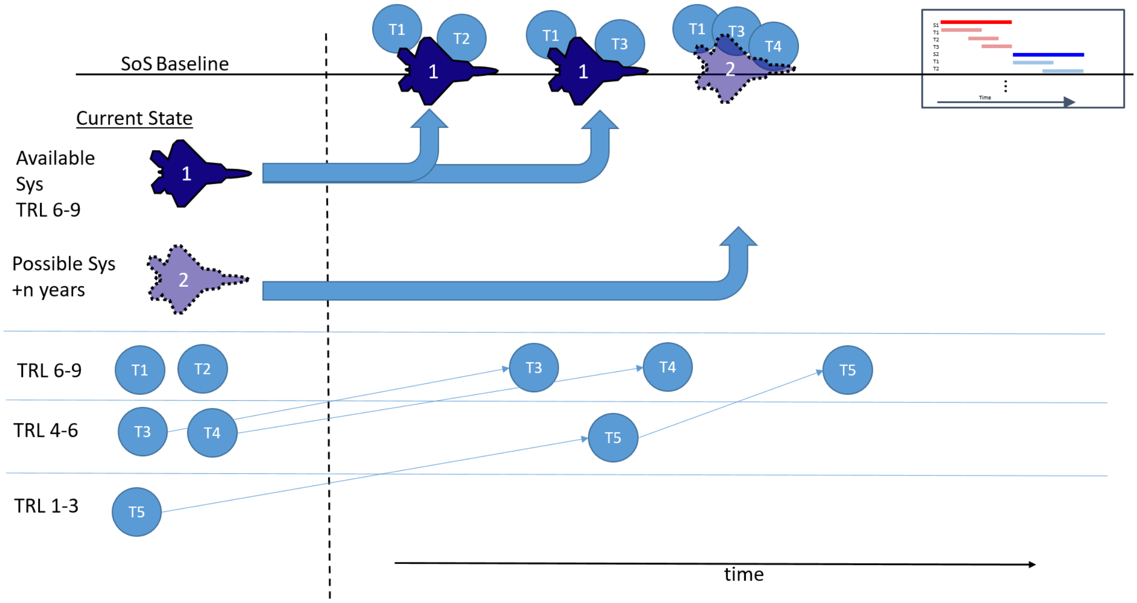


Figure B.6: System Development and Technology Insertion

have a delayed response with respect to observable mission metric changes. Between the two, the impact of technology investment decisions is not realized for a longer period of time and comes with much more outcome uncertainty than individual Asset Type decisions.

Technologies are used as enabling factors to calculate the capabilities of an system type. Figure B.6 depicts the flow of technology into an Asset Type's development or technology refresh. Decisions are made by stakeholders to invest in and increase the TRL of technologies at a cost. Once reaching a TRL threshold, technologies are available to the stakeholder for development or refresh of an Asset Type. The technologies can then be associated with a given asset as shown is Figure B.7.

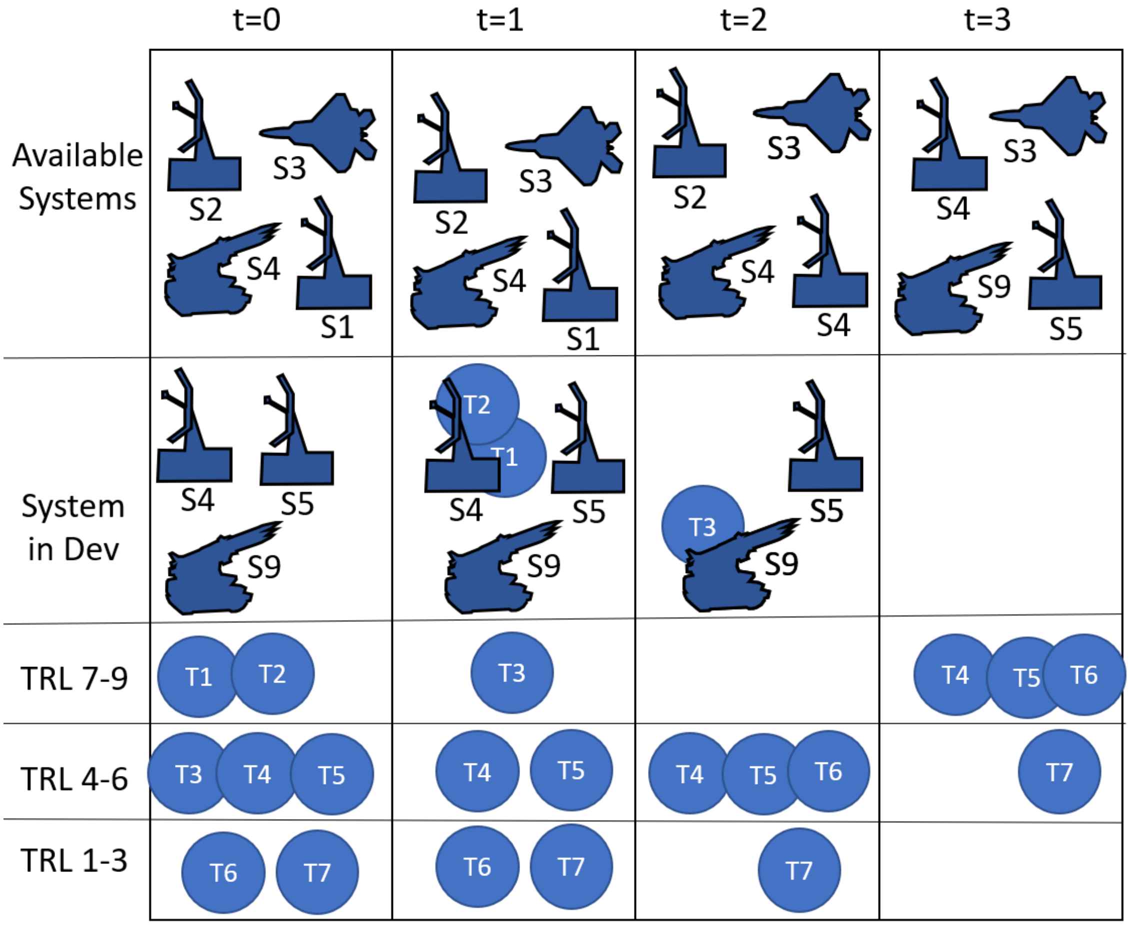


Figure B.7: Technology and System Evolution Example

# APPENDIX C

# ANNOTATED SELECT RISK-TOLERANCE SENSITIVITY PROFILES

Appendix C examines selected Risk-Tolerance Sensitivity Profiles in the context of the supporting metrics. This allows a more thorough explanation of examples by connecting inputs, intermediate metrics, and outputs of the RTSP generation process. The purpose is to provide consolidated a tutorial on RTSP meaning and interpretation of RTSPs beyond what is discussed in Chapter 5, Chapter 6, and Chapter 7. Selected information from each of the chapters is combined and presented to provide a more consolidated look at individual cases.

## C.1 Repeated Action Examples: Equivalent Reward and Return

The first set of examples were selected from Experiment Set 1b Case 1 and represent the simplest of cases. The input MDP is generated using a sequence of repeated equal decisions at each time step. The repeated decisions result in repeated Reward outcomes. The repetition causes the short term Reward and long term Return to become equivalent. This allows the defined Reward mean and variance to be directly tied to the resulting RTSP.

The first example is depicted in Figure C.1. The left of the figure shows the relative action Reward and the left shows the resulting RTSP for the initial state in the repeated decision sequence. The Pareto frontier for the mean-variance of the action Reward space is depicted on the left and highlights the higher risk action in red and lower risk action in blue. With two actions there is never a specific low risk option. Moving along the Reward Pareto frontier from the blue action to the red action is depicted on the right RTSP by moving from the $\xi = -1$ to $\xi = 1$ along the $x$-axis. Near $\xi = -1$, Action 1 will make up most of the weightings along the Reward Pareto frontier. This carries over, due to the repeated decisions, to the RTSP as a preference at lower risk-tolerance levels. Symmetrically, Action

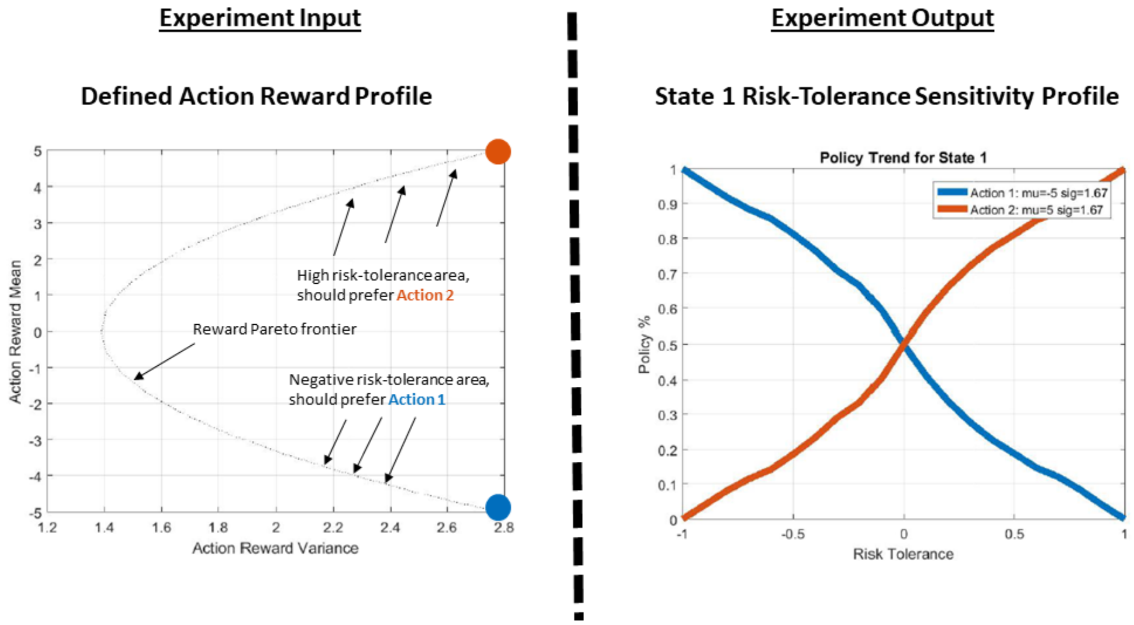


Figure C.1: Annotated Two-Action Decision Space and RTSP

2 has a higher weighting on the Reward Pareto frontier as Action 1 is approached. This weighting of Action 2 shows on the RTSP as a preference as $\xi$ nears 1.

The second examples adds a single action with a mean in between the previous actions (Figure C.2). The results from the first example are still scene with Action 1 (yellow) and Action 3 (blue). The addition of the red intermediate action will begin to be weighted heaviest near the minimum risk point ($\xi$ 1). This manifests on the RTSP with a peak near a zero risk-tolerance. This demonstrates the importance of understanding the trends and not just a maximum at any given risk-tolerance value.

The third example adds a fourth action which represents a mild Pareto inefficient action (Figure C.3). The variance of the new action is set high relative to the previous three (Action 4, purple). Similar to Action 2 and the RTSP peak near $\xi = 0$, Action 4 results in a reduced preference near the same location. The reduced preference is due to a lower weighting along the entirety of the Reward Pareto frontier. The fourth example (Figure C.4) takes Action 4 to another extreme with a significantly increased variance making the action significantly inefficient. The result is a near zero preference for all risk-tolerances in

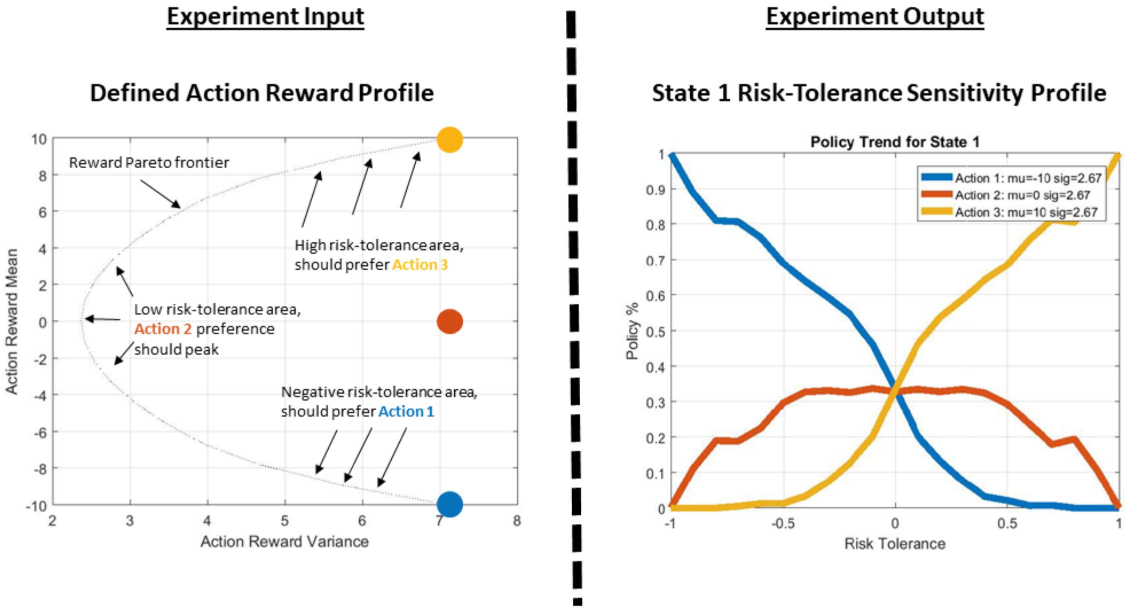


Figure C.2: Annotated Three-Action Decision Space and RTSP

the RTSP. The resulting policy impact can be viewed as noise within each calculated policy across risk-tolerance levels.

The fifth and sixth examples step the complexity up one more level. Figure C.5 shows the inputs and outputs for a seven action set-up built to represent a Pareto frontier. Moving from Action 1 to Action 7 is moving from a $\xi = -1$ to a $\xi = 1$. As The risk-tolerance is varied from low to high, the resulting peaks for each action correspond to moving along the Reward Pareto frontier. Figure C.6 introduced inefficient actions to the original Pareto frontier based actions shown in Figure C.5. The additional actions can be seen to be noise below the Pareto efficient actions in the RTSP similar to what was scene in Figure C.4.

## C.2 Multi-Step Return

An example with a similar set-up to Experiment 1b Case 2 was selected to demonstrate multi-stage decision with temporal and performance uncertainty. Figure C.7 depicts the setup on the left and the resulting initial state RTSP on the right. At any given state when the stakeholder can make an action, the stakeholder can choose between acquiring one of

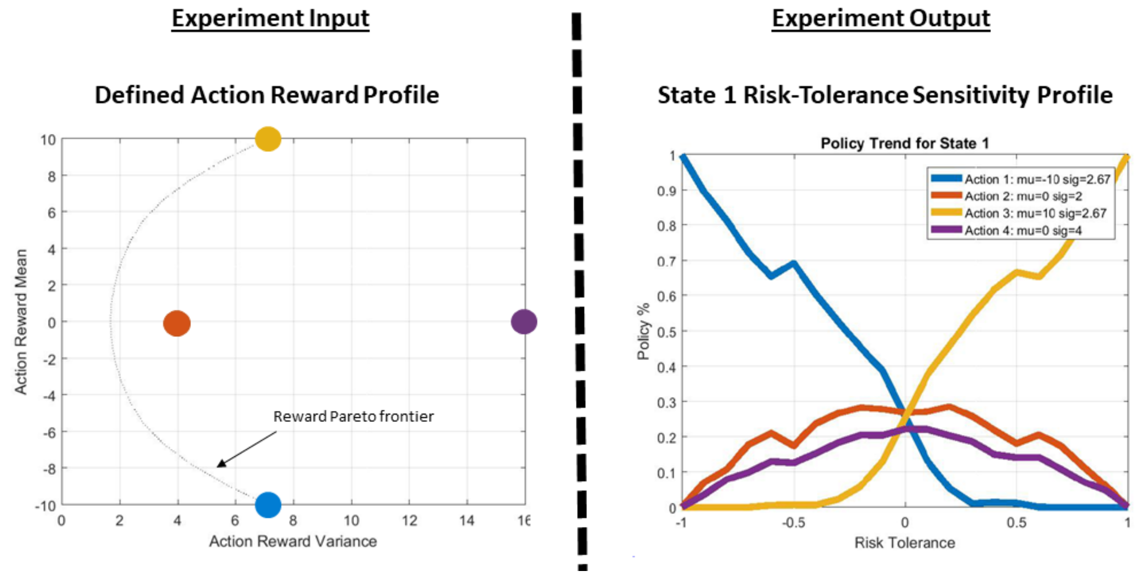


Figure C.3: Annotated Three-Action with One Mild-Inefficient Action Decision Space and RTSP

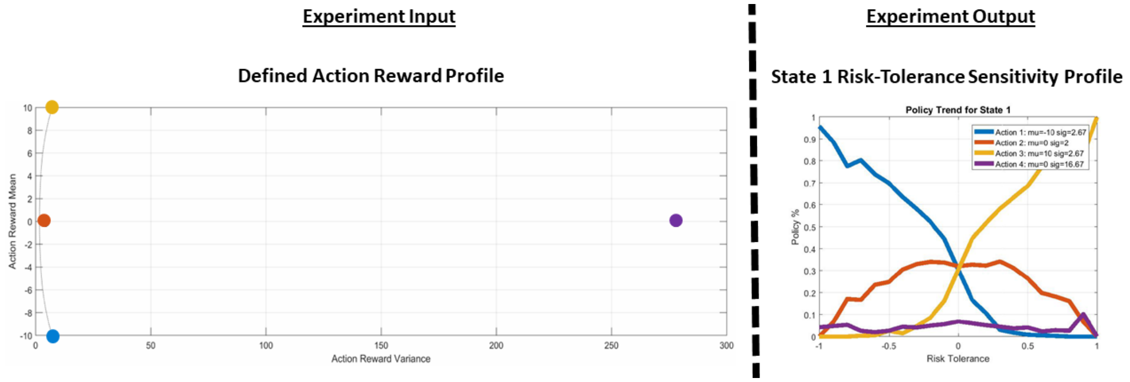


Figure C.4: Annotated Three-Action with One Significant-Inefficient Action Decision Space and RTSP

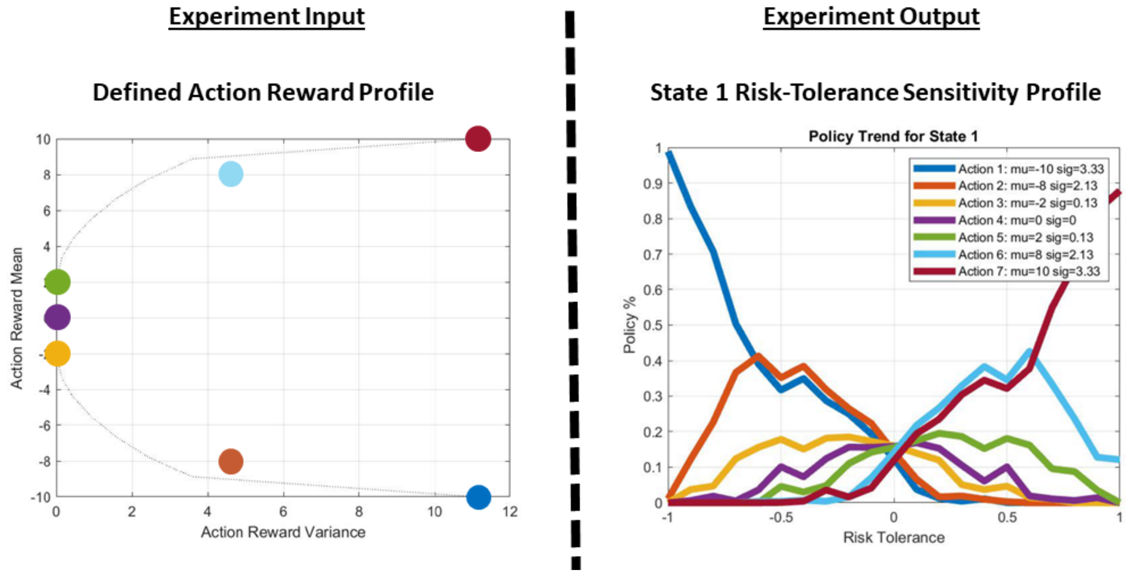


Figure C.5: Annotated Seven Action Pareto Efficient Decision Space and RTSP

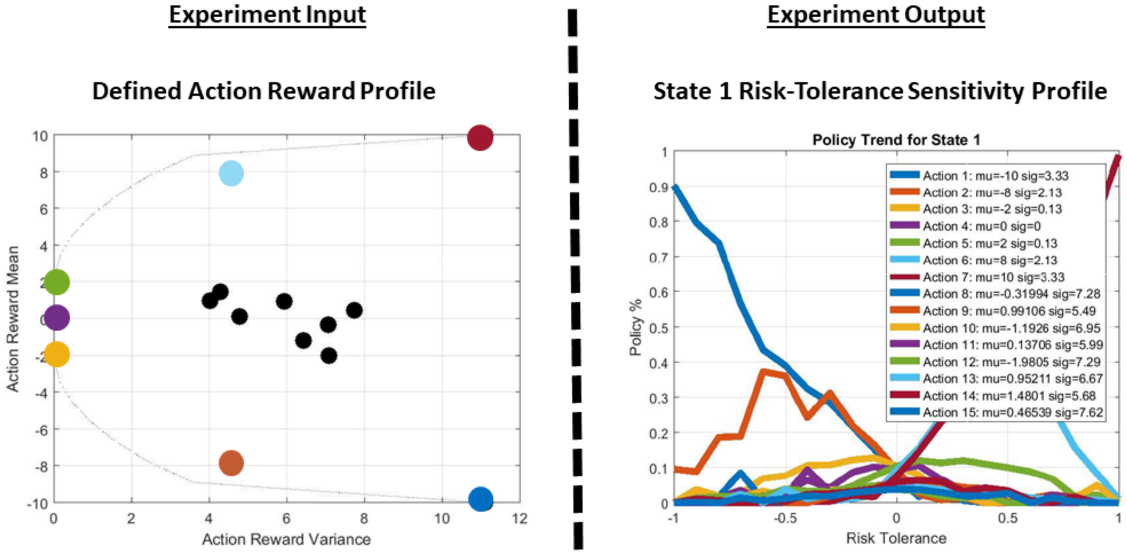


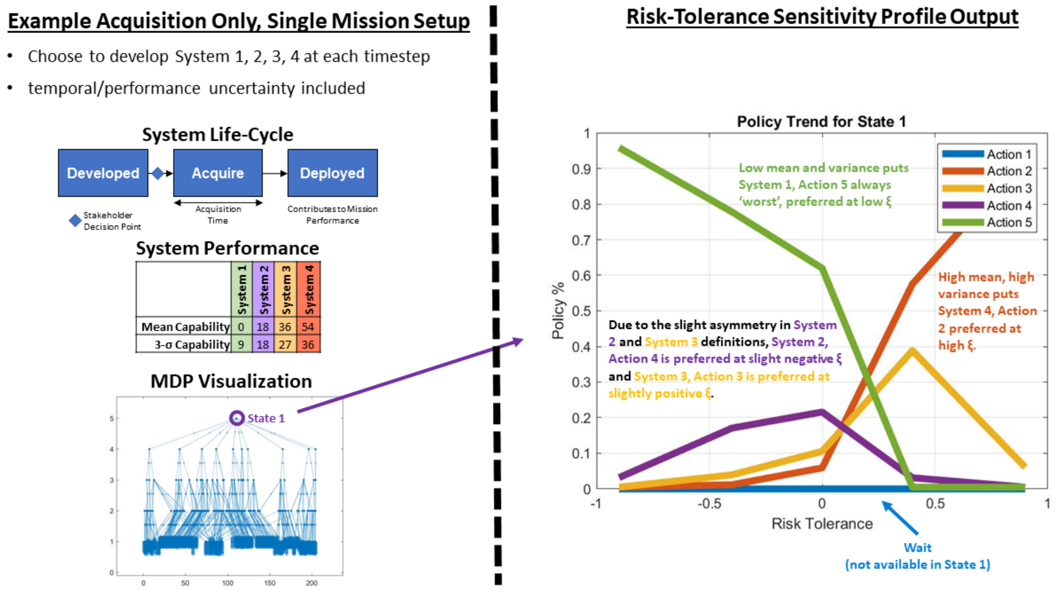
Figure C.6: Annotated Seven Action Pareto Efficient with Inefficient Actions Decision Space and RTSP

four systems. The systems performance is depicted by a set mean and variance designed to vary similar to those seen in the first example problem set. The resulting RTSP follows expectations built from the previous simpler examples.

The initial state can be analyzed as all future states are essentially repetitions of the first. Once again, this allows an equation of short term Reward and long term Return. Action 1 is a ‘wait’ action that occurs when an acquisition is underway and no additional acquisition action can be taken. The action is not available in the initial state or when an acquisition can be made. Action 5 is preferred near the worst case ($\xi\ -1$) and Action 2 is preferred near the high-risk case ($\xi \approx 1$). This is in line with the low-mean low-variance of System 1 and the high-mean high-variance of System 4. The two intermediate actions show an interesting trend. There is a built in asymmetry from System 2 and System 3 performance. This results in an asymmetrical Reward and therefor an asymmetrical Return. This asymmetry is represented in the RTSP as well. Action 4, acquiring System 2, peaks just under and at a $\xi = 1$ with Action 3, acquiring System 3, peaks above and more significantly between $\xi = 0$ and $\xi = 1$. This demonstrates that Pareto efficient actions can be selected from a more complex repeated action scenario with relative nuanced information.

## C.3 Return versus Reward Examples

Two scenarios were selected similar to Experiment 1b Case 2 Scenario 2. The selected examples show the difference between the impact of short term Reward and long term Return. The setup is based on developing and acquiring systems in a sequential order with each system being more or less risky based on design. The baseline scenario results are shown in Figure C.8. The system performance setup results in a relative mean and variance Return that is nearly constant for all risk-tolerance levels. The mean-variance plot on the left shows the Return mean and variance as the risk tolerance is varied. There is little relative motion. This results in an RTSP that is similar to what was observed in Figure C.1 in the simplest of scenarios.



**Example Acquisition Only, Single Mission Setup**

- Choose to develop System 1, 2, 3, 4 at each timestep
- temporal/performance uncertainty included

**System Life-Cycle**

**System Performance**

| | System 1 | System 2 | System 3 | System 4 |
|---|---|---|---|---|
| Mean Capability | 0 | 18 | 36 | 54 |
| 3-σ Capability | 9 | 18 | 27 | 36 |

**MDP Visualization**

**Risk-Tolerance Sensitivity Profile Output**

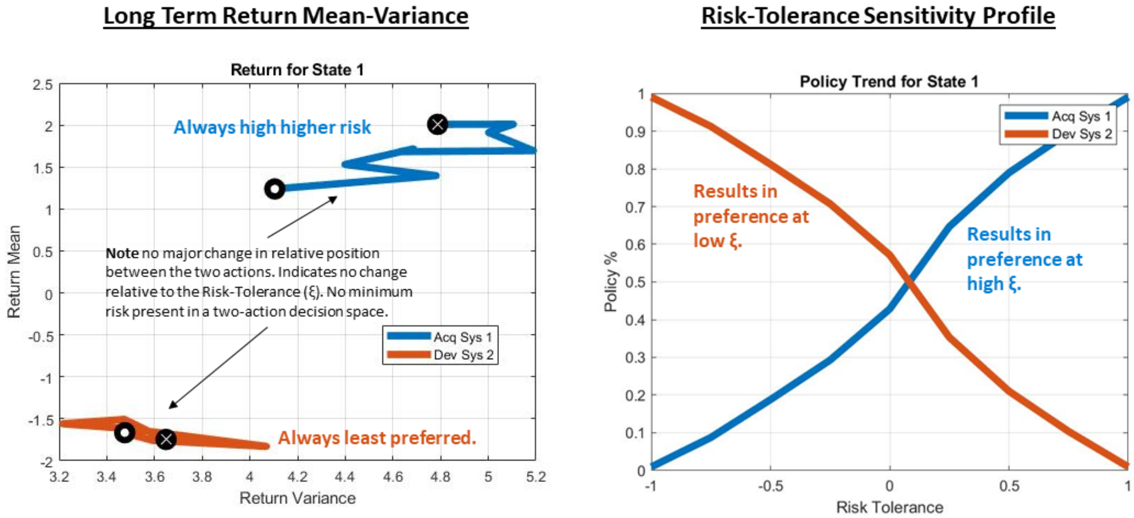
Low mean and variance puts System 1, Action 5 always 'worst', preferred at low ξ

High mean, high variance puts System 4, Action 2 preferred at high ξ.

Due to the slight asymmetry in System 2 and System 3 definitions, System 2, Action 4 is preferred at slight negative ξ and System 3, Action 3 is preferred at slightly positive ξ.

Wait
(not available in State 1)

Figure C.7: Annotated Repeated Acquisition Only Decision Space and RTSP



**Long Term Return Mean-Variance**

Always high higher risk

**Note** no major change in relative position between the two actions. Indicates no change relative to the Risk-Tolerance (ξ). No minimum risk present in a two-action decision space.

Always least preferred.

**Risk-Tolerance Sensitivity Profile**

Results in preference at low ξ.

Results in preference at high ξ.

Figure C.8: Annotated Baseline Acquire vs. Develop Decision Space and RTSP

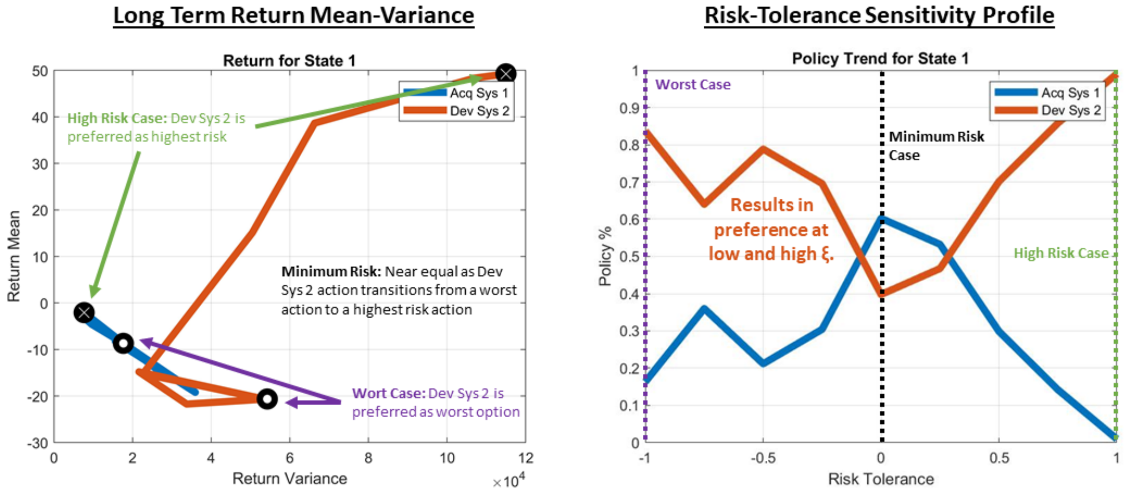
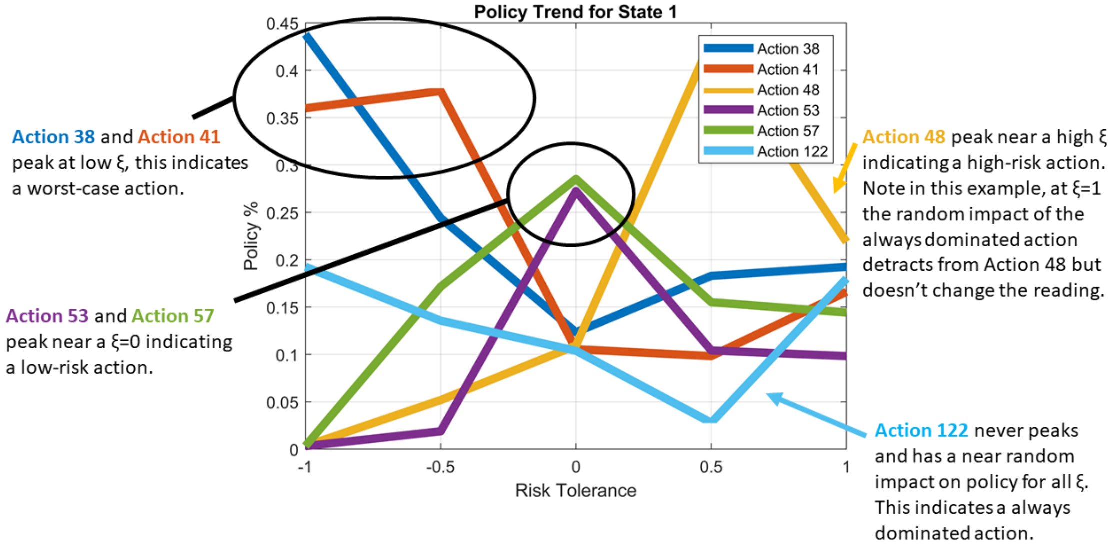


Figure C.9: Annotated Adjusted Acquire vs. Develop Decision Space and RTSP

In a second setup, the variance in return of System 2 is heavily increased. The results are shown in Figure C.9. The Return mean-variance plot on the left shows the worst point as a white circle in a black circle. The highest risk point as a black circle with an ‘x’. Note that the long term Return of selecting to develop System 2 moves from a higher variance lower mean position relative to acquiring system 1 to having a higher-variance and higher-mean. This means that as the risk-tolerance is changed, the relative weightings along the Pareto frontier of each action vary. When the risk-tolerance is low, system 2 development is in the worst position. When risk-tolerance is high, system 2 development is in the highest risk position. This is apparent in the RTSP on the right side of the figure. At a low and high $\xi$, developing system 2 is preferred. There is a spot near $\xi = 0$ where acquiring system 1 is preferred. The corresponding point in the Return mean-variance plot is when the returns of both are near equal and the variance of developing system 2 is less than that of developing system 1. This shows that some time in the future there is another decision point that has a significant impact on the mean and variance outcome. The results are based on the impact of using a worst, low-risk, or high-risk policy to make future decisions. These decisions present themselves in the Return seen at State 1 and impact the resulting RTSP.



Action 38 and Action 41 peak at low ξ, this indicates a worst-case action.

Action 53 and Action 57 peak near a ξ=0 indicating a low-risk action.

Action 48 peak near a high ξ indicating a high-risk action. Note in this example, at ξ=1 the random impact of the always dominated action detracts from Action 48 but doesn’t change the reading.

Action 122 never peaks and has a near random impact on policy for all ξ. This indicates a always dominated action.

Figure C.10: Annotated Complex RTSP Example

## C.4 Higher Complexity RTSP

The final example was selected from Experiment 3b and has been divorced from the setup to allow a direct interpretation to be evaluated. The RTSP is shown in Figure C.10.

There are four classes of actions that can be derived from the RTSP shown in Figure C.10. There are the worst actions, the low-risk actions, the high risk-actions, and the always dominated actions. Action 38 and 41 peak near a low risk-tolerance level and can be categorized as the worst actions. These actions should never be taken since they are always dominated. Actions 53 and 57 represent the low-risk options as they peak near a $\xi = 1$. These actions should be considered when a stakeholder has a low tolerance for risk. Action 48 peaks near a high risk-tolerance and can be categorized as a high-risk action. Action 122 never peaks and has a random impact as a function of risk-tolerance. This indicates the action falls in the non-efficient class. Note that the random increase near a high risk tolerance pulls from the impact of Action 48. Despite this, the overall trend still results in the high-risk and inefficient classifications given to Actions 48 and 122 respectively.

# APPENDIX D
# FULL COMPLEXITY PROBLEM REFERENCES

Appendix D contains content that supports the description of the setup, results, and analysis of the full complexity test problem. The full complexity test problem is used in Experiment 3b.

## D.1 System Quantities versus Time

The follow plots are created based on the raw sampling of the Truth Model for Experiment 3b (Section 6.3.2). The plots support the characterization of the Truth Model sampling for Experiment 3b which is depicted in Section 7.3.2.

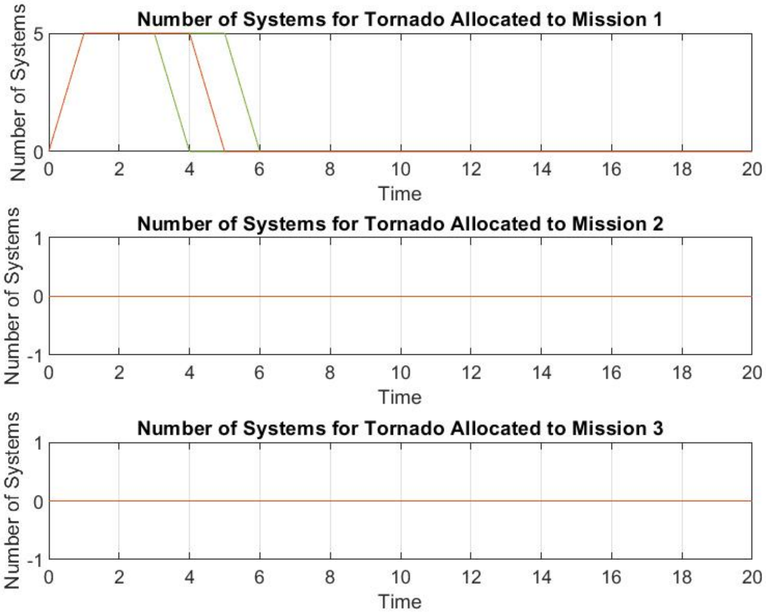


Figure D.1: Sampled Tornado Air Wings Deployed versus Time

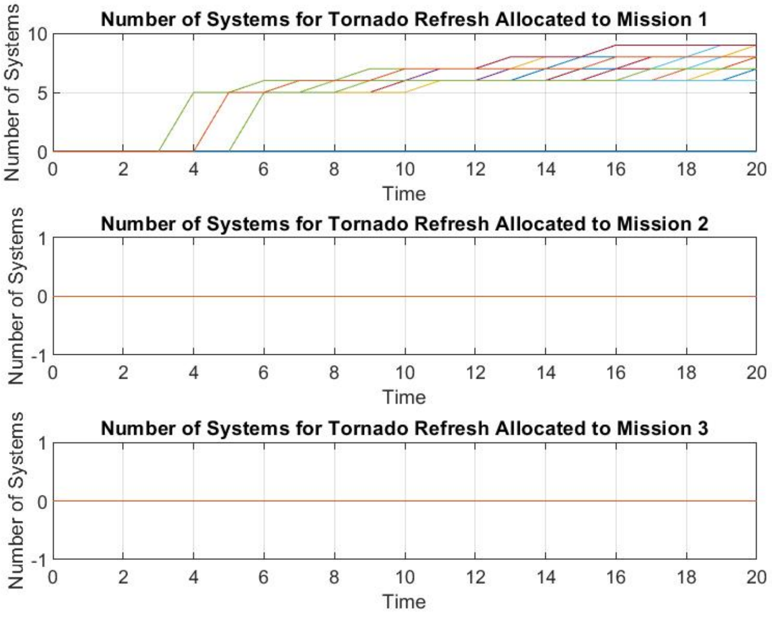


Figure D.2: Sampled Refreshed Tornado Air Wings Deployed versus Time

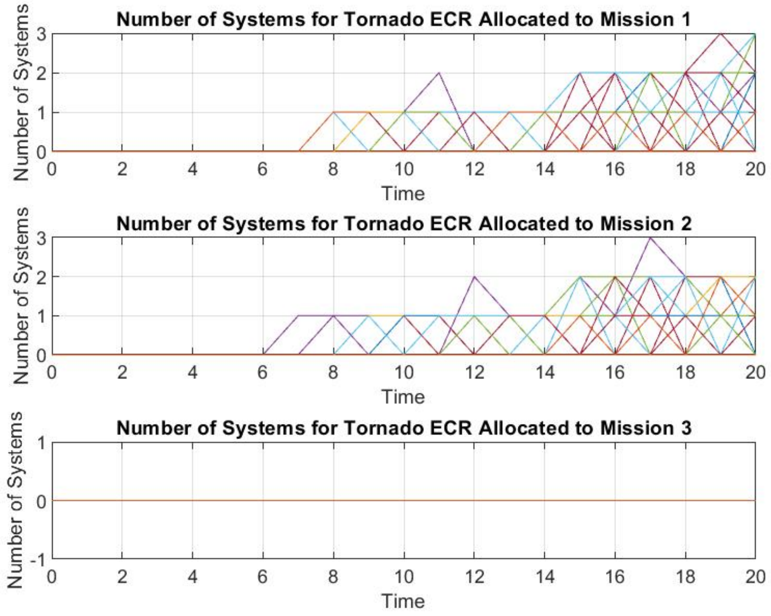


Figure D.3: Sampled Tornado ECR Air Wings Deployed versus Time

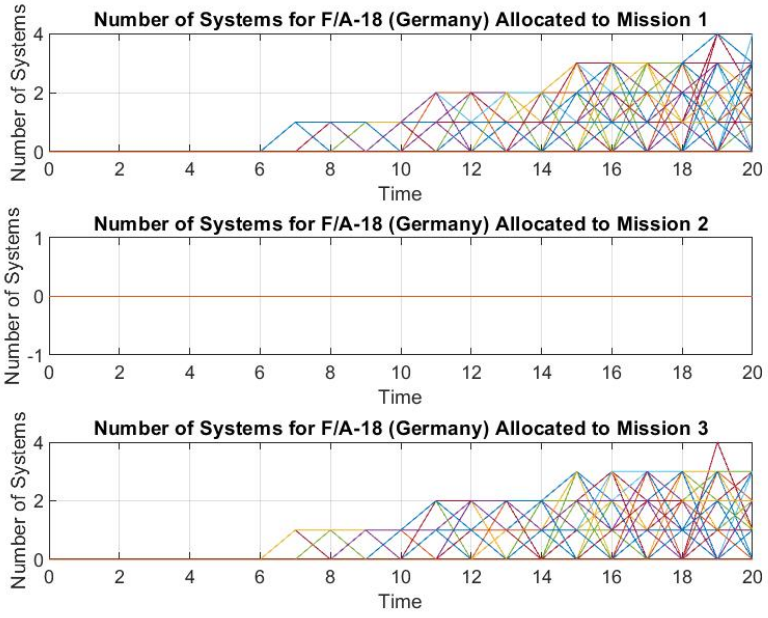


Figure D.4: Sampled F/A-18 Air Wings Deployed versus Time

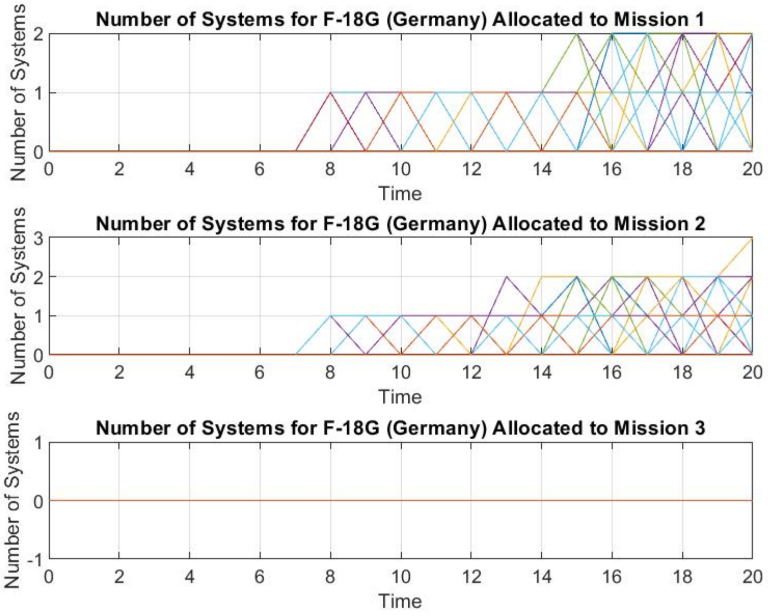


Figure D.5: Sampled F-18G Air Wings Deployed versus Time

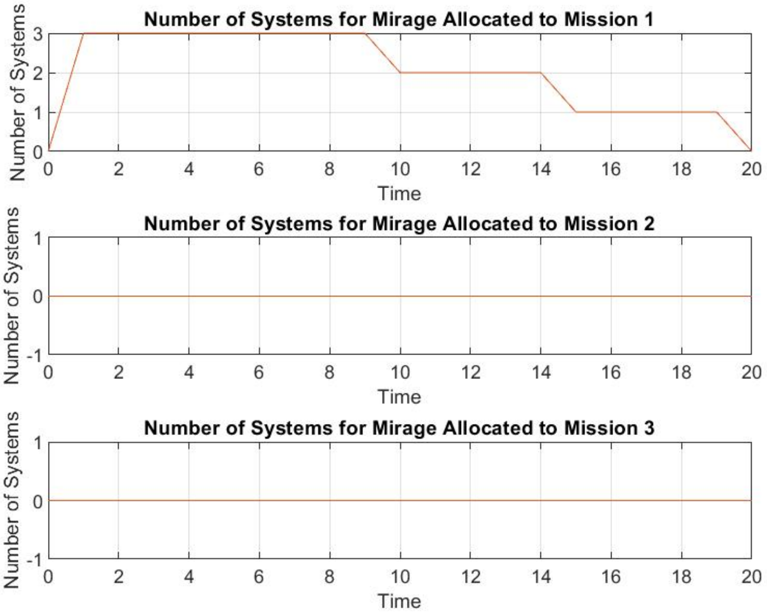


Figure D.6: Sampled Mirage Air Wings Deployed versus Time

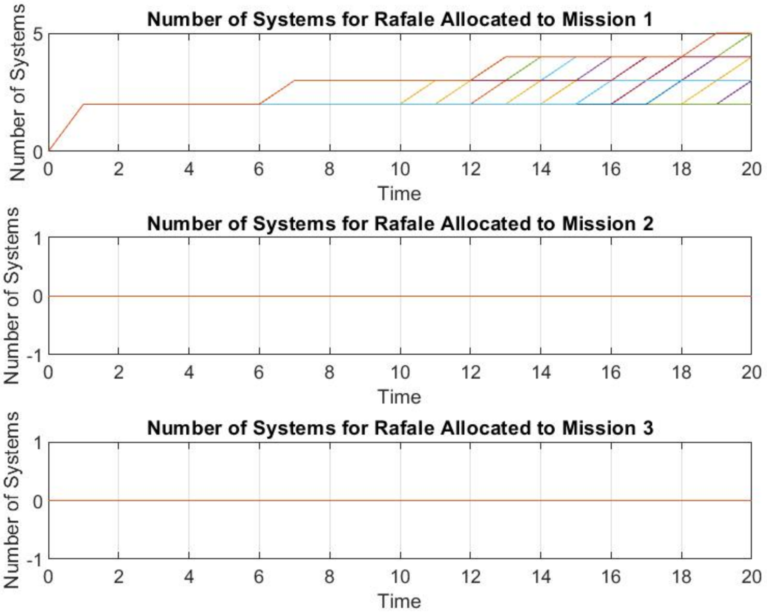


Figure D.7: Sampled Rafale Air Wings Deployed versus Time

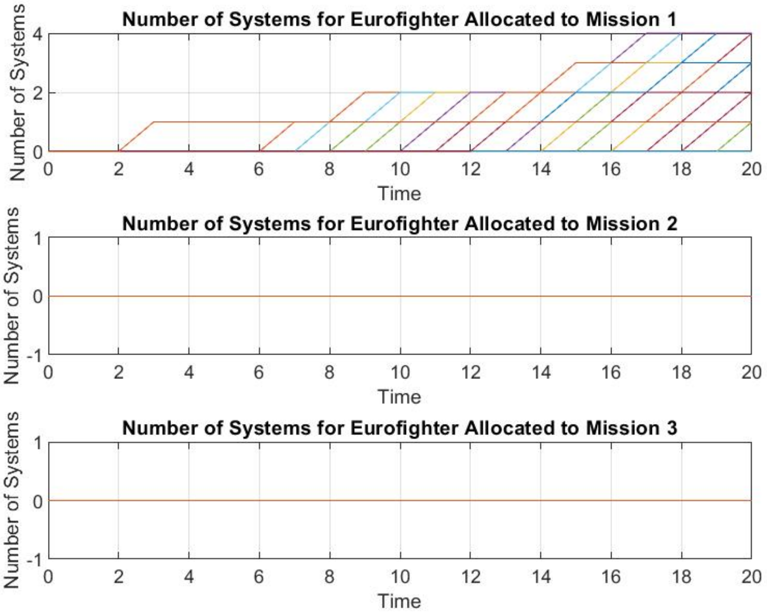


Figure D.8: Sampled Eurofighter Air Wings Deployed versus Time

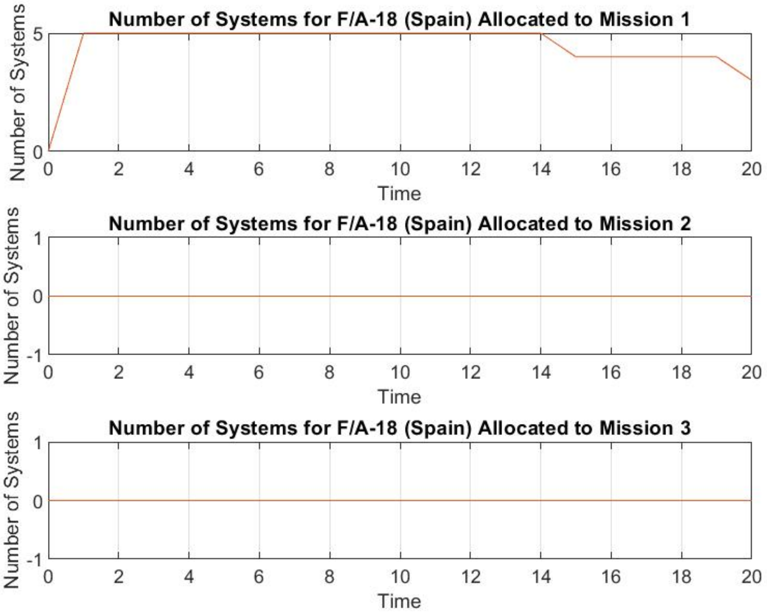


Figure D.9: Sampled F/A-18 Air Wings Deployed versus Time

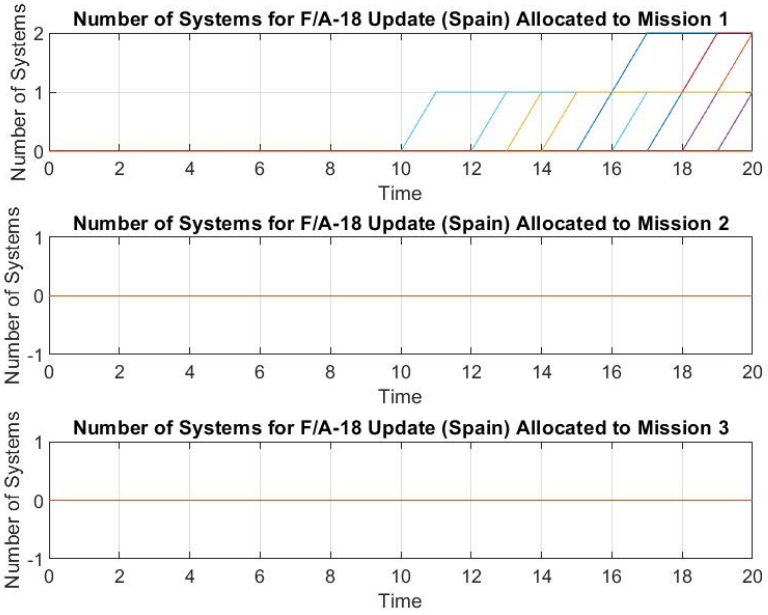


Figure D.10: Sampled F/A-18 Update Air Wings Deployed versus Time

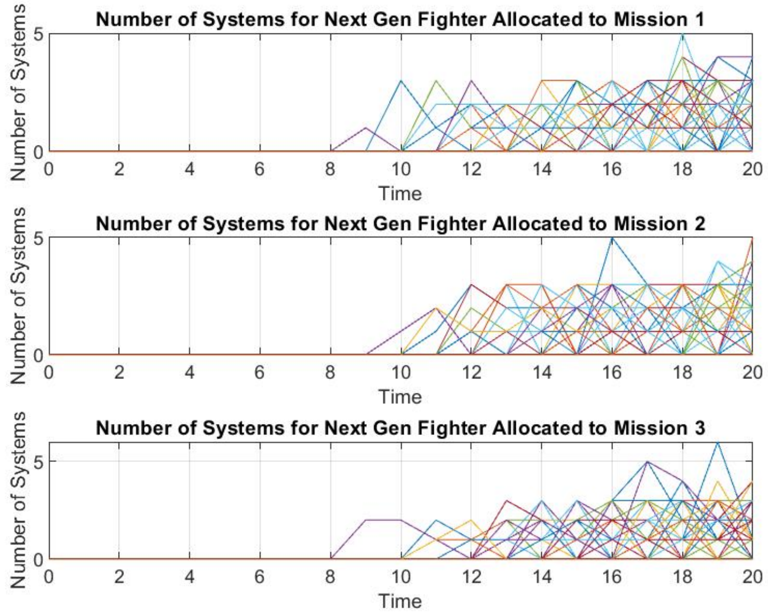


Figure D.11: Sampled Next Gen Fighter Air Wings Deployed versus Time

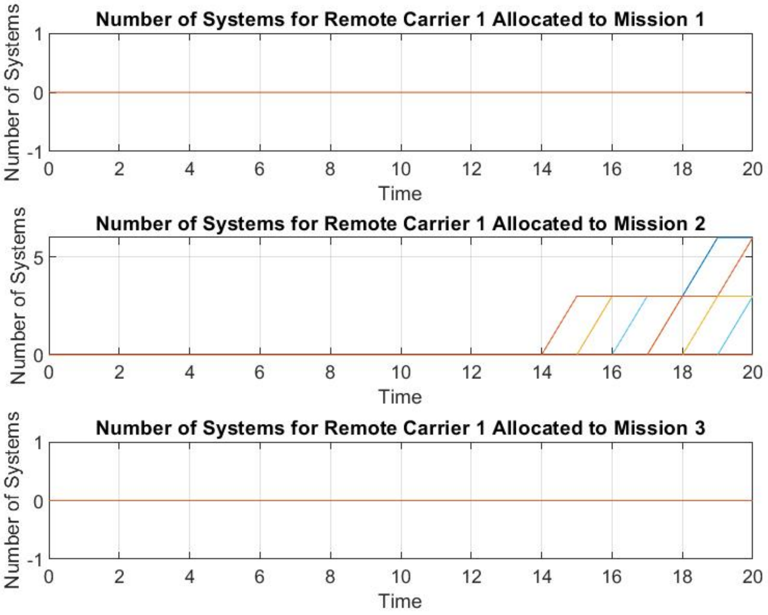


Figure D.12: Sampled Remote Carrier Air Wings Deployed versus Time

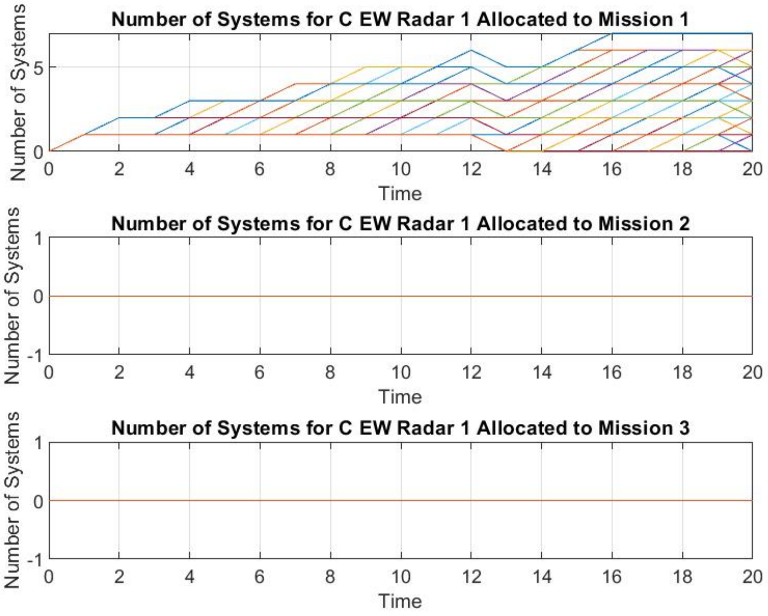


Figure D.13: Sampled Conventional EW Radars Deployed versus Time

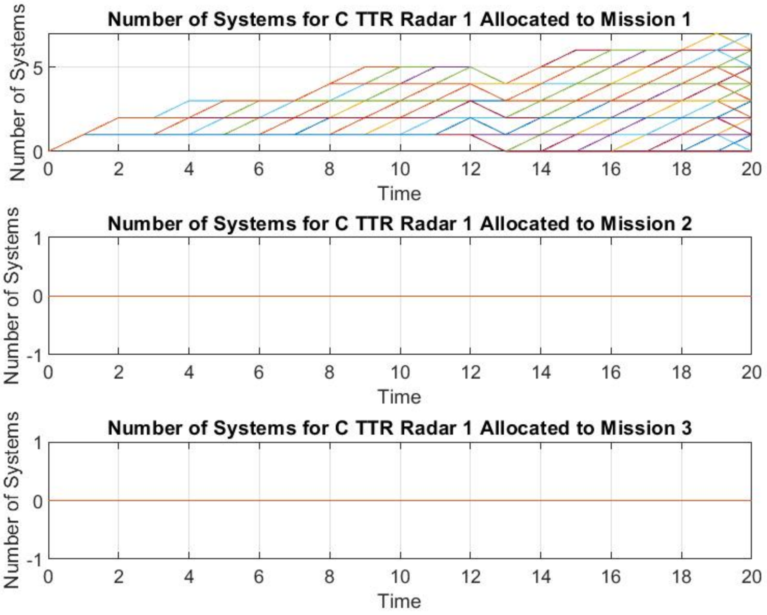


Figure D.14: Sampled Conventional TTR Radars Deployed versus Time

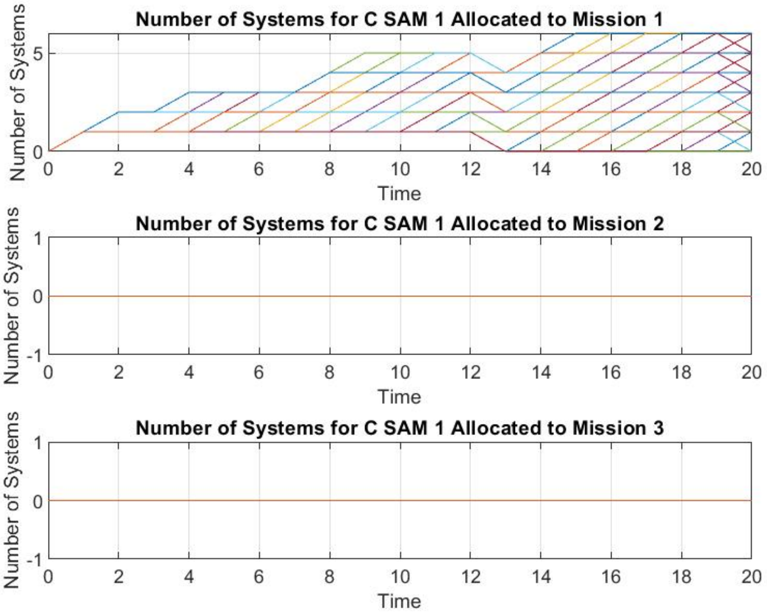


Figure D.15: Sampled Conventional SAM Systems Deployed versus Time

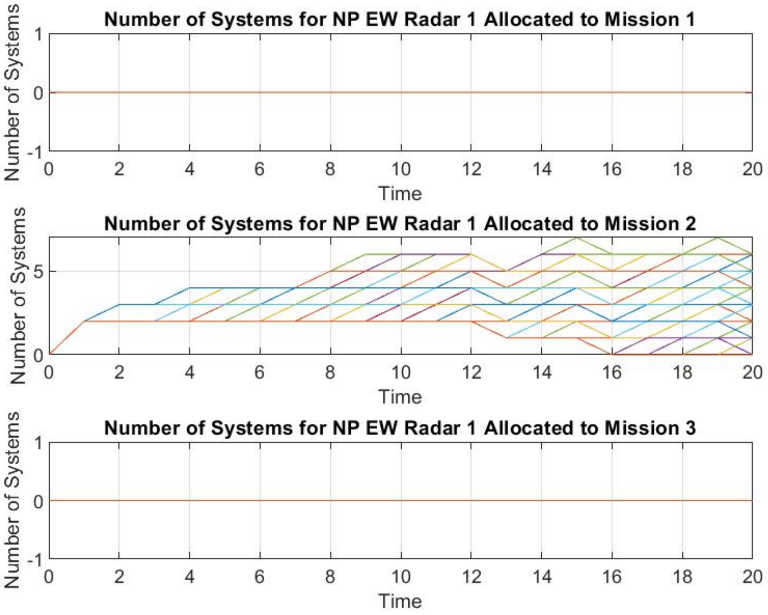


Figure D.16: Sampled Near Peer EW Radars Deployed versus Time

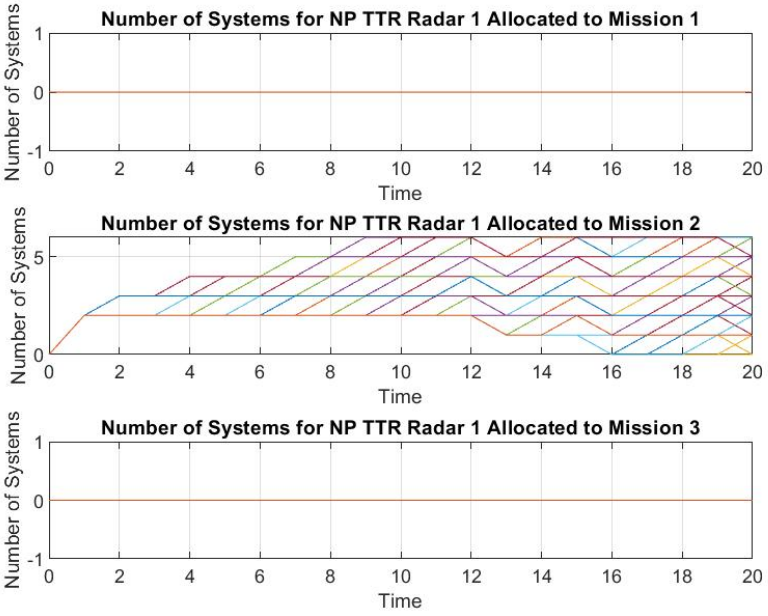


Figure D.17: Sampled Near Peer TTR Radars Deployed versus Time

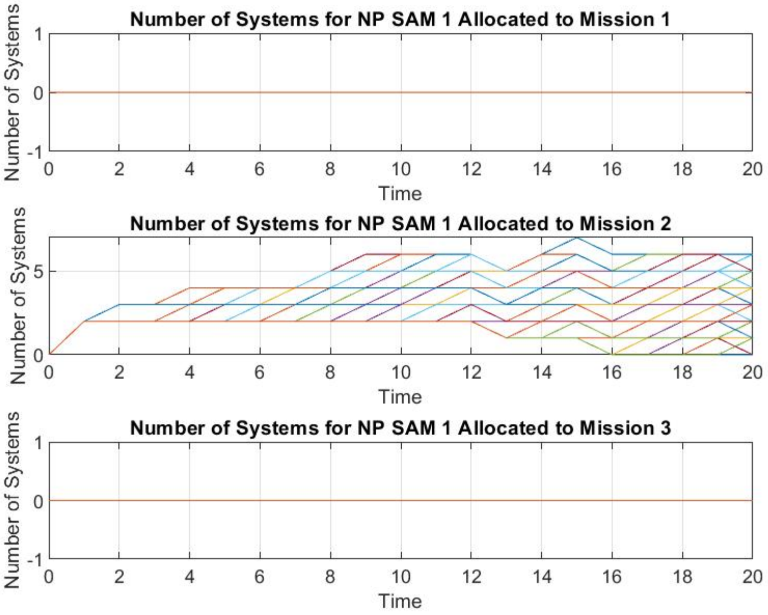


Figure D.18: Sampled Near Peer SAM Systems Deployed versus Time

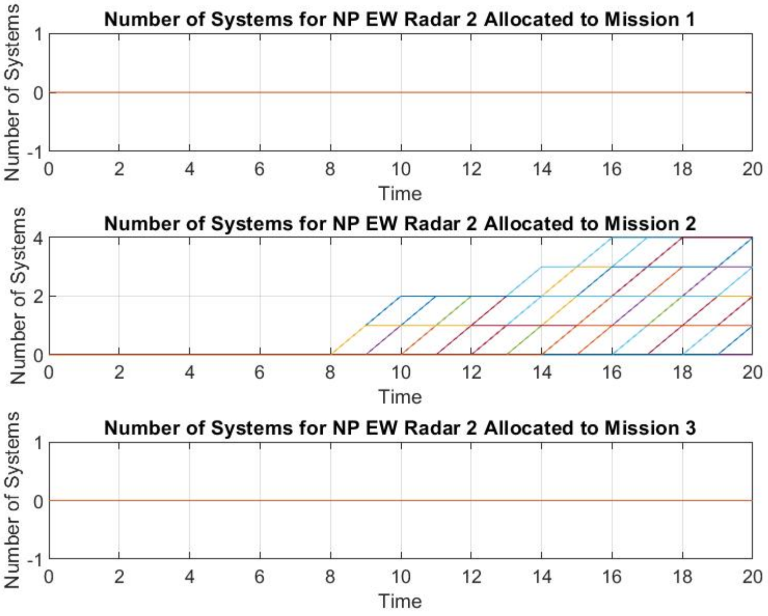


Figure D.19: Sampled Near Peer Next Gen EW Radars Deployed versus Time

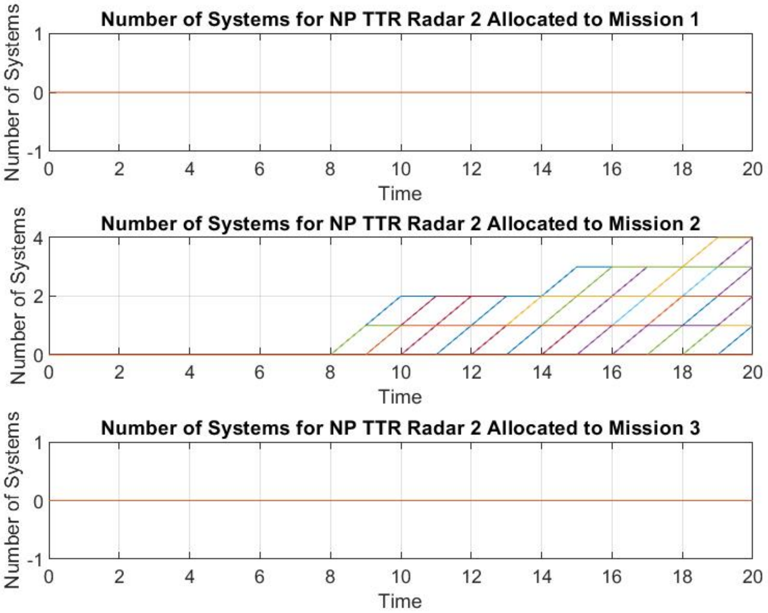


Figure D.20: Sampled Near Peer Next Gen TTR Radars Deployed versus Time

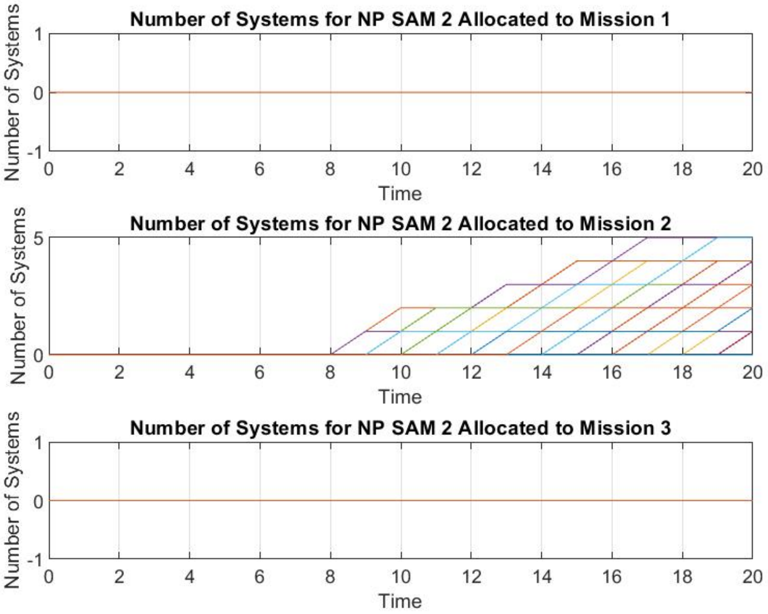


Figure D.21: Sampled Near Peer Next Gen SAM Systems Deployed versus Time

# REFERENCES

[1] M. J. Mazarr, K. L. Best, B. Laird, E. V. Larson, M. E. Linick, and D. Madden, "The U.S. Department of Defense's Planning Process: Components and Challenges," RAND, Tech. Rep., 2019.

[2] P. H. Liotta and R. M. Lloyd, "From Here to There: The Strategy and Force Planning Framework," *Naval War College Review*, vol. 58, no. 2, 2005.

[3] "CJCSI 3170.01H Joint Capabilities Integration and Development System," Department of Defense, Tech. Rep., 2012.

[4] "Manual for the Operation of the Joint Capabilities Integration and Development System (JCIDS)," Department of Defense, Tech. Rep. February, 2015.

[5] S. W. Wilson, "Strategic Master Plan 2014," Air Force Global Strike Command, Tech. Rep., 2014.

[6] "OMB Sequestration Update Report to the President and Congress for Fiscal Year 2015," Office of Management and Budget, Tech. Rep., 2014.

[7] "Estimated Impacts of Sequestration-Level Funding Fiscal Year 2015 Budget Request," United States Department of Defense, Tech. Rep., 2014.

[8] L. M. Williams and D. P. Wees, "FY2017 Defense Spending Under an Interim Continuing Resolution (CR): In Brief," Congressional Research Service, Tech. Rep., 2016.

[9] M. E. Manyin *et al.*, "Pivot to the Pacific? The Obama Administration's "Rebalancing" Toward Asia," Tech. Rep., 2012.

[10] C. Pellerin, *Hagel Announces New Defense Innovation, Reforms Efforts*, 2014.

[11] "Documenting and Assessing Lessons Learned Would Assist DOD in Planning for Future Budget Uncertainty," Government Accoutability Office, Tech. Rep. May, 2015.

[12] "Performance of the Defense Acquisition System 2014 Annual Report," Under Secretary of Defense, Acquisition, Technology, and Logistics, Tech. Rep., 2014.

[13] L. Carl and J. McCain, “Defense Acquisition Reform: Where do we go from here? A Compendium of Views by Leading,” Committee on Homeland Security and Governmental Affairs, Permanent Subcommittee on Investigations, Tech. Rep., 2014.

[14] “Assessments of Selected Weapon Programs,” Government Accountability Office, Tech. Rep. March, 2015.

[15] “Performance of the Defense Acquisition System 2016 Annual report,” Under Secretary of Defence, Acquisition, Technology, and Logistics, Tech. Rep., 2016.

[16] A. Lyle, *National Security Advisor Explains Asia-Pacific Pivot*, 2015.

[17] C. Pellerin, *Deputy Secretary: Third Offset Strategy Bolsters America’s Military Deterrence*, 2016.

[18] C. Hagel, “The Defense Innovation Initiative,” Secritary of Defense, Tech. Rep., 2014.

[19] C. Hagel, “Reagan National Defense Forum Keynote,” 2014.

[20] “Best Practices: An Integrated Portfolio Management Approach to Weapon System Investments Could Improve DOD’s Acquisition Outcomes,” Government Accountability Office, Tech. Rep. March, 2007.

[21] “Weapon System Acquisitions: Opportunities Exist to Improve the Department of Defense’s Portfolio Management,” Government Accountability Office, Tech. Rep. August, 2015.

[22] “Missed Trade-off Opportunities During Requirements Reviews,” Government Accountability Office, Tech. Rep. June, 2011.

[23] “DOD’s Requirements Determination Process Has Not Been Effective in Prioritizing Joint Capabilities,” Government Accountability Office, Tech. Rep., 2008.

[24] “Defense Management: Perspectives on the Involvement of the Combatant Commands in the Development of Joint Requirements,” Government Accountability Office, Tech. Rep. August 2010, 2011.

[25] M. J. Sullivan, “DOD Must Balance Its Needs with Available Resources and Follow an Incremental Approach to Acquiring Weapon Systems,” Government Accountability Office, Tech. Rep., 2009.

[26] “Opportunities to Reduce Potential Duplication in Government Programs, Save Tax Dollars, and Enhance Revenue,” United States Government Accountability Office, Tech. Rep. March, 2011.

[27] "Inelligence, Surveillance, and Reconnaissance: DOD Can Better Assess and Integrate ISR Capabilities and Oversee Development of Future ISR Requirements," Government Accountability Office, Tech. Rep. March, 2008.

[28] "Inelligence, Surveillance, and Reconnaissance: Actions Are Needed to Increase Integration and Efficiencies of DOD's ISR Enterprise," Government Accountability Office, Tech. Rep. June, 2011.

[29] K. H. Hicks, "Defense Strategy and the Iron Triangle of Painful Trade-offs," Cemter fpr Strategic and International Studies, Tech. Rep., 2017.

[30] M. F. Cancian, "Military Force Structure : Trade-offs, Trade-offs, Trade-offs," Center for Strategic and International Studies, Tech. Rep., 2018.

[31] J. P. Wong, "Balancing Immediate and Long-Term Defense Investments," Ph.D. dissertation, 2016.

[32] Wiley, Ed., *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, 4.0. Hoboken, NJ, 2015, ISBN: 9781118999400.

[33] "ISO/IEC 15288 Systems and software engineering — System life cycle processes," IEEE, Tech. Rep., 2008.

[34] E. Alonso, N Karcanias, and A. Hessami, "Multi-agent systems: A new paradigm for systems of systems," *ICONSE 2013: The Eighth International Conference on Systems*, no. c, pp. 8–12, 2013.

[35] L. A. N. Amaral and J. M. Ottino, "Complex networks," *The European Physical Journal B - Condensed Matter*, vol. 38, no. 2, pp. 147–162, 2004.

[36] J. Boardman and B. Sauser, "System of Systems - the meaning of of," in *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, Los Angeles, CA: IEEE, 2006.

[37] C. D. Sage, Andrew P. and Cuppan, "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," *Inf. Knowl. Syst. Manag.*, vol. 2, no. 4, pp. 325–345, 2001.

[38] C. B. Keating, "Research Foundations for System of Systems Engineering," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, IEEE, 2005, pp. 2720–2725.

[39] A Gorod, B Sauser, and J Boardman, "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward," *Systems Journal, IEEE*, vol. 2, no. 4, pp. 484–499, 2008.

[40] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. I. M. Woodcock, and J. A. N. Peleska, “Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions,” *ACM Computing Survey*, vol. 48, no. 2, 2015.

[41] M. W. Maier, “Architecting Principles for Systems-of-Systems,” *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[42] M. W. Maier, “Integrated Modeling: A unified approach to system engineering,” *Journal of Systems and Software*, vol. 32, no. 2, pp. 101–119, 1996.

[43] M. W. Maier and E. Rechtin, *The Art of Systems Architecting: Second Edition*. Washington, D.C.: CRC Press, 2000, ISBN: 0849304407.

[44] J. S. Dahmann, “Systems of Systems Characterization and Types (STO-EN-SCI-276),” NATO Science and Technonlogy Orgainization, Tech. Rep., 2015.

[45] D. DeLaurentis and W. Crossley, “A Taxonomy-based Perspective for Systems of Systems Design Methods,” *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 86–91,

[46] A Gorod, B Sauser, and J Boardman, “Paradox: Holarchical view of system of systems engineering management,” in *System of Systems Engineering, 2008. SoSE ’08. IEEE International Conference on*, 2008, pp. 1–6.

[47] M. W. Maier, “Architecting Principles for Systems-of-Systems,” *INCOSE International Symposium*, vol. 6, no. 1, pp. 565–573, 1996.

[48] “Systems Engineering Guide for Systems of Systems Essentials,” Department of Defense, Tech. Rep. December, 2010.

[49] J. Dahmann, G. Rebovich, R. Lowry, J. Lane, and K. Baldwin, “An Implementers’ View of Systems Engineering for Systems of Systems,” *IEEE Aerospace and Electronic Systems Magazine*, 2011.

[50] J. Dahmann, G. Rebovich, J. A. Lane, and R. Lowry, “System Engineering Artifacts for SoS,” in *Systems Conference, 2010 4th Annual IEEE*, IEEE, 2010, pp. 13–17.

[51] C. Haskins, K. Forsberg, M. Krueger, D. Walden, and R. D. Hamelin, “Systems Engineering Handbook,” INCOSE, Tech. Rep. October, 2011.

[52] *Systems Engineering Guide for Systems of Systems*, August. 2008, ISBN: 7036957417.

[53] J. A. Lane and R. Lowry, “Systems Engineering Artifacts for SoS,” no. April, 2010.

[54] J. A. Lane, "Key System of Systems Engineering Artifacts to Guide Engineering Activities October 2010," no. October, 2010.

[55] M. R. Kirby and D. N. Mavris, "Forecasting Technology Uncertainty in Preliminary Aircraft Design," *World Aviation Conference*, p. 14, 1999.

[56] M. R. Kirby and D. N. Mavris, "An approach for the intelligent assessment of future technology portfolios," *AIAA Aerospace Sciences Meeting Exhibit 40th Reno NV UNITED STATES 1417 Jan 2002*, no. January, p. 13, 2002.

[57] M. R. Kirby and D. N. Mavris, "A Technique for Selecting Emerging Technologies for a Fleet of Commercial Aircraft to Maximize R & D Investment," 2001.

[58] M. R. Kirby, "Technology Identification, Evaluation, and Selection for Commercial Transport Aircraft," in *58th Annual Society of Allied Weight Engineers Conference*, 1999.

[59] M. Kirby, "A methodology for technology identification, evaluation, and selection in conceptual and preliminary aircraft design," no. March, p. 255, 2001.

[60] D. N. Mavris, D. S. Soban, and M. C. Largent, "An Application of a Technology Impact Forecasting ( TIF ) Method to an Uninhabited Combat Aerial Vehicle," 1999.

[61] P. T. Biltgen, "A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems," Ph.D. dissertation, Georgia Institute of Technology, 2007.

[62] *About the Unified Modeling Language Specification Version 2.5.1*, 2017.

[63] *About the OMG System Modeling Language Specification Version 1.5*, 2017.

[64] S. Friedenthal, *A Practical Guide to SysML, Third Edition: The Systems Modeling Language*, r. Edition, Ed. Morgan Kaufmann, 2014, p. 630, ISBN: 978-0128002025.

[65] L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley, 2013, p. 304, ISBN: 0321927869.

[66] *DoD Architecture Framework Version 2.02*.

[67] C Piaszczyk, "Model Based Systems Engineering with Department of Defense Architectural Framework," *Systems Engineering*, vol. 14, no. 3, pp. 305–326, 2011.

[68] K. Griendling, F. Drive, and D. N. Mavris, "Development of a Dodaf-Based Executable Architecting Approach to Analyze System-of-Systems Alternatives," 2011.

[69] L. Li, Y. Dou, B. Ge, K. Yang, and Y. Chen, "Executable System-or-Systems architecting based on DoDAF Meta-model," in *International Conference on System of Systems Engineering (SoSE)*, vol. 7th, 2012.

[70] Z. Fang, D. DeLaurentis, and N. Davendralingam, "An Approach to Facilitate Decision Making on Architecture Evolution Strategies," *Procedia Computer Science*, vol. 16, pp. 275–282, 2013.

[71] K. A. Griendling, "ARCHITECT: The Architecture-Based Technology Evaluation and Capability Tradeoff Method," Ph.D. dissertation, Georgia Institute of Technology, 2011.

[72] A. K. Raz, D. A. Delaurentis, and W. Lafayette, "A System-of-Systems Perspective on Information Fusion Systems : Architecture Representation and Evaluation," no. January, pp. 1–13, 2015.

[73] E. Honour, "DANSE: Tools Training Participant Manual v2.1," 2014.

[74] E. Gamma, R. Helm, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Orientated Software*. Reading, Massachusetts: Addison-Wesley, 1995, p. 395, ISBN: 0201633612.

[75] H. Gomaa, *Software modeling and design : UML, use cases, architecture, and patterns*, 1st Editio. Cambridge, New York: Cambridge University Press, 2011, p. 578, ISBN: 0521764149.

[76] R. S. Kalawsky, D. Joannou, Y. Tian, and a. Fayoumi, "Using Architecture Patterns to Architect and Analyze Systems of Systems," *Procedia Computer Science*, vol. 16, pp. 283–292, 2013.

[77] J. W. Coleman *et al.*, "COMPASS Tool Vision for a System of Systems Collaborative Development Environment," in *Proceedings of the 7th International Conference on System of System Engineering, IEEE SoSE 2012*, ser. IEEE Systems Journal, vol. 6, 2012.

[78] *COMPASS Research Website*.

[79] C. Ingram, "Roadmap for Research in Model-Based SoS Engineering," COMPASS Research, Tech. Rep. 1.0, 2014, p. 121.

[80] J. W. Coleman and A. K. Malmos, "Final Simulator for CML User Manual," COMPASS Research, Tech. Rep. September, 2013, pp. 1–22.

[81] J.-F. Pétin, D. Evrot, G. Morel, and. P. Lamy, “Combining SysML and formal models for safety requirements verification,” in *ICSSEA 2010 22nd International Conference on Software & Systems Engineering and their Applications*, 2010.

[82] S. Perry *et al.*, “Final Report on SOS Architetural Models,” COMPASS Research, Tech. Rep. May, 2012, pp. 1–139.

[83] G. Taguchi, *Introduction to quality engineering: designing quality into products and processes*. Asian Productivity Organization, 1986, p. 191, ISBN: 9283310845.

[84] J. H. Saleh, D. E. Hastings, and D. J. Newman, “Flexibility in System Design and Implications for Aerospace Systems,” *Acta Astronautica*, vol. 53, no. 12, pp. 927–944, 2003.

[85] J. M. Lafleur, “A Markovian State-Space Framework for Integrating Flexibility into Space System Design Decisions,” Ph.D. dissertation, Georgia Institute of Technology, 2012.

[86] M. N. Cheng, J. W. Wong, C. F. Cheung, and K. H. Leung, “A scenario-based roadmapping method for strategic planning and forecasting: A case study in a testing, inspection and certification company,” *Technological Forecasting and Social Change*, vol. 111, pp. 44–62, 2016.

[87] R. Alizadeh, P. D. Lund, A. Beynaghi, M. Abolghasemi, and R. Maknoon, “An integrated scenario-based robust planning approach for foresight and strategic management with application to energy industry,” *Technological Forecasting and Social Change*, vol. 104, pp. 162–171, 2016.

[88] C. M. Raczynski, “A Methodology for Comprehensive Strategic Planning and Program Prioritization,” Ph.D. dissertation, Georgia Institute of Technology, 2008.

[89] P. K. Davis, *Lessons from RAND’s Work on Planning Under Uncertainty for National Security*. 2012, ISBN: 9780833076618.

[90] B. D. Lee and C. J. J. Paredis, “A Conceptual Framework for Value-Driven Design and Systems Engineering,” *Procedia CIRP*, vol. 21, pp. 10–17, 2014.

[91] J. Cheung, J. Scanlan, J. Wong, J. Forrester, H. Eres, and S. Briceno, “Application of Value-Driven Design to Commercial Aeroengine Systems,” *Journal of Aircraft*, vol. 49, no. 3, pp. 688–703, 2012.

[92] P. D. Collopy, “Value-Driven Design,” *Journal of Aircraft*, vol. 48, no. 3, pp. 749–760, 2011.

[93] F. Delsing, “Cost Capability Analysis: Introduction to a Technique,” *Defense Acquisition, Technology and Logistics*, no. September-October, pp. 12–15, 2015.

[94] D. Delaurentis, “Panel: Research on Complex Enterprise Systems of Systems Complex Adaptive Systems Conference,” 2013, pp. 1–13.

[95] D. Delaurentis and K. Marais, “Progress Towards an Analytic Workbench for SoS Architectures,” 2015, pp. 1–58.

[96] D. Delaurentis, N. Davendralingam, K. Marais, C. Guariniello, Z. Fang, and P. Uday, “An SoS Analytical Workbench Approach to Architectural Analysis and Evolution,” pp. 70–74, 2016.

[97] D. A. Delaurentis, *System of Systems Analytic Workbench Toolset*, 2015.

[98] N. Davendralingam *et al.*, “An Analytic Workbench Perspective to Evolution of System of Systems Architectures,” *Procedia Computer Science*, vol. 28, pp. 702–710, 2014.

[99] N. Davendralingam and D. DeLaurentis, “An Analytic Portfolio Approach to System of Systems Evolutions,” *Procedia Computer Science*, vol. 28, no. Cser, pp. 711–719, 2014.

[100] C. Guariniello and D. DeLaurentis, “Integrated Analysis of Functional and Developmental Interdependencies to Quantify and Trade-off Ilities for System-of-Systems Design, Architecture, and Evolution,” *Procedia Computer Science*, vol. 28, no. Cser, pp. 728–735, 2014.

[101] D. A. Delaurentis, “A Conditional Value-at-Risk Approach to Risk Management in System-of-Systems Architectures,” pp. 457–462, 2015.

[102] 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.

[103] D. A.DeLaurentis and D. K. Marais, “Assessing the Impact of Development Disruptions and Dependencies in Analysis of Alternatives of System-of-Systems,” System Engineering Research Center, Tech. Rep., 2013.

[104] J. Dahmann, “SoS Pain Points & Implications for MBSE,” no. January 2012, pp. 1–31, 2013.

[105] P. Uday and K. B. Marais, “Resilience-based System Importance Measures for System-of-Systems,” *Procedia Computer Science*, vol. 28, no. Cser, pp. 257–264, 2014.

[106] Z. Fang and D. DeLaurentis, "Dynamic Planning of System of Systems Architecture Evolution," *Procedia Computer Science*, vol. 28, no. Cser, pp. 449–456, 2014.

[107] N. Davendralingam, D. A. Delaurentis, and M. Jacobs, "A Perspective on Decision-Making Research in System of Systems Context," pp. 463–468, 2015.

[108] Z. Fang and D. De Laurentis, "Dynamic planning of system of systems architecture evolution," *Procedia Computer Science*, vol. 28, no. Cser, pp. 449–456, 2014.

[109] S. Agarwal, L. E. Pape, N. Kilicay-Ergin, and C. H. Dagli, "Multi-agent based architecture for acknowledged system of systems," *Procedia Computer Science*, vol. 28, pp. 1–10, 2014.

[110] L. Pape, S. Agarwal, K. Giammarco, and C. Dagli, "Fuzzy optimization of acknowledged system of systems meta-architectures for agent based modeling of development," *Procedia Computer Science*, vol. 28, pp. 404–411, 2014.

[111] S. Agarwal, R. Wang, and C. H. Dagli, "Complex Systems Design & Management," 2010.

[112] S. Agarwal *et al.*, "Flexible and intelligent learning architectures for SoS (FILA-SoS): Architectural evolution in systems-of-systems," *Procedia Computer Science*, vol. 44, no. C, pp. 76–85, 2015.

[113] S. Agarwal, C. H. Dagli, and L. E. Pape, "Complex Systems Design & Management," 2016.

[114] S. Agarwal, "Computational Intelligence Based Complex Adaptive System-of-Systems Architecture Evolution Strategy," Doctor of Philosophy, Missouri University of Science and Technology, 2015, p. 186.

[115] F. Yang, C. Dagli, and W. Wanga, "Cognition Evolutionary Computation for System-of-systems Architecture Development," *Procedia Computer Science*, vol. 6, pp. 40–45, 2011.

[116] J. Dahmann, "System of Systems Pain Points," pp. 108–121, 2013.

[117] J. N. Webb, *Game Theory: Decisions, Interaction, and Evolution*. 2007, ISBN: 9788578110796.

[118] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory*. Morgan & Claypool, 2008, ISBN: 9781598295931.

[119] T. S. Ferguson, "Two-Person Zero-Sum Games," in *Game Theory*, 1995, p. 212.

[120] R. P. Gilles, “The Cooperative Game Theory of Networks and Hierarchies,” in vol. 44, 2010, pp. 29–71, ISBN: 978-3-642-05281-1.

[121] K. M. Ramachandran and C. P. Tsokos, *Stochastic Differential Games: Theory and Applications*. Paris: Atlantis Press, 2012, ISBN: 9789491216466.

[122] C. D. Aliprantis and S. K. Chakrabarti, “Games and decision making,” p. 470, 2000.

[123] D. P. Bertsekas, “Dynamic Programming and Optimal Control 3rd Edition , Volume II by Chapter 6 Approximate Dynamic Programming Approximate Dynamic Programming,” *Control*, vol. II, pp. 1–200, 2010.

[124] T. S. Ferguson, “Game Theory Introduction,” in *Game Theory*, 1995, pp. 1–6.

[125] M. L. Littman, “Algorithms for Sequential Decision Making,” Ph.D. dissertation, Brown University, 1996, ISBN: 1-55860-274-7.

[126] K. Avrachenkov, L. Cottatellucci, and L. Maggi, “Cooperative Markov decision processes: Time consistency, greedy players satisfaction, and cooperation maintenance,” *International Journal of Game Theory*, vol. 42, no. 1, pp. 239–262, 2013.

[127] J. L. Salmon, “A Methodology for Quantitative and Cooperative Decision Making of Air Mobility Operational Solutions,” Ph.D. dissertation, Georgia Institute of Technology, 2013.

[128] M. W. Maier, “Research Challenges for Systems-of-Systems Context : Collaborative Systems,”

[129] Z. Fang, “Multi-Stakeholder Dynamic Optimization Framework for System-of-Systems Development and Evolution,” Ph.D. dissertation, Purdue University, 2017.

[130] N. Karcanias, “Systems of Systems: A Control Theoretic View,” *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1732–1737, 2013.

[131] I. Nikolic, “Framework for Understanding and Shaping Systems of Systems The case of industry and infrastructure development in seaport regions .,” 2007.

[132] C. Boutilier and S. Hanks, “Decision-Theoretic Planning : Structural Assumptions and Computational Leverage,” vol. 11, pp. 1–94, 1999.

[133] W. Yeoh, A. Kumar, and S. Zilberstein, “Automated generation of interaction graphs for value-factored Dec-POMDPs,” *IJCAI International Joint Conference on Artificial Intelligence*, pp. 411–417, 2013.

[134] C. Amato, J. S. Dibangoye, and S. Zilberstein, “Incremental Policy Generation for Finite-Horizon DEC-POMDPs.,” *Icaps*, pp. 2–9, 2009.

[135] F. Wu, S. Zilberstein, and X. Chen, “Online planning for multi-agent systems with bounded communication,” *Artificial Intelligence*, vol. 175, no. 2, pp. 487–511, 2011.

[136] J. Goldsmith and M. Mundhenk, “Competition Adds Complexity,” *Advances in Neural Information Processing Systems 20*, pp. 561–568, 2008.

[137] M. Spaan, C. Amato, G. Chalkiadakis, P. Doshi, and A.-I. Mouaddib, “Fifth Workshop on Multi-agent Sequential Decision Making in Uncertain Domains (MSDM),” *Science*, 2010.

[138] M. T. J. Spaan and F. A. Oliehoek, “The MultiAgent Decision Process toolbox: Software for decision-theoretic planning in multiagent-systems,” *Proceedings of the Third AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains (MSDM)*, no. May, pp. 107–121, 2008.

[139] E. Solan, “Stochastic Games,” pp. 3064–3074,

[140] T. Kemmerich and H. K. B, “A Convergent Multiagent Reinforcement Learning Approach for a Subclass of Cooperative Stochastic Games,” pp. 37–53, 2012.

[141] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer, 1997, p. 393, ISBN: 9781461284819.

[142] N. Hughes, “Solving large stochastic games with reinforcement learning,” vol. 2601, 2015.

[143] L. A. Prashanth and M. Ghavamzadeh, “Variance-constrained actor-critic algorithms for discounted and average reward MDPs,” *Machine Learning*, vol. 105, no. 3, pp. 367–417, 2016.

[144] D. G. Luenberger, *Investment Science*. New York: Oxford University Press, 1998, ISBN: 0195108094.

[145] A. Tamar, “Risk-Sensitive and Efficient Reinforcement Learning Algorithms,” Ph.D. dissertation, 2015, p. 134.

[146] H. M. Markowitz, *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press, 1959, p. 368, ISBN: 9780300013726.

[147] A. Tamar, D. Di Castro, and S. Mannor, “Learning the variance of the reward-to-go,” *Journal of Machine Learning Research*, vol. 17, pp. 1–36, 2016.

[148] A. A. Gosavi, "Variance-Penalized Markov Decision Processes: Dynamic Programming and Reinforcement Learning Techniques," *International Journal of General Systems*, 2014.

[149] M. Dehmer and A. Mowshowitz, "A history of graph entropy measures," *Information Sciences*, vol. 181, no. 1, pp. 57–78, 2011.

[150] M. V. Volkenstein, *Entropy and Information*, 25. 2009, vol. 104, pp. 10 335–10 339, ISBN: 978-3-0346-0077-4.

[151] P. K. Mary A. Bone Robert Cloutier and A. Carrigy, "System Architecture: Complexities Role in Architecture Entropy," in *5th International Conference on System of Systems Engineering*, 2010, pp. 1–6.

[152] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2009, p. 658, ISBN: 9781118029152.

[153] M. Campbell, A. J. Hoane, and F. H. Hsu, "Deep Blue," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002.

[154] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[155] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[156] M. Wiering and M. van Otterlo, *Reinforcement Learning: State-of-the-Art*. 2013, pp. 1–653, ISBN: 9783642015267.

[157] M. Jaderberg *et al.*, "Human-level performance in 3D multiplayer games with population- based reinforcement learning," vol. 865, no. May, pp. 859–865, 2019.

[158] A. G. B. Richars S. Sutton, *Reinforcement Learning*, Second. Cambridge, Massahusetts: The MIT Press, 2018, ISBN: 9780262039246.

[159] H. J. Van den Herik, J. W. Uiterwijk, and J. Van Rijswijck, "Games solved: Now and in the future," *Artificial Intelligence*, vol. 134, no. 1-2, pp. 277–311, 2002.

[160] C.-y. Wei, Y.-T. Hong, and C.-j. Lu, "Online Reinforcement Learning in Stochastic Games," *Advances in Neural Information Processing Systems*, vol. 31, no. Nips, 2017.

[161] S. Kapoor, "Multi-Agent Reinforcement Learning: A Report on Challenges and Approaches," pp. 1–24, 2018.

[162] H. M. Schwartz, *Multi-Agent Learning: A Reinforcement Approach*. John Wiley & Sons, Incorporated, 2014, H. M. Schwartz, ISBN: 9781118884478.

[163] M. L. Littman, *Markov games as a framework for multi-agent reinforcement learning*, 1910.

[164] H. M. S. Schwartz and H. M, *Multi-Agent Machine Learning : A Reinforcement Approach*, I. John Wiley & Sons, Ed. 2014, ISBN: 9781118362082.

[165] M. L. Littman, "Friend-or-Foe Q-Learning in General-Sum Games," in *International Conference on Machine Learnings*, 2001.

[166] A. Greenwald, "Correlated- Q Learning," *Proc. of the 20th International Conference on Machine Learning*, vol. 1, pp. 1–30, 2003.

[167] H. Qiao, J. Rozenblit, F. Szidarovszky, and L. Yang, "Multi-agent learning model with bargaining," *Proceedings - Winter Simulation Conference*, pp. 934–940, 2006.

[168] C. H. C. Ribeiro, R. Pegoraro, and A. H. R. Costa, "Experience Generalization for Concurrent Reinforcement Learners: the Minimax-QS Algorithm," *International Joint Conference on Autonomous Agents and Multi-Agent Systems AAMAS'2002*, pp. 1239–1245, 2002.

[169] T. Degris and O. Sigaud, "Factored Markov Decision Processes," *Markov Decision Processes in Artificial Intelligence: MDPs, beyond MDPs and applications*, pp. 99–126, 2013.

[170] O. Sykora, "State-space Dimensionality Reduction in Markov Decision Processes," *WDS'08 Proceedings of Contributed Papers*, vol. Part I, pp. 165–170, 2008.

[171] C. Boutilier, R. Dearden, and M. Goldszmidt, "Exploiting Structure in Policy Construction," *Proc. Fourteenth International Conference on AI (IJCAI-95)*, 1995.

[172] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pp. 195–210, 1996.

[173] C. Boutilier, R. Dearden, and M. Goldszmidt, "Stochastic dynamic programming with factored representations," *Artificial Intelligence*, vol. 121, no. 1, pp. 49–107, 2000.

[174] T. Dean and R. Givan, "Model Minimization in Markov Decision Processes," *AAAI-97 Procedings*, 1997.

[175] T. Dean, R. Givan, K.-e. Kim, T. Dean, and K.-e. Kim, “Solving Stochastic Planning Problems With Large State and Action Spaces,” 1998.

[176] C. Guestrin, D. Koller, and R. Parr, “Multiagent Planning with Factored MDPs,” in *Neural Information Processing Systems 14*, 2001.

[177] A. Kumar, S. Zilberstein, and M. Toussaint, “Scalable Multiagent Planning Using Probabilistic Inference,” *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2140–2146, 2011.

[178] H. Murao and S. Kitamura, “Q-Learning with adaptive state segmentation (QLASS),” pp. 179–184, 2002.

[179] S. Mishra, “Unsupervised Learning and Data Clustering,” *Towards Data Science*, pp. 1–18, 2017.

[180] C. Fraley, “How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis,” *The Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.

[181] O. Kramer, *Studies in Big Data 20 Machine Learning for Evolution Strategies*. 2016, p. 120, ISBN: 9783319333816.

[182] P. Gupta, “Robust clustering algorithms,” Ph.D. dissertation, Georgia Institute of Technology, 2011, p. 183.

[183] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, 1. 2009, vol. 3, pp. 1–130, ISBN: 9781598295474.

[184] R. R. Curtin, “Improving Duel-Tree ALgorithms,” Ph.D. dissertation, Georgia Institute of Technology, 2015.

[185] A. Subramanya, P. Pratim, and P. Talukdar, *Graph-Based Semi-Supervised Learning*. Morgan &Claypool Publishers, 2014, p. 127, ISBN: 9781627052016.

[186] R. Emery-montemerlo, G. Gordon, J. Schneider, and S. Thrun, “Approximate Solutions For Partially Observable Stochastic Games with Common Payoffs,”

[187] AIRBUS, *Future Combat Air System (FCAS)*, 2021.