Designing Systems for Many Possible Futures: The RSC-based Method for Affordable Concept Selection (RMACS), with Multi-Era Analysis

by

Michael Andrew Schaffner

B.S. Systems Engineering
University of Arizona, 2012

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology
June 2014

Signature of Author.....
Department of Aeronautics and Astronautics
May 22, 2014

Accepted by.....
Adam M. Ross
Research Scientist, Engineering Systems
Lead Research Scientist, Systems Engineering Advancement Research Initiative
Thesis Co-Advisor

Certified by.....
Donna H. Rhodes
Principal Research Scientist and Senior Lecturer, Engineering Systems
Director, Systems Engineering Advancement Research Initiative
Thesis Supervisor

Accepted by.....
Deborah Nightingale
Professor of Aeronautics and Astronautics and Engineering Systems
Aeronautics and Astronautics Academic Advisor

Accepted by.....
Paulo Lozano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Abstract

The current downward trend in funding for U.S. defense systems seems to be on a collision course with the state of the practice in systems engineering, which typically results in the increased pace and scale of capabilities and resultantly increased cost of complex national defense systems. Recent advances in the state of the art in systems engineering methodology can be leveraged to address this growing challenge. The present work leverages advanced constructs and methods for early-phase conceptual design of complex systems, when committed costs are still low and management influence is still high. First, a literature review is presented of the topics relevant to this work, including approaches to the design of affordable systems, assumptions and methods of exploratory modeling, and enabling techniques to help mitigate the computational challenges involved. The types, purposes, and limits of early-phase, exploratory models are then elucidated. The RSC-based Method for Affordable Concept Selection (RMACS) is described, which comprises nine processes in the three main thrusts of information gathering, evaluation, and analysis. The method is then applied to a naval ship case example, described as the Next-Generation Combat Ship, with representational information outputs and discussions of affordability with respect to each process. The ninth process, Multi-Era Analysis (MERA), is introduced and explicated, including required and optional informational components, temporal and change-related considerations, required and optional activities involved, and the potential types of outputs from the process. The MERA process is then applied to a naval ship case example similar to that of the RMACS application, but with discrete change options added to enable a tradespace network. The seven activities of the MERA process are demonstrated, with the salient outputs of each given and discussed. Additional thoughts are presented on MERA and RMACS, and 8 distinct areas are identified for further research in the MERA process, along with a brief description of the directions that such research might take. It is concluded that the affordability of complex systems can be better enabled through a conceptual design method that incorporates MERA as well as metrics such as Multi-Attribute Expense, Max Expense, and Expense Stability. It is also found that affordability of changeable systems can be better enabled through the use of existing path-planning algorithms in efficient evaluation and analysis of long-term strategies. Finally, it is found that MERA enables the identification and analysis of path-dependent considerations related to designs, epochs, strategies, and change options, in many possible futures.

Thesis Supervisor: Dr. Donna H. Rhodes
Title: Principal Research Scientist and Senior Lecturer, Engineering Systems
Director, Systems Engineering Advancement Research Initiative

Thesis Co-Advisor: Dr. Adam M. Ross
Title: Research Scientist, Engineering Systems

Acknowledgements

> "Selecting the one moment that changed your life, or the person that set you on the right or wrong track, is creating a story by selecting a memorable and interesting starting point…This can go back indefinitely, and any starting point would be exactly as relevant…There is no one person or event that completely changed my life. There are a few that stick out in memory…because they just made the best stories."
>
> -Peter Welch, *And Then I Thought I Was a Fish*

From a starting point of most recent memory:

I owe a great debt to my wife, Rachel, who has been a light to me in dark places, when all other lights went out. Though she could never carry my (academic) burdens for me, she did carry me. Rachel, I remain always yours and ever grateful for your support these past 6 years.

My lab- and classmates provided much of the value of the past 2 years. I hope to see all of you in the future less productive and less stressed, but perhaps wealthier (not necessarily monetarily). In no particular order: Marcus W, Nico R, Matt F, Brian M, Nirav S, Paul G, Hunter Z, Mike C, Paul L, Matt S, Pedro S, Margaret S, Caleb W, David S, Robbie H, Steven A, Brandon K, Johannes B, and Mark G, this includes you.

Kelvin and Pavan, your physical and virtual support and encouragement were always highly appreciated, even (especially?) when they were unexpected. Thanks.

Debbie, Donna, and Adam, I am grateful to you all for allowing me to join SEAri. Thank you for mentoring me and teaching me how to conduct research, and for supporting me even during my very confused moments. It has truly been a pleasure.

From slightly less recent memory:

Dr. Larry Head is truly inspirational, and I am quite confident I would be worse off having never met him. Dr. William Boynton and his team of planetary scientists provided an exceptional internship experience, without which I am sure my recent years would have taken a less stellar route. It is most assuredly a privilege to know Dr. Ricardo Valerdi.

In memoriam of Dr. Jerry Droege, whose optimism and enthusiasm were contagious.

Michael McKay's direct encouragement was a singular factor in my (rather late) decision to enter the academy; and Dan Hohn provided valuable mentorship in business and engineering. Dan Klein's tutelage prepared me to tilt at so many mathematical windmills.

And really stretching back:

Tim Barsuhn's singular influence saved this thesis from a host of vague antecedents.

Aaron and Megan were role models wise beyond their years. Could I ask for better siblings?

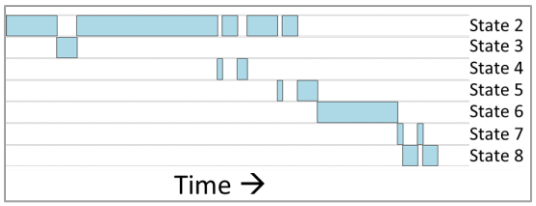
This thesis is dedicated to my parents, who taught me "to act justly…and to walk humbly" (Micah 6:8), and who always taught me to love knowledge, but to love wisdom more, "to do this, above all, to grasp what human will and human reason can do, and what they cannot" (Isaiah Berlin, *The Hedgehog and the Fox*).

Biographical Note

"The best laid plans of mike [sic] and men…"

-Aaron Schaffner, quoting Robert Burns' "To a Mouse"

Michael Schaffner was born in a paradisal seaside town on the west coast of Florida, where he was generally supremely happy for 8 years of his life. During that time he became fascinated with human space exploration, especially the International Space Station (ISS). In addition, he severely misinterpreted then-current technological capabilities from the movies *Short Circuit* and *Not Quite Human,* and as a result fixed his attentions from ages 4 through 6 on building a robot that could think on its own. Unfortunately, it was the 1980s.



Since that time, his life can perhaps best be summed up visually (see right; State 1, Florida, is not shown). Most of the travel in his life has been due to his love of seeing friends and family, who all apparently love living far from him. In more stationary years, he earned an Associates of Arts in Biblical Studies, started two technical support businesses, and met Rachel, by whom he was able to answer the age-old question, "Who can find a virtuous wife?" He also learned to play the saxophone and piano, nurtured a form of modernist Anglophilia, and dabbled in non-profit service, philosophy, soccer, ping pong, chess, hiking, fishing, and computer arts.

Eventually Michael came to study Systems Engineering at University of Arizona, during which time he worked as a summer intern in a simulation group at Raytheon and as a year-long NASA Space Grant Intern, where he contributed to a paper published in *JGR: Planets*. He also led a team in an international simulation competition, worked with BAE systems on the architecture for a cooperative unmanned vehicle software, and attended academic conferences, including presentation of a research poster to lawmakers in Washington, D.C. Upon graduation, he received the Wayne Wymore Award for Excellence in Systems Engineering and began work at Sandia National Laboratories, who enabled him to attend graduate school. In MIT's AeroAstro S.M. program, he took classes from EECS, Sloan, Technology and Policy, and Harvard Computer Science. His thesis represents somewhere between 16 hours[1] of work and 1,800 years[2] of work, depending on the assumptions made. Michael's fondest memories of MIT will include (remotely) taking part in a test session aboard the ISS, as well as watching a mobile robot use his code to successfully navigate a CSAIL obstacle test bed. One could *almost* say that the robot was thinking on its own.

An old proverb (Proverbs 16:9, to be specific) says that a man's heart plans his way, but his steps are directed by someone else. As a result, another proverb (20:24) asks, "Who can understand his own way?" While Michael has come to understand many things, he is delightedly mystified by the way in which life develops, and he thoroughly enjoys the people alongside whom his steps are found. While these things all could be rather accidental, he quite firmly believes in an underlying purpose, even if he cannot see it.

---

[1] Assuming a typing speed of one word per second.

[2] Assuming "work" includes the average 15 years of experience of the 120 cited authors.

Table of Contents

List of Figures

List of Tables

1 Introduction

"Because not everything happens as one expects or with the smoothness of a parade."

-Leo Tolstoy, *War and Peace*

1.1 Motivation

The motivation for the present work stems from the confluence of several factors. The first factor is the state of the practice of systems engineering for large-scale systems, especially defense systems that are ever increasing in pace and scale, and resultantly cost. Another factor is the current downward trend of federal funding for such systems, resulting from the termination of military operations in the Middle East and the rise of other higher-priority considerations in the national budget. Finally, the third factor is recent improvements in state of the art systems engineering methodology as captured in much of the literature from relatively recent years. The present research attempts to leverage these recent advances in systems engineering to address the growing challenges resulting from the first two factors.

1.1.1 Systems Engineering

According to the NASA Systems Engineering handbook, systems engineering is "a methodical, disciplined approach for the design, realization, technical management, operations, and retirement of a system" (Kapurch, 2010). Many systems may not need this type of methodical systems engineering effort to succeed, if they require low resource investment and their range of functionality, inputs, and outputs is very limited. Larger systems, however, especially those that involve more open and non-obvious interactions among components that are both technical and social, require much more effort in integration of the various subsystems in order to best serve the broader purpose of the system. The behavior of the overall system can be modified not only through modification of its constituent subsystems, but also through adding and removing entire subsystems, as well as adding, removing, and modifying relationships between the subsystems. Because the behavior of the system as a whole differs from that of the parts and must be treated accordingly, the methods of systems engineering prove to be valuable in many different disciplines where these properties emerge (Hall, 1962).

One of the early areas where such properties were identified was in the development of the missiles and missile-defense systems of the 1940s, which is when the use of

systems engineering by the U.S. Department of Defense (DoD) began (Goode, Machol, & Teichmann, 1957). Since that time, defense systems of all kinds have often exhibited the large-scale, open properties that require a high degree of systems engineering for successful mission completion, whether in land, sea, air, space, or virtual environments. These types of defense systems all require much up-front work on problem definition, exploratory planning, integration of subsystems, and plans for system testing and retirement. This extra effort is required due to several factors, including the many different domains involved (e.g., electrical, mechanical, software, social, political), potentially hazardous and varied operating environments, and significant investment of resources required for acquisition and operation (sometimes for even a single copy of a system).

The resource investments required for defense systems have been growing due to the advanced technology and the enlarged scale of operations, resulting in ever-increasing costs for defense systems, regardless of their domain. One of the examples of this increase is the cost of fighter aircraft for the US Air Force (USAF). The historical costs of several fighter aircraft are shown below in Figure 1-1, which adjusts the costs to near-present-day (2011) dollars.

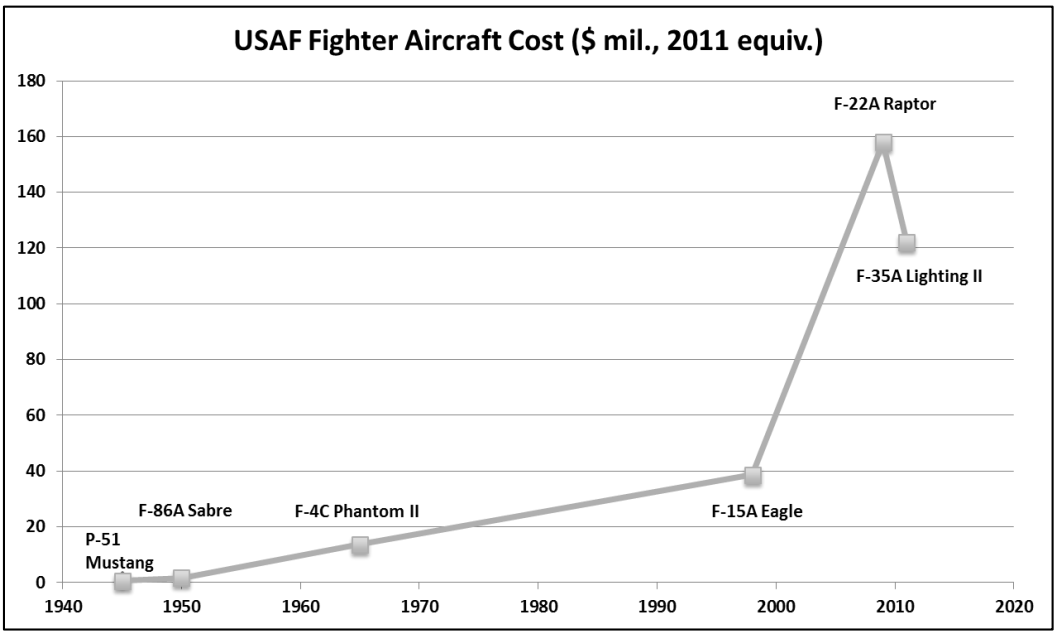


Figure 1-1. Historical costs of fighter aircraft for the USAF, in 2011 dollars.

While the costs of fighter aircraft have historically experienced an annual increase of roughly 10% per year, the US Navy's (USN's) increase in cost for naval ships such as "amphibious ships, surface combatants, attack submarines, and nuclear aircraft carriers have ranged from 7 to 11 percent in recent decades" (Arena, Blickstein, Younossi, & Grammich, 2006). It is certainly the case that today's air and naval

defense systems have capabilities far beyond their historical counterparts, but the budget-related concerns about costs of acquisition and operation are beginning to take widespread precedence over the concerns for higher performance.

1.1.2 Present Need for Affordability in Complex Defense Systems

As demonstrated by the historical data on air, naval, and other defense systems, the complexity and cost of acquisitions and operations are ever increasing; as a result, there is an emergent need for the US Department of Defense (DoD) to ensure the overall success of a system within established resource and schedule constraints. This need is exacerbated by the present conditions of the US winding down military operations in the Middle East, after which budgets are projected to be drastically reduced for multiple years and relatively flat afterward. A historical chart of the post-war periods in the US is shown below in Figure 1-2.

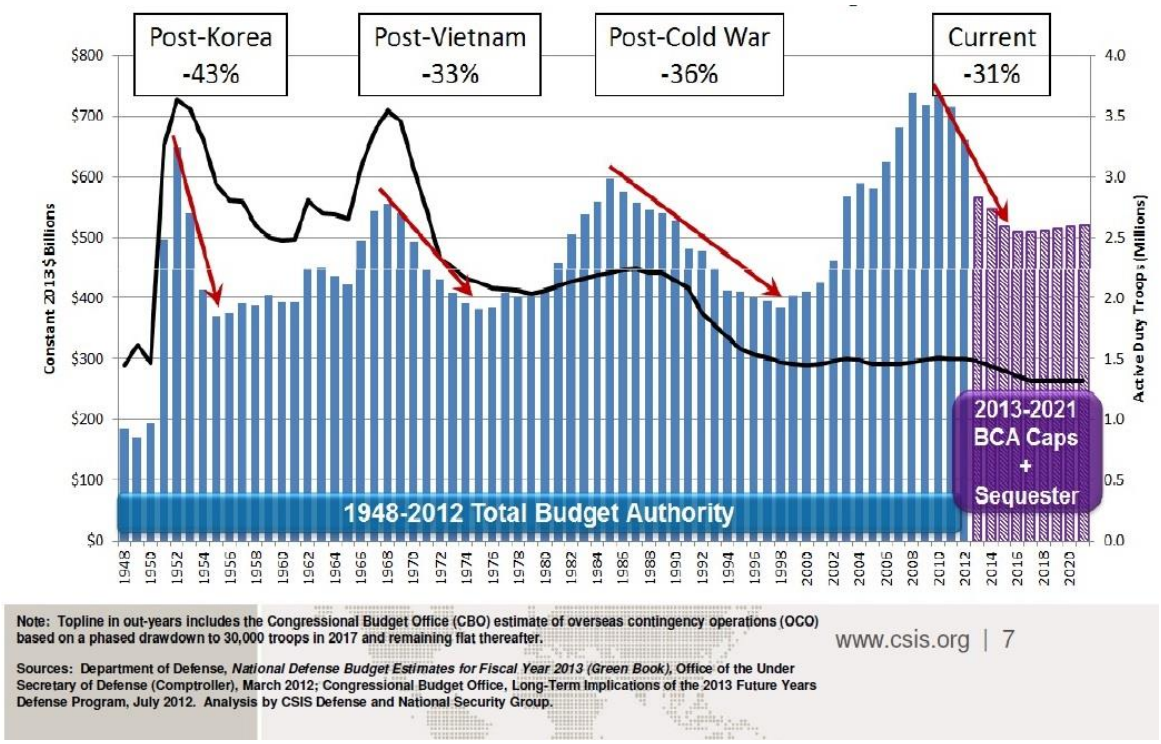


Figure 1-2. Historical drawdowns of military spending in post-war periods. ("Defense Drawdowns Compared," 2012)

With recent budget cuts and the prospect of relatively flat defense budgets in coming years (Schwenn et al., 2011), the DoD is seeking ways to yield better returns on its weapon system investments as well as methods to deliver defense capabilities for less than it has in the past. It is well understood that mitigating uncertainty in estimating cost and schedule parameters that plague the early phases of program formulation would help to identify the true costs of a weapons system or program

from the beginning, and resultantly reduce overruns. This idea is also in consonance with the advent of capability-based planning, which aims to counter external threats with the best warfighter capabilities deliverable under constrained economic conditions and uncertainty (Patterson, 2012).

Buying strategies are continuously evolving to place more emphasis on cost in the decision process. With the launch of the Better Buying Power (BBP) initiatives and Weapon Systems Acquisition Reform Act (WSARA), affordability has been mandated as a requirement at all milestone decision points of program development (Carter, 2010a, 2010b). Designing for affordability is thus imperative to early phase decision-making in the development of weapons systems and programs. Various interpretations of and approaches to designing for affordability are reviewed in the next chapter, and the definition of affordability for the purpose of this thesis is identified.

Efforts to improve cost and schedule estimation are ongoing, but there has been relatively little progress in addressing uncertainties related to costs stemming from alternative futures the system may face. The research described in this thesis is motivated by this latter aspect of uncertainty in the increasingly urgent need of designing for affordability.

1.1.3 Scope

Since the Carter memorandums of 2010, many definitions of affordability have been proposed along with methods intended for use in various phases of the system lifecycle. It is certainly the case that the affordability of any complex system is determined over its entire lifecycle, and its affordability directly depends on many uncontrollable factors as well as the many decisions made during every phase, including design, development, operations, and retirement. This thesis deals exclusively with the early lifecycle phase of conceptual development, what (Hall, 1962) deems the "exploratory planning" phase, in which the problem is defined, objectives are selected, and analysis on potential systems can be performed. In this conceptual development stage, the level of system knowledge (often including knowledge of the operating environment) is very low, but committed costs are low as well, and management leverage over the system is still high. A graphical depiction of these factors taken from (Blanchard & Fabrycky, 2006) is shown below in Figure 1-3.

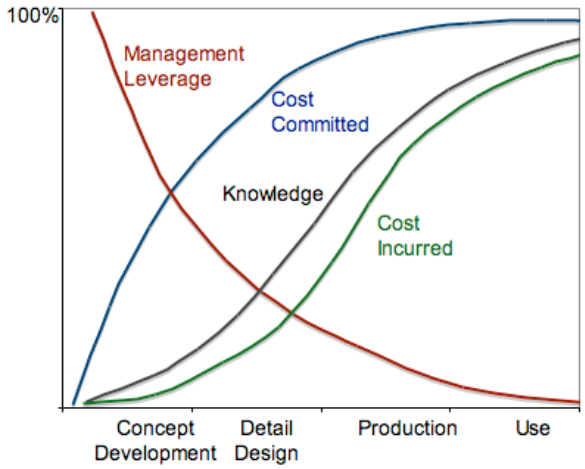


Figure 1-3. Qualitative description of system phases and associated factors (Blanchard & Fabrycky 2006).

By designing for affordability from the earliest system phases, architects can set a system up for success in the goal of more affordable solutions. For the purposes of the present work, this goal includes the ability of a system to provide value over its lifecycle, regardless of changing needs and changing operational environments.

1.1.4 Recent Advances in Early-Phase Systems Engineering

Since the early phases of systems design provide the most leverage over the resulting cost and performance of a system, it is only logical that improving the methods used in these stages will result in more favorable system outcomes. The Systems Engineering Advancement Research Initiative (SEAri) at the Massachusetts Institute of Technology (MIT) has been tackling this challenge for over a decade, providing definitions, methods, and metrics to enable more expedited system knowledge, retained leverage of management over system decisions, and potentially lower costs throughout the lifecycle. One of the core concepts in SEAri research is Multi-Attribute Tradespace Exploration (MATE), which results in consideration of overall benefit versus cost for each system concept alternative in a tradespace (Ross & Hastings, 2004; Ross, 2003). Visually representing the overall aggregate benefit significantly reduces the cognitive burden associated with decisions over attributes in many dimensions. Perhaps more importantly, though, it provides a foundation for *exploring* the tradespace with respect to the assumptions present in the modeling of systems' performance and costs, as well as assumptions present in stakeholder descriptions of value. Details of the contributors to benefit are still tracked and can be interrogated on demand. By avoiding a single metric to automate the selection of an "optimal" system and by not hiding the tradeoffs of concept alternatives, MATE allows analysts and stakeholders to quickly become acquainted with the traits of a

particular tradespace of enumerated system concepts – or even quickly locate features of a model that do not correspond to features of real systems. Preferences on benefits can be modified and real-time effects displayed on the tradespace, with no additional simulation or model execution required (Ricci, Schaffner, Ross, Rhodes, & Fitzgerald, 2014). All of these features help those who are conducting a conceptual design study to better understand the relationships between perceptions of, and values present in, complex problems and potentially complex system concepts.

Another key benefit of using MATE is that exogenous uncertainties in future system environments can be incorporated relatively easily into the design study, through the implementation of Epoch-Era Analysis (Ross & Rhodes, 2008). Epoch-Era Analysis (EEA) discretizes a system's lifecycle according to value-centric considerations rather than according to traditional system milestones. An *epoch* is then considered to be any period of fixed operating context and fixed stakeholder needs. An *era* is an ordered sequence of finite-duration epochs that make up an entire system lifecycle. These constructs allow clearer understanding of the relevant perturbations in the uncertain future that a system may encounter throughout its lifecycle, whether those perturbations impact stakeholder needs (and therefore the perceived value delivery) or the performance of the system (and therefore the value delivery).

Any discussion of a system's uncertain future, of course, inevitably leads to the questions surrounding a system's ability to *change* in some way. Previous research has examined the topic of changeability in the face of uncertain futures (Fitzgerald, 2012). The Valuation Approach for Strategic Changeability (VASC) method was developed specifically for valuing the future changeability of a system while still in the early design (i.e., conceptual development) phase. It enables this valuation through the use of change-related metrics and the creation of *change strategies*, which allow automated simulation of changeable designs in many different epochs (periods of fixed context and needs). Two key areas for further research identified in the VASC study include considerations of computational complexity as well as long-term considerations of the use of change options.

The VASC method, along with MATE and Epoch-Era Analysis, all provide activities, products, and analysis for system architects who desire to make more informed design choices early in the lifecycle. All of these methods and constructs for enhanced early-stage conceptual design can greatly aid in tackling the challenge of designing changeably affordable systems, making extensions of these a natural area of research and forming the third factor in the motivation for the present research.

1.2 Research Questions and Methodology

Motivated by the context and factors described in the preceding section, the questions addressed by this research are as follows:

1) Can the affordability of complex systems be better enabled through an early-phase design method that incorporates Multi-Attribute Tradespace Exploration and Epoch-Era Analysis?
2) Can the affordability of changeable systems be better enabled by leveraging existing path planning approaches to efficiently evaluate long-term change strategies for systems in many possible futures?
3) How can we better capture the considerations related to long-term, path-dependent properties of changeable systems in dynamic operating environments?

Several steps were involved throughout the course of this research over the two years in which it was conducted. The first step was a literature review of recent advances toward the design of affordable defense systems[1], of the background and purpose of exploratory design models[2], and of path-planning approaches used in artificially intelligent systems. Next, preliminary research questions were identified representing areas for further investigation, and development began on the RSC-based Method for Affordable Concept Selection, which was intended to directly address the first research question. The next step included the development of related affordability metrics, and application of the method and metrics to a case study involving a handful of point designs of naval ship concepts. The results of the application of the first eight processes of the method were validated by peer-review, resulting in acceptance to conference proceedings. Finally, the ninth process is introduced in the present work with application to a similar system as the first 8 processes, and is intended to directly address the second and third research questions.

1.3 Thesis Overview

The overview of the thesis is now described. Chapter 2 is a literature review, beginning with various definitions of and approaches to affordability. After a brief introduction to some of the considerations with regard to modeling and computational complexity of the activities in the RMACS method, the chapter concludes with a review of the case applications encountered in the thesis. Chapter

---

[1] The author was privileged to work alongside Marcus Wu for most of this portion of the research.

[2] The author was privileged to work alongside Nicola Ricci for most of this portion.

3 continues with an in-depth discussion of the basic assumptions inherent in any early-stage predictive modeling effort, including the purposes and limitations of modeling the performance and value parameters of complex future systems. It then identifies the types of models used in the present method, and describes the nine processes in the RSC-based Method for Affordable Concept Selection (RMACS). Chapter 4 performs a case demonstration of the first eight processes of RMACS on the Next-Generation Combat Ship (NGCS) study of a small set of six point designs. Chapter 5 introduces the ninth and final process of RMACS, which is Multi-Era Analysis (MERA). The informational components, both required and optional, are outlined, and the nature of and activities in the process are described in depth. In Chapter 6, MERA is then demonstrated on an NGCS-like case application, adding in discrete change options that form a tradespace network. Chapter 7 discusses several issues arising in the application of the MERA process and the RMACS method, and also identifies many key areas for future work. Finally, Chapter 8 provides concluding thoughts with respect to the original research questions.

2 Literature Review

"Don't only practice your art, but force your way into its secrets."

-Ludwig van Beethoven

This chapter covers a portion of the literature essential to the terminology and to the various constructs and methods used in modern early-stage design of complex systems. The chapter begins with a review of the various interpretations of the concept of "affordability" with respect to defense systems. Issues arising in the use of early-phase system models are reviewed, and the models used in the present work are introduced, including Multi-Attribute Tradespace Exploration (MATE), Epoch-Era Analysis (EEA), and the Valuation Approach to Strategic Changeability (VASC). Enabling techniques used or recommended in this thesis are briefly discussed, including algorithms for multi-dimensional spaces, computational complexity analysis, and tradespace approximation techniques. Finally, the case studies encountered and used in application are covered.

2.1 Affordability Concepts and Definition

Because the word *affordability* has been frequently used without definition in various and sundry contexts, it is not surprising that several distinct definitions of affordability in defense systems have recently arisen, being proposed by various organizations. The 2010 Carter memorandum defines affordability as "conducting a program at a cost constrained by the maximum resources the Department can allocate for that capability" (Carter, 2010a, 2010b). INCOSE defines affordability as "the balance of system performance, cost and schedule constraints over the system life while satisfying mission needs in concert with strategic investment and organizational needs" (*Affordability Analyses: How do we do it?*, 2012). NDIA defines affordability as "the practice of ensuring program success through the balancing of system performance (KPPs), total ownership cost, and schedule constraints while satisfying mission needs in concert with long-range investment, and force structure plans of the DOD" (*NDIA Status Report*, 2011). The Defense Acquisition Guidebook defines affordability as "the degree to which the life-cycle cost of an acquisition program is in consonance with the long-range modernization, force structure and manpower plans of the individual DoD Components, as well as for the Department as a whole" (Department of Defense, 2011).

In addition to the definitions above, (Bobinis, Haimowitz, Tuttle, & Garrison, 2012) provide the result of development without consideration for affordability: a system

that has been designed as a point solution in isolation, to meet a specific need at a specific time, possibly requiring the procurement of an entirely new system when customer needs evolve.

With regard to helping define affordable decisions, (Tuttle & Bobinis, 2012) describe the Affordability Triangle, with required capabilities (determined by stakeholder needs) forming the base of the triangle, and the affordability decision criteria comprising cost, schedule, and performance. It is clear that examining each of these latter multi-criteria considerations in multi-year, billion-dollar weapons systems and programs adds several layers of complexity to a standard trade study. In addition, Tuttle and Bobinis note that an affordability trade study must "extend the time horizon" of the traditional analyses. In order to extend the time horizon of analysis, any study must address the contextual and capability developments over time, along with their potential impacts to the performance, cost, and schedule of the system (whether the impacts are objective or subjective). The ultimate goal is to design affordable systems that meet the needs of warfighters – *and remain affordable* – regardless of future circumstances. This goal can be challenging due to in part to future events that are not trivial developments, but rather *wild cards* (Mendonça, e Cunha, Kaivo-Oja, & Ruff, 2004), leading to "deep uncertainty" regarding a system's future (Davis & Kahan, 2007). To achieve the goal of affordability in the face of such uncertainties, efforts toward affordability must begin in the earliest phases of system planning, when resource commitments and solution-constraining decisions are not yet present. Performing such analysis at this stage allows design engineers to "see, evaluate, accept and reject a large number of courses of action…without actually committing resources" (Hall, 1989), potentially elevating system-specific knowledge to better inform the high-impact decisions made at the earliest stages of the system lifecycle. With specific attention to the idea of considering affordable alternatives that can change in response to future developments, Neches and Madni list several technological and modeling challenges, including characterizing changing operational environments in addition to performing tradespace analysis (Neches & Madni, 2013). To include the idea of adaptability, and for the purposes of early-lifecycle conceptual development of such systems, the ***affordability*** of a system is defined for this thesis as "**the property of becoming or remaining feasible relative to resource needs and resource constraints over time**" (Wu, Ross, & Rhodes, 2014).

2.2 Modeling Considerations

Tradespace analysis, characterization of changing operational environments, and other early-phase design activities all require methods to represent the relevant

portions of reality. This section reviews the established thought on representing these portions of reality (i.e., *modeling*) and resultantly predicting the outcome of various design choices. First, the types and purposes of models are reviewed, along with more recent developments in modeling the traditionally less-emphasized aspects of complexity in system design. Next, the limitations of modeling unobservable systems are identified. Finally, ramifications of these limitations and recommended approaches are then discussed within the contexts of system design, scenario planning, and solving general complex problems.

2.2.1 Use of Models in System Design

(Ricci et al., 2014) describe the types and purposes of models used in early system design. Model purpose can include prediction of the performance or of the value delivery of a system, while model type is either mental or constructed, depending on where the model resides. (Hall, 1989) describes the choices involved in modeling the value that a system is expected to deliver, and emphasizes the importance of this choice to the overall success of the system design process. (Fischhoff et al., 1991) describes the spectrum of alternate interpretations that can be assumed in attempting to model the value systems present in decision problems.

(Rhodes & Ross, 2010) describe five types of system complexity in present-day engineering systems that potentially hamper the ability to understand and model a system. Several methods have been proposed to help manage the contextual, temporal, and perceptual complexities in the early phases of system design. One is the use of a constructed value model for system performance attributes based on Multi-Attribute Utility (Keeney & Raiffa, 1976), proposed by (Ross, 2003) in the creation of Multi-Attribute Tradespace Exploration (MATE). Another is Epoch-Era Analysis (EEA), which involves the creation of *epochs* (periods of fixed context and needs for a system) and *eras* (an ordered sequence of finite-duration epochs), proposed by (Ross & Rhodes, 2008). A combined application of MATE and EEA was demonstrated with the Responsive Systems Comparison (RSC) method applied to the Satellite Radar System by (Ross, McManus, & Long, 2008; Ross, McManus, Rhodes, Hastings, & Long, 2009). (Richards, 2009), (Fulcoly, 2012), and (Fitzgerald, 2012) all build upon these models with methods and metrics designed to capture the contextual, temporal and behavioral complexities of systems over their lifetimes. (Diller, 2002) captures stakeholder preferences on expense attributes using the Multi-Attribute Expense (MAE) function, analogous to the Multi-Attribute Utility function. (Nickel, 2010) demonstrates the MAE applied to a MATE case study for the Chicago Express Rail.

2.2.2 Accuracy of Models for Complex Future Systems

In *Systems Engineering and Analysis*, (Blanchard & Fabrycky, 2006) discuss the accuracy of models and simulations in early-phase complex systems. They emphasize that the models of these systems represent information that cannot be validated, since the system does not yet exist. As a result, the concept of accuracy cannot be invoked for these types of models and simulations, since there exists no referent by which to compare different models. In earlier work on modeling and simulation, (Law & Kelton, 1991) posit the same point, stating that it is impossible to completely validate models of future complex systems, as opposed to models of simple, presently observable processes (e.g., the frequency of arrivals to an existing storefront).

2.2.3 Use of Exploratory Models in System Design

The generation and use of data (that cannot be validated) from predictive models of complex systems cause (Bankes, 1993) to distinguish *exploratory* models (of complex and future systems) from *consolidative* models (of data that can be validated by empirical observation). While consolidative models are intended to use external data to create and store (potentially new) relationships that can explain the data, exploratory models are intended to generate artificial data by creating and storing smaller (relative to the overall system), well-established relationships and allowing them to interact. These exploratory models and artificial data can inform modelers and decision-makers of the ramifications of various sets of assumptions, as well as provide consistent communication and expectations regarding the various aspects involved in the approach to a complex problem. Similarly, (Sarewitz & Pielke Jr, 1999) delineate between model prediction for scientific purposes (i.e., explaining empirical observations) and model prediction used to predict behavior of complex systems of the future. In work on computational scenario planning, (Abbass et al., 2008) point out that the goal of modeling complex systems is not the optimization of some overarching objective function, but rather a means to better understanding a complex problem and better understanding of the complex approaches to that problem. (Schön, 1993) describes how immersive experience in questioning assumptions and viewing a problem different ways can lead to sudden insights into problem solving, deeming the useful alternate views “generative metaphors”.

2.3 Computational Considerations

The models used in modern system design often reach limits in the amount of information which they can represent and on which they can operate. This section reviews the computational complexities of various models used in the

representations of systems and their evaluations in alternative futures. The first topic reviewed is that of search algorithms for discrete networks, which are used later in the creation of eras of interest as well as era-level strategies. A type of search algorithm for continuous spaces is also reviewed, along with the complexities of several types of combinatorial problems related to computational scenario planning. Finally, methods are discussed to alleviate the computational challenges associated with fully-enumerated tradespaces in fully-enumerated sets of discrete futures.

2.3.1 Search Algorithms for Discrete Networks

In *Artificial Intelligence: A Modern Approach*, (Russell & Norvig, 2009) summarize uninformed (or *blind*) search strategies such as breadth-first search (BFS) and depth-first search (DFS), comparing their worst-case time and spatial computational complexities. "Blind" (or "uninformed") algorithms simply operate on each node and edge in turn, with no knowledge of what might be a "good" direction to take next in the search for the goal node. The difference in order of operations of these two uninformed algorithms is depicted below in Figure 2-1, provided by www.cse.unsw.edu.au.

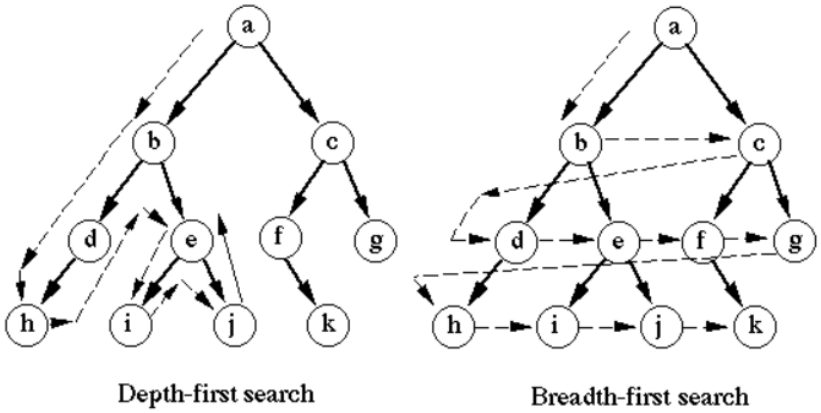


Figure 2-1. Depth-First Search (DFS) vs. Breadth-First Search (BFS). Courtesy (W, n.d.)**.**

The network structure of the diagrams shown in Figure 2-1 is important to several topics in this thesis, including design changes throughout an era, as well as the temporal and spatial computational complexities of constructing eras. The worst-case complexities of each of the two blind search algorithms are the same, shown below in Table 2-1.

Table 2-1. Worst-case complexities for simple search algorithms on a network, where b=branching factor of the network, and d=depth of the "goal" node in the network.

Network Search Method	Worst Time Complexity	Worst Space Complexity
Blind (e.g. BFS or DFS)	b^d	b^d
Informed (e.g. A*)	b^d	b^d

In addition to those of the uninformed algorithms, the complexities of the (informed) best-first search algorithm A* are also shown. A best-first search algorithm has some estimate of distance to the goal (i.e., a *heuristic*) that it uses in evaluating which edge to traverse next in the search. For example, a common heuristic in physical path-planning of a robot is the straight-line distance to the goal region. This type of heuristic is shown below in Figure 2-2, where the value of the straight-line distance is represented by color. The A* algorithm keeps track of each node's *estimated* distance to the goal, and explores first the nodes with the lowest estimates. Before deciding which edge to traverse next in the search, A* will add the heuristic cost of a node (its *estimated* distance to the goal) to the actual cost to reach that node, and will only traverse the edge if that sum is less than the shortest estimated path found so far. More information on the results from different heuristics operating in different environments can be found on theory.stanford.edu.

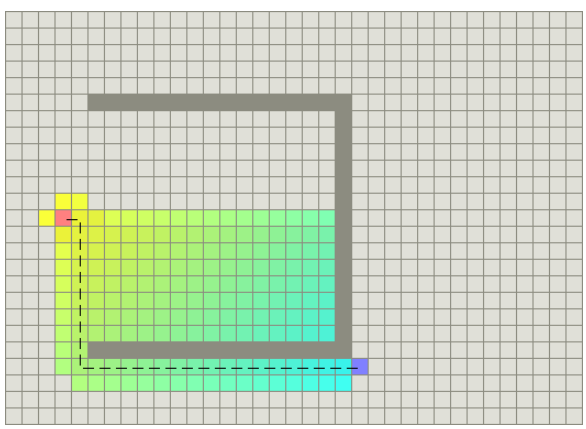


Figure 2-2. The nodes (i.e., colored squares) searched by the A* algorithm operating on a discrete grid, starting in the red square (upper left) with the goal node being the blue square (lower right). The heuristic (i.e., estimate of distance to the goal) for each searched square is represented by a color that represents its straight-line distance to the goal region. Courtesy (Patel, n.d.)**.**

2.3.2 Rapidly Exploring Random Trees for Continuous Spaces

Probabilistic roadmaps (PRMs) were introduced in the mid-1990's as a method for path planning in high dimensional configuration spaces (Kavraki & Svestka, 1996).

Research in the following years developed algorithms to complement PRMs, such as rapidly exploring random trees (RRTs). The basic idea of RRT is to connect each newly sampled node on the map to only its nearest neighbor, creating a tree of nodes, which unlike a graph, does not have to be searched for the optimal path to a goal region. RRT was recently proven to converge to suboptimal solutions, and an updated version known as RRT* has been developed that modifies some of the sub-algorithms of the classic RRT (Karaman, Walter, Perez, Frazzoli, & Teller, 2011). While RRT* quickly produces feasible solutions (as RRT does), it also almost always converges to optimal solutions (unlike RRT).

2.3.3 Spatial Complexity of Combinatorial Problems

The complexity of path-dependent combinatorial decision problems is well established, going back to the origins of the Traveling Salesman Problem (an NP-Hard problem, though variants can be in NP-Complete) with Menger and Whitney (Schrijver, 2005). Other similar problems, such as finding a solution to a Boolean satisfiability problem (SAT problem), are known to be in NP-Complete, while counting the number of solutions proves to be much harder (Biere, 2009; Sabharwal, 2008; van Harmelen, Lifschitz, & Porter, 2008). Since the complexities of these types of combinatorial problems scale exponentially with the number of elements being considered, enumeration of the possibilities quickly exceeds any temporal bounds and storage capabilities of computing resources of any size. Even partial sampling of such spaces requires highly sophisticated methods to ensure uniformity of the samples (Erenrich & Selman, 2003; Lin & Iii, 2012). While some promise has been shown by these highly specialized algorithms developed for sampling and finding solutions in specific constructions of these problems, such problems remain the area of much active research for their wide applicability to task planning and scheduling, adversarial gameplay, and computational scenario planning (Sabharwal, 2008).

2.3.4 Tradespace Approximation Techniques

Tradespace exploration with models of sufficient complexity quickly becomes computationally challenging due to the fact that the number of evaluations required does not scale linearly with the number of design variables modeled. One type of response to this issue is a systems design method based around enumerating only portions of a tradespace. Such methods include: SPIDR (Kichkaylo, Hoag, Lennon, & Roesler, 2012), which chooses designs to evaluate in a tradespace based on optimization of individual performance or cost attributes; techniques that use models of varying levels of fidelity in order to achieve initial computational

tractability (Howell, Uebelhart, Angeli, & Miller, 2005); and methods such as DAKOTA (Adams et al., 2013) that combine these ideas together.

Another type of response to this issue uses a surrogate model (or metamodel) to make computationally cheap predictions about the results of a design point's evaluation in a fully enumerated tradespace. The creation of epochs for an entire tradespace requires a separate evaluation of the entire tradespace for each epoch, and continuous change options can also further exacerbate computational challenges. Tradespace approximation methods can help alleviate these compounded computational considerations, allowing a design study to be completed within reasonable time given reasonable computational power. (Fulcoly, 2012) proposed the Expedited Tradespace Approximation Method (ETAM) to harness the ability of surrogate models, specifically Kriging models, in the exploration of tradespaces with Epoch-Era Analysis.

In a 2012 Sandia National Labs report, (Swiler et al., 2014) compared three different types of surrogate models specifically for *hybrid* evaluation models that include discrete and continuous variables. Large-scale complex systems are usually modeled with both of these kinds of variables, as are those models incorporating Epoch-Era Analysis (which frequently add categorical and ordinal variables to a study). While their report includes individual Kriging models (such as those applied in ETAM), the report also covers treed Gaussian processes (Gramacy & Lee, 2008) and the adaptive smoothing spline method ACOSSO (Storlie, Bondell, Reich, & Zhang, 2011).

It is assumed for the purposes of the present work that methods like those above are viable for generating high-quality approximations of unevaluated design points throughout a space of continuous-range change options in designs of interest. It is also assumed, however, that such methods cannot be used to approximate metrics on designs over eras.

2.4 Case Studies

Several previous system design cases are encountered and used in portions of the present work. They are the Space Tug Satellite, X-Terrestrial Observer Swarm (X-TOS) system, and a hypothetical Next-Generation Combat Ship (NGCS) naval system.

2.4.1 Space Tug

A space tug is a satellite whose primary purpose is the orbital transfer of other satellites, as well as removing space debris and potentially observing hostile targets. Such an orbital transfer vehicle has been previously proposed, and a basic MATE

study was performed by (McManus & Schuman, 2003) to compare many design alternatives in a tradespace. The dataset was later extended to include different epochs for demonstration of Epoch-Era Analysis, including changing preference sets and technology levels, and several studies on change options have used the dataset for case application of the methods proposed (Fitzgerald & Ross, 2012; Richards, 2009; Ross & Hastings, 2006). (Fulcoly, 2012) uses the Space Tug in the application of the Epoch Syncopation Framework (ESF), in which epoch variable durations and transitions are modeled as Markov processes which can then be simulated to study the behavior of change options, choices of evolvability in a system, and strategies for the timing of future changes.

2.4.2 X-Terrestrial Observer Swarm (X-TOS)

The X-Terrestial Observer Swarm is a continuation of previous distributed satellite design studies such as A-TOS and B-TOS (Ross, 2003). The study exists in different versions, with the version of the study used in the present thesis comprising 3,384 unique designs with 8 change options. The same version is used in (Fitzgerald, 2012) for one application of the VASC method.

2.4.3 Next Generation Combat Ship

A final case study used throughout the thesis is the representational Next-Generation Combat Ship (NGCS), which is described as a larger version of the Navy's current Littoral Combat Ship (LCS) that would support air and sea operations over diverse areas of interest for the next 30 years. The application in (Schofield, 2010) of the original RSC method was to a smaller naval application, the Coast Guard's Offshore Patrol Cutter (OPC). The NGCS case combines the design variables, attributes, and epochs from Schofield's OPC study with the evaluated outputs of the MIT Math Model (Smith, Smith, & Marcus, 2008), a standard naval modeling tool regularly used for the evaluation of potential designs for Naval frigates (slightly larger than the LCS).

The proposed NGCS requirements, therefore, reflects some similarity with both the OPC and LCS. For example, the OPC is designed to operate in a variety of mission areas, including ports, near shore, and open sea, with a range in excess of 8,500 nautical miles and endurance minimum of 45 days (Schofield, 2010). The LCS is designed to have a range in excess of 3,500 nautical miles and an endurance of 21 days. The NGCS that is the focus of this research, meanwhile, is required to operate in mission areas at least as varied as the OPC, have a minimum endurance of 30 days, and have a range in excess of 4,000 nautical miles. The operating context of

the NGCS is also largely unchanged from that of the OPC, with many of the NGCS's contextual variables borrowed from the OPC study.

3 Methodology for the Early-Phase Design Method

> "Although this may seem a paradox, all exact science is dominated by the idea of approximation. When a man tells you that he knows the exact truth about anything, you are safe in inferring that he is an inexact man."
>
> -Bertrand Russell

This chapter elucidates the methodology of the present work. It begins with an in-depth discussion of the use of exploratory models in the design process. This discussion helps contextualize the types of models used and the associated results expected throughout the rest of the thesis. The design method for selection of affordable system concepts is then proposed, with brief explanation of the activities involved and results expected. The next chapter then applies the method to a demonstration case.

3.1 Exploratory Models in Design

Models are essential tools in system design and are used by analysts and engineers throughout the design process. A key function of models is to generate data when empirical sources are not available, which are then used by stakeholders and decision makers to discern among potential alternatives. The key issues that arise in this process are then the challenge of building trust and the problem of building truthfulness in the models used to represent the performance of engineering systems that cannot be externally validated. There is another significant challenge, however: that of building trust and the problem of building truthfulness in the models that represent the *value* that stakeholders attribute to the predicted performance of engineering systems.

These issues are now discussed in order to lay a foundational perspective for the design method and activities proposed in later chapters of the thesis. The design-performance-value loop (hereafter referred to as the "design loop") is discussed, and a framework is delineated for the comparison of important model types involved at various stages in the design loop (performance v. value, mental v. constructed). The problem of building trust and truthfulness in models can then be defined within this proposed framework. The models upon which the present research builds are then discussed, with emphasis on the exploratory nature of the method and the models.

3.1.1 Complexity and perception

In the conceptual design phase of modern engineering systems, analysts are confronted with exploring and representing several layers of complex information

about the design alternatives they are considering. (Rhodes & Ross, 2010) outline five aspects of complexity in engineering systems: structural (related to systems' forms), behavioral (systems' operations), contextual (environment in which the systems operate), temporal (systems changing over time), and perceptual (stakeholder preferences on systems' performance outcomes). Given these complexities, not only is it difficult to model the performance of a system that does not yet exist, but also it is difficult to represent stakeholders' views on what the system should do (i.e., their needs). These difficulties bring about the problem of *trust* (do I "believe"?) in the models that are used to evaluate both the performance and the preferences on that performance (i.e., value models). Another important problem is related to how *truthful* (is it "correct"?) the models actually are – independent of trust (since it is possible to trust a very poor model).

3.1.2 The Design Loop

Conceptual design of complex systems involves a design-performance-value loop (see Figure 3-1). First, a design space of alternatives is generated based on a preliminary understanding of stakeholder needs. Each design alternative is mapped onto a performance space, spanned by attributes of interest quantified by a constructed performance model. The performance space is then mapped onto a value space via a value model (e.g., functional requirements, utility functions), through which the stakeholder evaluates the attractiveness of the alternatives. At this point, with an understanding of how each design alternative scores according to stakeholders' values, it is possible to either make a decision on what design (or set of designs) to focus on, or to go back and change the initial design space and repeat the loop.

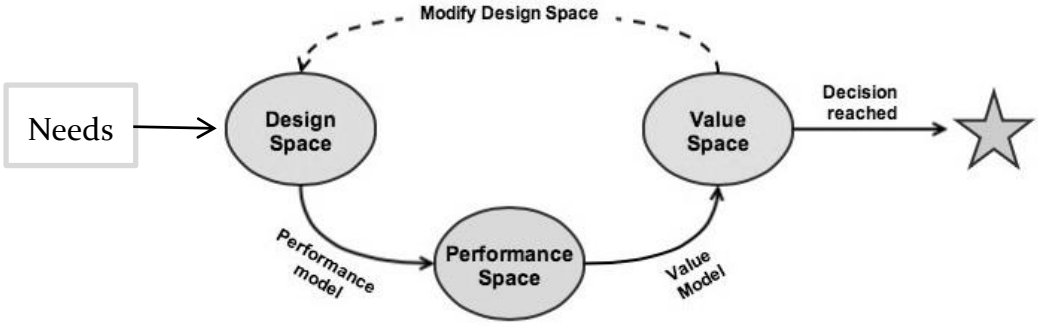


Figure 3-1. Illustration of the design-performance-value loop.

3.1.2.1 Artificial Data in the Design Loop

Engineers designing complex systems of the future are often forced to use complicated models and simulations in order "to explore…system performance without actually producing and testing each candidate system" (Blanchard & Fabrycky, 2006). These models and simulations have embedded in them causal and functional relationships, as well as empirical data from the past, which enable the synthesis of new data describing how the system is going to perform. In these cases, the data generated is *artificial* (synthetic) – i.e., it is not obtained by direct measurement of system properties, since an identical (or even similar) system may not yet exist. As a result, the artificial data (as well as the model) "cannot be classified as accurate or inaccurate in any absolute sense" (Blanchard & Fabrycky, 2006). Thus, artificial data stands in stark contrast to empirical data (e.g., temperature readings, historical stock prices), which is directly measured and thereby holds a potentially higher degree of validity (i.e., reliability with respect to the relevant components). Law and Kelton express the difficulties of model validation for complex future systems:

> The ease or difficulty of the validation process depends on the complexity of the system being modelled and on *whether a version of the system currently exists*… For example, a model of a neighbourhood bank would be relatively easy to validate since it could be closely observed. However, a model of the effectiveness of a naval weapons system in the year 2025 would be impossible to validate completely…. (Law and Kelton 1991)

As Law and Kelton point out, however, even after a simulation is (partially) validated, it "can only be an *approximation* to the actual system". There must remain some departures from reality in the artificial data – as (Simon, 1996) notes: "artificiality connotes perceptual similarity but essential difference." Regardless of the amount of "essential difference", though, it is the "perceived similarity" to reality that leads to model *credibility* (i.e., trustworthiness), which is when a "manager and other key project personnel accept [it] as 'correct'" (Law & Kelton, 1991).

The issues related to the artificial data from performance models (of "unobservable" systems) also apply to the value models in the design loop (see Figure 3-1). For the purposes of this thesis, it is assumed that a stakeholder's values cannot be observed by direct measurement – meaning any data provided by a mental or constructed model of values (e.g., utility function, set of requirements) is artificial data. Therefore, when dealing with artificial data from the performance models as well as

the value models, stakeholders and design engineers are inherently concerned with the following questions: Are the models truthful? Do I trust the models?

3.1.2.2 Model Purpose and Type

For the remainder of this section – and in the context of complex engineering systems design – two meta-dimensions of models are explored: model purpose and model type. Model purpose can be prediction of the *performance* of the system (i.e., performance model) or prediction of the *value* stakeholders assign to that performance (i.e., value model), as reflected in Figure 3-1. Model type differentiates between where the abstraction of reality (or of values) resides: a *mental* model "represents entities and persons, events and processes, and the operations of complex systems" (Johnson-Laird, 2005) and resides in the mind; a *constructed* model is a formalization of (one or more) mental model(s), and it can reside, for example, in a computer simulation or on a piece of paper in the form of a diagram. Mental models about reality are created automatically through perception and cognition (Johnson-Laird, 2005), whereas constructed models are intentionally created for widespread applications, such as system dynamics models of climate change, a discrete-event simulation of a surveillance system, or a system of differential equations describing the diffusion of heat in a homogeneous body. These meta-dimensions of model type as well as model purpose are illustrated in Table 3-1.

Table 3-1. Matrix illustrating different kinds of models.

		Purpose	
		Performance	Value
Type	Mental	Mental performance model	Mental value model
	Constructed	Constructed performance model	Constructed value model

3.1.2.3 The Performance Models

The performance model for a system is an attempt to represent some relevant portion of reality and predict the outcomes of interest that will occur in that portion of reality. As noted in the matrix above, two distinct types of performance models are present throughout the design process: *mental* and *constructed* models. The first type, the mental model, begins with observation of existing physical systems and their properties. The mental model of even the least complicated system is severely limited, however, by the inherent bounded rationality of human cognition (Simon, 1996). These general intuitions regarding a system's properties lead to the construction of models of all kinds (e.g., abstract, physical, computer simulations) to predict, capture, and study the specific interactions and behavior of complex systems. The constructed performance model for a system thus becomes a "store" for all related models of reality (e.g., stochastic models, laws of physics, cost models), attempting to formalize human thought and extend the bounds of human rationality.

Unlike empirical data from existing systems, the artificial data produced by performance models of unobservable complex systems cannot be relied on as reliable predictions of reality. This notion leads to the conclusions of (Bankes, 1993), who proposed distinguishing "consolidative" models (for observable systems) from "exploratory" models (used for predictive models of complex, future systems). Through a convincing thought experiment, Bankes demonstrates the inability of any predictive model to provide the information necessary for "effective" decision making for complex systems. He then proposes that such predictive models' role in decision making and analysis should be primarily *exploratory*. In other words, these models do extend the bounds of human rationality, but the extension benefits modelers and decision makers by helping them better understand the decisions and assumptions made in the modeling process, *not in directly perceiving the future behavior of the complex system itself*. (Sarewitz & Pielke Jr, 1999) concur in their treatment of the differences between model prediction for scientific purposes and model prediction used for complex systems of the future. In more recent work on computational scenario-based planning, (Abbass et al., 2008) note that "planning problems are not optimization in uncertain or dynamic environments problems." Rather than being a performance optimization problem, the point of exploring a design space in planning for an uncertain future is to better prepare – not necessarily to pick a purportedly optimal design at a time when relatively little information is available regarding system performance and future developments. It is when researchers are "immersed in experience of the phenomena" that new perspectives and understandings of a problem can be generated (Schön, 1993).

3.1.2.4 The Value Models

The value model discussed above is an attempt to capture stakeholder preferences on performance (i.e., their values). (Fischhoff et al., 1991) discusses a spectrum of philosophies with regard to the nature (and elicitation) of values. On the one end of the spectrum is the philosophy of articulated values, which assumes that values are self-evident in people's choices. This philosophy is intrinsic, for example, in some valuation methods used by economists (e.g., empirical estimates of a demand curve, or of the value of life (Viscusi, Harrington, & Vernon, 2005)). On the other end of the spectrum is the philosophy of basic values, which derives value models from some core set of values through an inferential process (e.g., interviews). The present work assumes a basic values philosophy, as it best approximates a "requirements-based" approach common in systems engineering. Additionally, the risks of such an assumption are potentially less relevant or impactful to a design outcome (of multi-stakeholder, large-scale complex systems) than the assumption of articulated values: (Fischhoff et al., 1991) identifies the risks associated with assumption of articulated values as "incomplete" or "meaningless" values and imposing a "single perspective" on the problem, while the risks of assuming basic values are to "shake confidence" of, "discourage", and "distract" decision-makers. The former risks seem to be more consequential than the latter. Other intermediate positions exist on the spectrum, such as that of partial perspectives.

With this understanding and assumption, a value model can be used to predict the preferential order of design alternatives. Such a model assumes that some approximation can be created for a stakeholder's basic values. The *mental* value model can be described as the stakeholder's own *grasp* of what is important to them and how important it is (i.e., the decision-making criteria and their respective weights they carry for the decision). Similar to the case of performance models, a *constructed* value model can store one or more mental models of values.

Within the context of complex system design, there are usually too many alternatives for a stakeholder to evaluate. In addition, the many dimensions of value (and their interactions) are often beyond the capability of a stakeholder's mental value model, due to *bounded valuation* – i.e., bounded rationality applied to the description/prediction of preferences (Ricci et al., 2014). For the problem of deciding among too many alternatives, a constructed value model can serve as a "stand-in" for the stakeholder's mental value models. Similarly, for the problem of bounded valuation, a constructed value model can serve as a reliable predictor for a potentially overwhelmed or confused stakeholder. Examples of constructed value models are customer value models in economics (i.e., a representation of the worth

– in monetary terms – of what a company does for its customers), or Multi-Attribute Utility (MAU) functions from classical decision theory (Keeney & Raiffa, 1976; von Neumann & Morgenstern, n.d.). The constructed value model's reliability is critical to success in system design; for as Hall points out, "to design the wrong value system is to design the wrong system" (Hall, 1989).

3.1.3 Models Used in Present Design Method

With the understanding of how exploratory models can be used in predicting the future behavior of complex systems, the models used in the present work are now presented. Each intends to capture certain aspects of system design that are of concern to stakeholders and analysts considering the motivation and scope of the present thesis. None are intended to provide the most amount of information possible about complex systems of the future, since such a goal is surely unattainable (Bankes, 1993). Rather, these models and constructs are intended to simplify relevant aspects of system complexity to a manageable level, allowing easier formation of mental models of the properties and behaviors of complex systems.

3.1.3.1 Multi-Attribute Tradespace Exploration (MATE) and Epoch-Era Analysis (EEA)

MATE is a conceptual design method that considers large numbers of designs through combinations of nonlinear functions of their performance attributes, and compares their costs and utilities (Ross, 2003). Enumeration and evaluation of many alternative designs allow for a more complete exploration of a larger design tradespace. This exploration process helps avoid the biases associated with premature fixation on single-point designs, while also encouraging the questioning of assumptions in the model-building and exploration process. One of the core concepts in MATE is the Multi-Attribute Utility (MAU) function as the constructed value model of the tradespace (Keeney & Raiffa, 1976); the output of this function on each alternative is then plotted with the respective cost of each alternative, forming the tradespace.

The analysis of affordability tradespaces can become a multifaceted process when the impacts of uncertainties (including risks) inherent in alternative futures are incorporated into conducting tradeoffs during acquisition. Epoch-Era Analysis (EEA) (Ross & Rhodes, 2008) was developed to effectively evaluate the impacts of these dynamic variations, and can be applied to the early lifecycle design of affordable systems. EEA has been developed to consider and clarify the effects of changing contexts and needs over time on the perceived value of a system in a structured manner. Instead of discretizing the system lifecycle according to traditional system milestones, EEA discretizes the lifecycle according to impactful

changes in the operating environment, stakeholders, or the system itself, through the constructs of *epochs* and *eras*.

An *epoch* is a time period of fixed contexts and needs under which the system operates, and it can be characterized using a set of variables that define any factor, such as technology level and supply availability, which impacts the usage and value of the system. An ordered sequence of finite-duration epochs constitutes an *era* and describes a potential progression of contexts and needs over time. Any futures relevant to system performance or costs can be described through assignments to the available epoch variables, providing a form of computational scenario planning (Roberts, Richards, Ross, Rhodes, & Hastings, 2009).

Figure 3-2 below illustrates a notional system trajectory across an ordered sequence of epochs forming an era. In this illustration, the impact of changing contexts can be seen as lowering the system's fulfillment of stakeholder needs as the system progresses over time. Rising expectations are also shown, illustrating how perception of a successful system can be dependent not only on how it performs within a context, but also how that performance compares to changing expectations. In the final epoch of the illustrated era, the system must change in order to meet expectations. In this way, Epoch-Era Analysis can structure consideration of changing contexts and needs on system success, and suggest strategies for how to sustain value in both the short run and the long run.

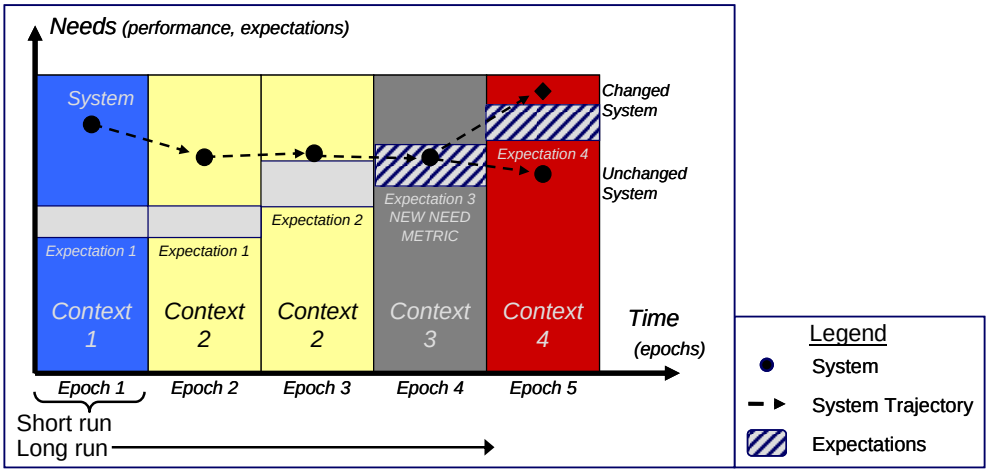


Figure 3-2. Partitioning short run and long run into epochs and eras (Ross & Rhodes, 2008)

EEA can be used with dynamic Multi-Attribute Tradespace Exploration. Evaluation of a single point design in which time-dependent performance variables are present can also be performed. Therefore, the application of EEA to designing for affordability in a system can allow analysis of value delivery for single or multiple point designs across multiple epochs and multiple eras. System engineers can thus

contribute to realizing better buying power by examining affordable systems previously overlooked or discarded (e.g. more affordable solutions may emerge from previously neglected regions of a single tradespace).

3.1.3.2 VASC

Fitzgerald created the VASC method in (Fitzgerald & Ross, 2012; Fitzgerald, 2012) to generate quantitative descriptive information on the value of included change options in systems, without resorting to many of the assumptions implicit in Real Options Analysis and other financial options methods. The VASC method was applied to two separate satellite systems, X-TOS and Space Tug. The method introduces metrics such as Fuzzy Pareto Number (FPN), effective Fuzzy Pareto Number (eFPN), effective Normalized Pareto Trace (eNPT), Fuzzy Pareto Shift (FPS), and Available Rank Improvement (ARI), which are intended to capture the behavior of designs with change options as they encounter the uncertainty space of epochs. These metrics were all applied in the Multi-Epoch Analysis section of VASC. The method also introduced the idea of defining "strategies" (e.g. Maximize Efficiency, Survive) that allow automated execution of change options for designs encountering an epoch shift. This automation of change option execution allows large-scale evaluation of designs in Multi-Epoch Analysis, as well as evaluation of designs' utilities versus costs through many eras.

3.1.3.3 Multi-Attribute Expense (MAE) as a Constructed Value Model

Designing for affordability is not only concerned with the monetary lifecycle cost of a system. While many definitions of affordability exist, there is general consensus that any evaluation of affordability must include a system's "schedule" of development and its responsiveness to emerging needs (Herald, 2011; "INCOSE Affordability Working Group Jan 2013 Summary," 2013; Mallory, 2011). However, such temporal considerations are often difficult or impossible to represent in dollars. Non-monetary measures beyond traditional forms like lifecycle cost are thus required. An additional concern is that dollars for a system are often allocated from different budgets, for example development versus operations. These different 'colors' of money may be allocated (and spent) with differing degrees of ease. Analysis without aggregating these different types of dollar budgets may provide additional insights that would otherwise be lost if dollars were aggregated into a single monetary measure.

A possible measure capable of keeping track of both monetary and non-monetary considerations, as well as keeping different 'colors' of money separate, is the Multi-Attribute Expense (MAE) function. MAE has previously been used in a satellite system design case study as an independent variable in tradespace exploration to

capture both a system's development time and initial operating costs (Diller, 2002). MAE is formulated similarly to a Multi-Attribute Utility (MAU) function (Keeney & Raiffa, 1976). Expense refers to aspects of the system design and development that the designer desires to keep at low levels, a concept akin to the notion of negative utility. Expense is principally focused on "what goes into a system" in contrast to utility, which is focused on "what comes out of a system." Typically quantified on a zero to one scale, an expense level of one denotes complete dissatisfaction and an expense level of zero denotes minimal dissatisfaction. As such, a stakeholder typically demands maximal utility and minimal expense in an ideal design (Nickel, 2010).

An MAE function requires careful construction through stakeholder interviews to elicit informed responses and aggregate preferences to capture articulated value. Just like the Multi-Attribute Utility metric, MAE is a dimensionless, non-ratio scale metric; this means that an entity with twice the MAE number over another does not imply that it is twice as expensive in terms of monetary (or any other) value.

Since temporal elements like schedule constraints and time-to-build have extensive leverage on the different 'colors' of money, the MAE can be extended to affordability applications in federal acquisition processes. Instead of comparing monetary costs against utility, MATE can be modified to compare MAE against MAU in order to perform affordability-driven analysis that captures the elements of both time and costs.

A method that leverages the EEA approach and MAE metric can allow for the effective comparison of benefits and costs across a range of alternative futures. Also, this method may transform traditional engineering practices in acquisition management if it is able to account for system changes due to shifts and perturbations, manage lifecycle differences between subsystem or subprogram components, evaluate feedback, and be adaptive to evolving system behaviors (Bobinis et al., 2012). Since affordability is a concept evaluated over time, such a method can provide options for improvement to enable enhanced design for affordability, proving to be a more representative constructed value model than a single monetary metric such as lifecycle cost.

3.2 Proposed Method based upon Epoch-Era Analysis

A method leveraging the *EEA* approach and *MAE* metric can help enable the design of affordable systems by allowing for structured evaluation of design alternatives across many alternative futures, which can help ensure that a potential design's cost is acceptable across the entire lifecycle. The method proposed in this thesis is

inspired by the *Responsive Systems Comparison (RSC)* method, which was developed earlier to support designing for changeability (Ross et al., 2008, 2009). RSC is a prescriptive operationalization of MATE and EEA, and has been previously demonstrated on applications to the design of a satellite radar system and naval ship systems (Gaspar, Ross, Rhodes, & Erikstad, 2012; Schofield, 2010). RSC is designed to "guide the…practitioner through the steps of determining how a system will deliver value, brainstorming solution concepts, identifying variances in contexts and needs (epochs) that may alter the perceived value delivered by the system concepts, evaluating key system trade-offs across varying epochs (eras) to be encountered by the system, and lastly developing strategies for how a designer might develop and transition a particular system concept through and in response to these varying epochs" (Ross et al., 2008). It is hypothesized that through modifying several original processes in RSC, incorporating recent refinements to EEA, and utilizing MAE to better capture the diversity of expenditures on a given system, the proposed method can more effectively address the time- and resource-centric approach of designing for affordability.

3.3 Description of the RSC-based Method for Affordable Concept Selection (RMACS)

The overall structure of the proposed method consists of nine processes, which are grouped into three distinct parts: information gathering (Processes 1 through 3), alternatives evaluation (Process 4), and alternatives analysis (Processes 5 through 9). A graphical representation of the method is shown below in Figure 3-3. The information-gathering portion, Processes 1 through 3, consists of defining the context and problem statement, stakeholders and respective needs, and contextual variables. The alternatives analysis portion, Processes 5 through 9, compares the dynamic properties of potential designs across the potential futures that the system may encounter. These two main portions of the proposed method are bridged by Process 4 (Design-Epoch Tradespaces Evaluation), which can provide feedback to decision makers and stakeholders, creating an opportunity to revisit the information gathering processes. Process 4 also provides cursory analysis of potential designs in preparation for the more in-depth alternatives analysis in the second half of the method.

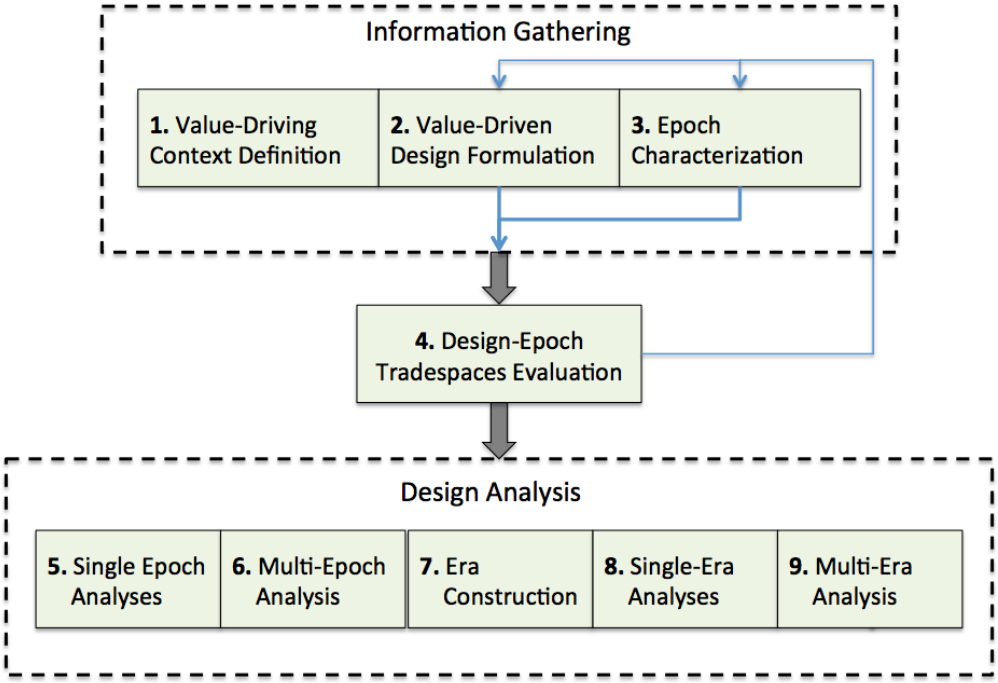


Figure 3-3. A graphical overview of the Gather-Evaluate-Analyze structure of the method.

Each process of the RMACS method, with brief description of the activities involved, is now listed, with modifications to the prior RSC method emphasized *in italics*:

Process 1: Value-Driving Context Definition
The first process of the RMACS method involves development of the basic problem statement. The stakeholders are identified, relevant exogenous uncertainties are elicited, and an initial value proposition is formed. *The resources available to each stakeholder are examined along with the associated uncertainties.*

Process 2: Value-Driven Design Formulation
The second process begins by defining the needs statements for all stakeholders, which become the attributes of system performance, along with utility functions describing each stakeholder's preference for each attribute. *The stakeholder resources statements are also elicited (with corresponding expense functions), which then become the attributes of the system's expense function.* The system solution concepts are proposed from past concepts or expert opinions. These concepts are decomposed into design variables of the system.

Process 3: Epoch Characterization
In this process, the key contextual uncertainties are parameterized as epoch variables, and possible future contexts are identified. Uncertainties in stakeholder needs are elicited. *Uncertainties in resource supply and availability are also identified, along with changes to stakeholder preferences on resource usage.*

Process 4: Design-Epoch Tradespaces Evaluation
This process utilizes modeling and simulation to map the design and epoch variables to system performance attributes *and expense attributes*. Stakeholders' utility *and expense* functions are then used to generate the MAU *and the MAE* for each design, within each epoch.

Process 5: Single Epoch Analyses
This process includes the analysis of MAU *and MAE* of alternatives within particular epochs, including designs graphically compared on an MAU vs. MAE scatterplot for any given epoch (time period of fixed operating context and stakeholder needs). Within-epoch metrics, such as yield, give an indication of the difficulty of a particular context and needs set for considered designs.

Process 6: Multi-Epoch Analysis
After completing the traditional tradespace exploration activities of Process 5, in which the practitioner compares potential designs within a particular epoch, metrics are derived from measuring design properties across multiple (or all) epochs to give insight into the impact of uncertainties on potential designs, including evaluation of short run passive and active strategies for affordability (i.e. feasible MAU and MAE, as well as efficient MAU for a given MAE). *In addition, resource usage can be analyzed to identify designs that are robust to the factors identified in Process 3 (e.g. decreasing budgets or labor availability).*

Process 7: Era Construction
This process constructs multiple sequences of various fixed duration epochs together to create alternative eras, which are long-term descriptions of possible futures for the system, its context, and stakeholder needs. This process can be performed with the aid of expert opinion, probabilistic models (e.g. Monte Carlo or Markov models), and scenarios of interest to stakeholders.

Process 8: Single-Era Analyses
This process examines the time-dependent effects of an unfolding sequence of future epochs created in Process 7. By examining a particular series of epochs for a given length of time, decision-makers can identify potential strengths and weaknesses of a design and better understand the *potential impact of path-dependent, long run strategies for affordability*.

Process 9: Multi-Era Analysis
This process extends Process 8 by evaluating the dynamic properties of a system across many possible future eras, *identifying patterns of strategies that enable affordability across uncertain long run scenarios*.

4 Application of RMACS to the Next Generation Combat Ship

"…Consider with all diligence the difference which exists
between matters of mere opinion and matters of demonstration."

-Galileo Galilei

The case chosen for an initial demonstration of RMACS was drawn from (Schofield, 2010), which describes an $8 billion Coast Guard Offshore Patrol Cutter (OPC) acquisition program for over 20 ships, each with a service life of 20 to 30 years. For this thesis, only the acquisition of one ship will be considered, and the alternatives will be limited to a few point designs rather than an exhaustive tradespace of alternative designs. Concurrent work by (Wu, 2014) extends the analysis to the program level to examine measures of affordability on multi-year and multi-unit acquisitions.

The OPC operates in a variety of areas to perform many different missions, including Ports, Waterways, and Coastal Security (PWCS), Search and Rescue (SAR), Drug Interdiction (DRUG), Migrant Interdiction (AMIO), Living Marine Resources (LMR), Other Law Enforcement (OLE), and Defense Readiness (DR) (Fabling, 2010). These mission areas include autonomous operations as well as cooperative missions with other vessels, requiring endurance and maneuverability respectively.

While the OPC and LCS systems are designed for many-unit acquisitions occurring over a period of several years, the present study examines only a single unit acquisition for the purposes of demonstrating the salient points of analysis.

4.1 Process 1: Value-Driving Context Definition

The value-driving context for the OPC is made up of the Value Propositions as well as the key Stakeholders involved in decision-making and funding. (Schofield, 2010) defines the Value Propositions for each Stakeholder as follows:

Project Office: Provide a new cutter fleet meeting operational requirements within a defined budget level and delivery to coincide with decommissioning of current WMEC fleet.

Sponsor: Develop operational requirements that meet the mission needs of the Coast Guard and Coast Guard user requirements.

Technical Authorities: Ensure new developed system meets legacy, external constraints, and design standards with technologies that maximize capability within established risk requirements.

It is clear from the value propositions that concern for resource usage is not consistent across stakeholders; as one might expect, each stakeholder has different expectations and goals with regard to resources involved in the project. The Project Office specifically addresses two standard resources: budget ("defined budget level") and schedule ("delivery to coincide with…"). The Sponsor appears to be primarily concerned with the mission needs and user requirements of the organization, and resource usage is not of primary concern. The Technical Authorities' value statement includes the aspect of technological resources that enable core capabilities.

Because each of the stakeholders' value propositions reveals the different priorities of their respective organizations, the present case of the NGCS combines the various points of view into one representative stakeholder for simplicity of analysis. This stakeholder desires to provide the new fleet of USN frigates for use in air and sea operations in a variety of operating areas. In the second process of RMACS, interviews with this stakeholder will better reveal the relevant preferences on the usage of the resources.

4.2 Process 2: Value-Driven Design Formulation

The second process builds upon the initial system context definition by first proposing the system design concept and then eliciting the attributes desired by (as well as expense attributes of importance to) the stakeholder. Through stakeholder interviews, the attributes' characteristics can be determined and weighted according to the preferences revealed. In this case, the weights placed on each attribute reflect the "combined" stakeholder from Process 1. Two types of attributes are delineated in the results: those attributes which represent resources that would ideally be conserved from the stakeholder's perspective, or "expense" attributes (e.g., acquisition cost, crew size), and those attributes which represent performance that would ideally be maximized, or "utility" attributes (e.g., range, speed). The results of this activity can be seen below in Table 4-1.

Table 4-1. Decomposition of mission statement into attributes.

Mission Statement:	Provide new frigate fleet for the USN.			
Objective:	Provide defense, air, and sea support across the defined mission areas.			
Attribute (Acronym)	**Units**	**Range ["worst" to "best"]**		**Weight (.1 to 1)**
Acquisition Cost (Acq.)	$ million	1500	500	0.4
Lifecycle Cost (LC)	$ million	7500	3000	0.3
Initial Operating Cap. (IOC)	year	2020	2018	0.2
Crew Size (Crew)	# persons	250	200	0.1
Displacement (Disp.)	cubic ft.	11000	4000	0.2
Range	nm	4500	10000	0.2
Speed	kts	12	25	0.1
Air Capability (Air Cap)	hrs/day	2	12	0.2
Endurance (Endr)	days	30	60	0.1
Small Boat Capability (SBCap)	#ppl-hrs/day	20	100	0.2

The design concepts are then partitioned into potential design variables for the proposed system. To better identify the key design drivers, the relationships of design variables to utility attributes and expense attributes are then assessed qualitatively by the values 'none', 'low', 'medium', or 'high' impact, using a Design-Value Matrix (DVM) with values of 0, 1, 3, and 9, as a visual aid for this activity. Following the example in (Schofield, 2010) of decomposing the value propositions generated in Process 1 to infer the utility attributes, the present study creates a DVM mapping the impact of design variables to the resource expenditures of the system. Thus the impacts are assessed of each design variable on each expense attribute in addition to each utility attribute, generating the DVM shown below in Table 4-2.

Table 4-2. A Design-Value Mapping, reflecting the notional impact of design variables on attributes.

Design-Value Mapping (Notional Values)												
Representative Stakeholder												
Expense Attributes	Design Variables	Length	Beam	Draft	Prop Type	Hull Material	Deckhouse Material	Defense Cabability	Helos	ASUW	AAW	Total Impact
Acquisition Cost		9	3	3	3	3	3	3	3	3	3	**36**
Lifecycle Cost		9	3	3	9	1	1	1	1	1	1	**30**
IOC		1	3	1	3	1	0	3	1	1	1	**15**
Crew Size		3	1	1	1	1	1	3	3	3	3	**20**
Total		**22**	**10**	**8**	**16**	**6**	**5**	**10**	**8**	**8**	**8**	
Utility Attributes												
Displacement		9	9	3	3	9	3	3	3	1	1	**44**
Range		9	3	1	9	3	1	1	3	1	1	**32**
Speed		9	3	3	9	3	1	3	1	1	1	**34**
Air Capacity		3	3	1	0	1	1	3	9	3	3	**27**
Endurance		1	3	1	3	1	1	3	3	3	3	**22**
Small Boat Cap.		3	3	1	0	1	1	3	3	3	3	**21**
Total		**34**	**24**	**10**	**24**	**18**	**8**	**16**	**22**	**12**	**12**	

Several benefits exist from creating such a DVM. First, by summing the rows and columns of the DVM, a practitioner can quickly determine which design variables have the most impact on general resource usage (in the notional example, the length and propeller type are the most impactful), as well as which resources are more sensitive to the present design choices (again, from the notional data, the Acquisition Cost is the most sensitive, followed by Lifecycle Cost and Crew Size). Generating an enhanced DVM, with both utility attributes and expense attributes, provides an expanded cost and benefit perspective on the ramifications of various design decisions. Second, if low-impact design variables are identified (e.g., Draft, Deckhouse Material in Table 4-2), they can be removed from the analysis to simplify the process going forward and concentrate effort on the design drivers. Finally, these impacts can be used to inform the modeling and simulation of the system necessary to evaluate system attributes in Process 4, Design-Epoch Tradespaces Evaluation.

4.3 Process 3: Epoch Characterization

After identification of the design variables, performance and expense attributes, and their corresponding relationships, the internal and external uncertainties are added into the analysis. (Schofield, 2010) lists the external uncertainties (in the associated categories) related to the OPC as follows:

Technology: VUAV integration; major C4ISR system upgrade; and new and more capable (size, range, personnel carried) small boats.

Policy: Marine engine emission reductions; reduced copper content from shipboard systems (sea water systems); increased intelligence gathering into government-wide system.

Budget: Loss of acquisition budget prior to IOC; increase in operational funding for increased operational usage.

Systems of Systems (SoS): Deploying with National Security Cutters; new cutter-deployed helicopters.

Missions: Support of arctic region for fisheries; adding environmental cleanup response capability; more frequent international presence particularly for peacekeeping missions.

Epoch variables are generated from these uncertainties by determining the primary source of the possible changes in operating context. For instance, Schofield uses the marine engine emission reductions uncertainty in the Policy category to generate the "Engine Emissions Rating" epoch variable, which has an integer value range from 2 to 4. Due to the similarities of operating contexts and missions, the epoch variables chosen for the NGCS are a subset of those outlined for the OPC. The epoch variables chosen are shown below in Table 4-3, with the corresponding category, associated ranges of values, and corresponding units.

Table 4-3. A list of the epoch variables modeled for the NGCS context and needs.

	Levels	
Epoch Variable	**[Range]**	**Units**
VUAV	[Small, Large]	Storage Level
Small Boat Size	[24, 35]	ft
Engine Emissions	[2, 3, 4]	(Tier)
Range Increase	[0, 10, 20]	%
Ice Region Use	[Low, Med, High]	Level

Once each epoch variable is created, the impact of the epoch variables on each of the design variables, performance attributes, and resource attributes can then be depicted with an Epoch Descriptor Impact Matrix, similar to the DVM in Process 2.

The complete Epoch Descriptor Impact Matrix with values (both notional and taken from (Schofield, 2010)) is shown below in Table 4-4.

Table 4-4. An Epoch Descriptor Impact Matrix, reflecting the notional impact of epoch variables on design variables, utility attributes and expense attributes.

Epoch Descriptor Impact Matrix						
Epoch Variables	VUAV	Small Boat Size	Engine Emissions	Range Increase	Ice Region Use	**Total Impact**
Utility Attributes						
Range	1	1	1	9	9	**21**
Speed	1	1	9	9	3	**23**
Displacement	1	1	0	3	3	**8**
Air Capability	3	3	0	3	3	**12**
Endurance	1	1	1	9	3	**15**
Small Boat Capability	1	9	1	3	3	**17**
Total	**8**	**16**	**12**	**36**	**24**	
Expense Attributes						
Acquisition Cost	3	3	3	1	0	**10**
Lifecycle Cost	3	3	1	9	3	**19**
IOC	1	0	3	1	3	**8**
Crew Size	3	3	1	3	1	**11**
Total	**10**	**9**	**8**	**14**	**7**	

Similar conclusions can be drawn as in Process 2; for example, it is clear from the sums of rows in Design Variables that Speed and Range are the utility attributes most impacted by the uncertainties, and Lifecycle Cost is the most impacted expense attribute. Conversely, the Range Increase epoch variable (by quite a margin) is the most impactful on all attributes, with Ice Region Use heavily impacting performance. Gaining an understanding of these relationships early in the design process allows a practitioner to begin considering how a design should be oriented to cope with uncertainties, as well as to keep in mind those contexts which are especially detrimental to the utility or expense of the system, whether directly or through opportunity costs. In addition, this mapping will aid in the evaluation of designs in each epoch and era in the subsequent processes of the method.

4.4 Process 4: Design-Epoch Tradespaces Evaluation

Once the value-driving context has been defined, along with the value-driven designs (variables and attributes) and epochs, a practitioner is ready to begin evaluating candidate designs in all epochs. The evaluation of the potential designs' attributes in the case of the NGCS was achieved through use of the MIT Math Model, which is a set of mathematical relationships developed at MIT and used for over 20 years in the design of Navy frigates for academic and government studies. The model's inputs and outputs include length, beam, draft, crew size, weapon packages and many other factors (around 50 in all). It incorporates detailed calculations of payload size, hull geometry, machinery, power and space requirements, weight, stability, and a simplified cost model. Using the model, a naval subject matter expert generated six feasible ship designs based on the design variables provided, producing the attributes of Acquisition and Lifecycle Costs, Crew Size, Range, Speed, Displacement, and IOC for the 6 representative NGCS designs. These attributes were combined with several others – Air Capability, Endurance, and Small Boat Capability – along with the notional impacts of the epoch variable levels from Process 3. The resulting (adjusted) attribute levels were mapped to stakeholder preferences through the use of single attribute utility functions. The utility curves for the levels of each attribute – normally captured through stakeholder interviews, but here generated through assessment of current capabilities and the concepts of loss aversion and anchoring in prospect theory (Kahneman & Tversky, 1979, 1984) – provide a single attribute utility value for each system attribute's level. The utility curves defined for all attributes in the Baseline epoch are shown below in Figure 4-1.

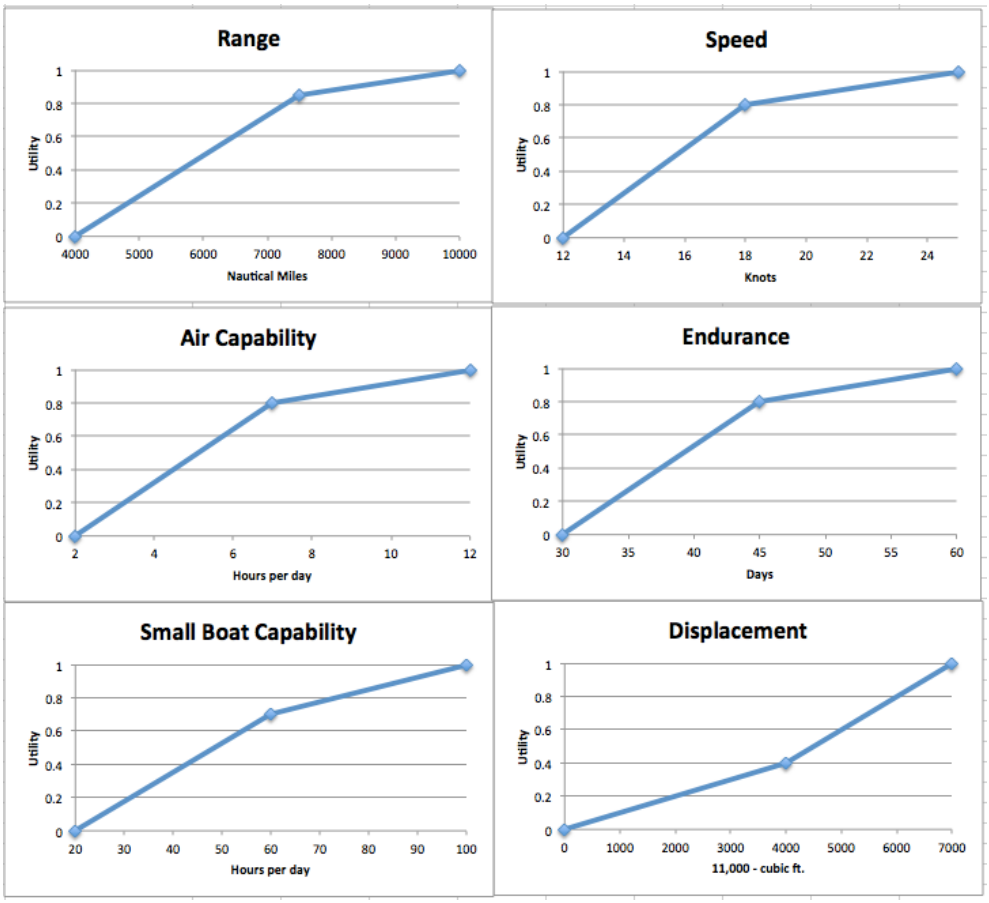


Figure 4-1. Single Attribute Utility (SAU) curves on each system attribute. The leveling off of stakeholder satisfaction/dissatisfaction occurs around levels established by previous systems.

All of the single attribute utility values are then aggregated into a Multi-Attribute Utility (MAU) value for each design point. Two key assumptions are made during this step: preferential independence among attributes and utility independence among attributes. If each attribute of the system contributes independently to utility, then the *swing weights* (relative ranking of an attribute's importance when it is at its best values and all others are at their minimally acceptable values) on each attribute sum to 1. These assumptions allow the MAU to be calculated using a simple weighted sum of the single attribute utilities:

$$\sum_{i}^{N} k_i U_i(X_i)$$

where

$$\sum_{i}^{N} k_i = 1$$

The MAU metric is commonly plotted against each design's monetary cost to help visualize a tradespace. As the present study is focused on resource usage, however, monetary cost is replaced with the Multi-Attribute Expense (MAE) metric, which captures stakeholder preference on other resource usage in addition to financial cost (e.g., initial operating capability, crew size) through the use of Single Attribute Expense (SAE) functions, akin to the SAU and MAU functions from above. These preferences are shown below Figure 4-2.

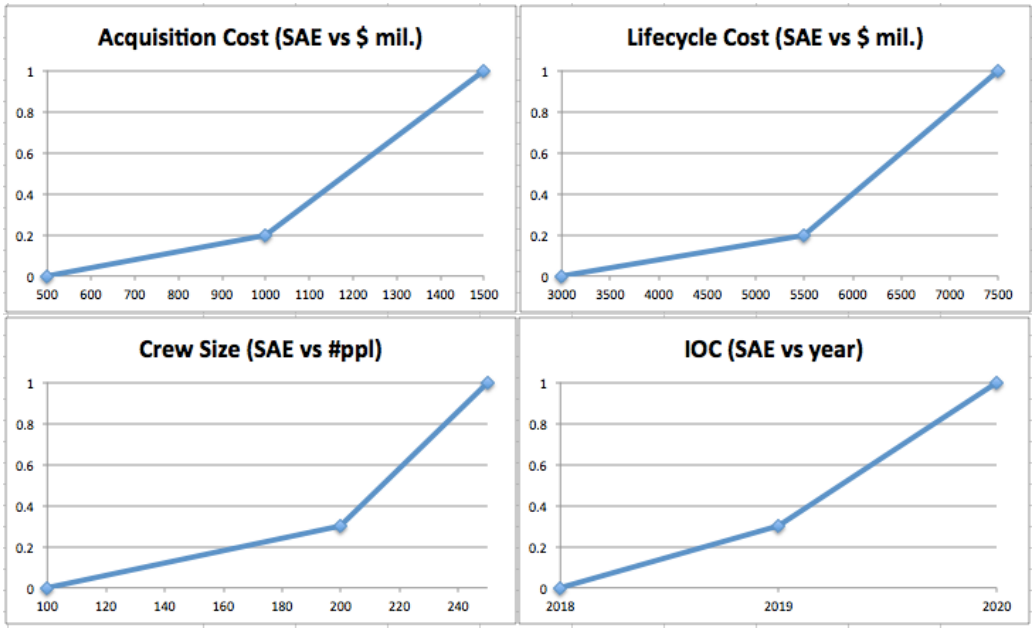


Figure 4-2. SAE functions for the expense attributes of the NGCS, where a value of 1 represents complete dissatisfaction. The "knees" in the curves represent anchoring from similar expenses of previous systems.

For the present study, six representative designs were chosen for evaluation throughout epochs, shown below in Table 4-5. The resulting evaluations of these designs in the Baseline epoch is shown in Table 4-6, which lists all of the attribute values and the resulting MAE and MAU values for each design. One of the visual results of the MAU and MAE evaluations in the Baseline Epoch is a tradespace plotting the MAU vs. MAE metrics for the handful of NGCS designs, shown below in Figure 4-3. These evaluations are performed for all designs in all epochs (a representative six epochs, in the case of the NGCS), providing the metrics necessary for the remaining steps of the method. Different sets of stakeholder preferences on utility attributes were used in addition to those shown in Figure 4-1, depending on the epoch.

Table 4-5. The six representative designs (with corresponding design variable levels) for the initial concept selection of the NGCS.

Design #:	1	2	3	4	5	6
Length (ft)	519	530	440	521	510	562
Beam (ft)	59.8	59	51.1	57.5	56.8	62.2
Draft (cu.ft.)	19.1	18.8	16.3	18.3	20.3	19.4
Prop Type	CRPP	CRPP	CRPP	CRPP	CRPP	CRPP
Hull Material	OS	OS	HTS	HTS	OS	OS
DH Material	Aluminum	Steel	Aluminum	Aluminum	Steel	Aluminum
Def Capability	Low	High	Low	Low	High	Low
Helos	2	1	1	2	2	2
ASUW	High	Medium	Low	Medium	Medium	High
AAW	High	Medium	Low	Medium	Medium	High

Table 4-6. The evaluated attributes of the six representative designs in the Baseline Epoch, with aggregated MAE and MAU values. (See Table 4-1 for units of measurement.)

			EPOCH: Baseline			
	Blue Diam.	Red Square	Green Triangle	Purple X	Blue X	Yellow Circle
Design #:	1	2	3	4	5	6
Acquisiton	1156	968	627	1011	1009	1230
Lifecycle	5086	4364	3244	4510	4505	5263
IOC	2018	2018	2018	2018	2018	2018
Crew Size	225	230	200	225	220	250
Displacement	7663	8143	4654	7734	8639	9800
Range	4500	6500	4500	6500	8000	6500
Speed	20	20	30	30	30	40
AIR CAP	8	4	2	8	8	12
ENDRNC	50	50	30	45	45	60
SB CAP	60	20	20	60	40	100
MAE:	0.36	0.28	0.13	0.29	0.28	0.41
MAU:	0.59	0.51	0.31	0.73	0.73	0.90

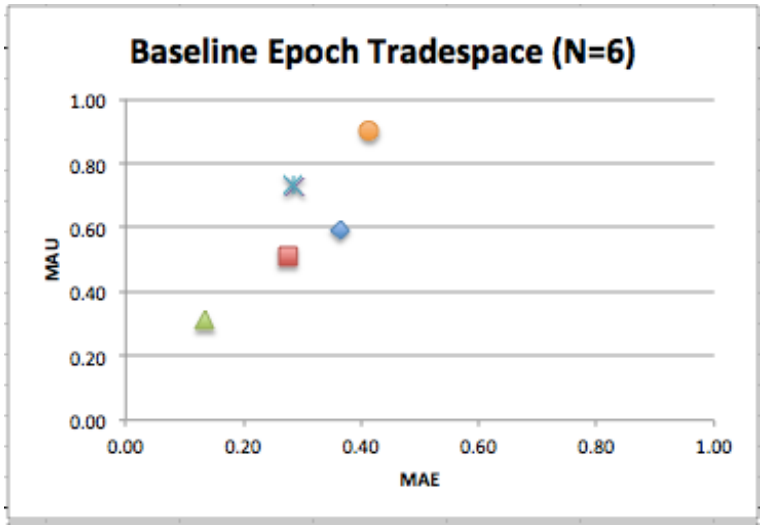


Figure 4-3. One of the results of attribute evaluation through the Math Model, epoch variable impacts, and Single- and Multi-Attribute Utilities: a tradespace (MAU vs MAE) of 6 NGCS designs operating in the Baseline Epoch.

The six epochs evaluated for the NGCS study are shown below in Table 4-7. They represent all of the 108 possible epochs from the combinations of the epoch variable levels (see Table 4-3).

Table 4-7. The six representative epochs constructed for the NGCS study (of 108 possible).

	VUAV	Small Boat Size	Engine Emissions	Range Increase	Ice Region Use
Conflict	Large	35	4	20	High
Mothership	Large	35	2	10	Low
Sojourning	Small	24	2	20	High
Sea Support	Small	35	3	10	Medium
Non-Polluter	Small	24	4	0	Low
Baseline	Small	24	2	0	Low

4.5 Process 5: Single-Epoch Analyses

Once the evaluations of Process 4 are complete for all designs in all epochs, analyses can begin of design characteristics in single epochs. In addition to the MAU and MAE metrics, the present case considers the monetary cost of each design as well as each design's Pareto efficiency in the tradespace. This analysis can be repeated for any number of epochs of interest, which can be chosen through various means – those most likely to occur, those most likely to hinder value delivery, or those of concern for other reasons to stakeholders and analysts. The present study chooses only a few of the epochs created in Process 4: Mothership, Sea Support, and Conflict.

Mothership is an epoch characterized by Large VUAVs and 35ft Small Boats, with a 10% increase in Range of mission over the Baseline Epoch. This combination of epoch variables represents a period in which stakeholders would desire the NGCS to support air and sea operations over non-Arctic waters. The evaluated designs for this epoch are shown below in Table 4-8, and the corresponding tradespace is depicted below in Figure 4-4.

Table 4-8. The evaluated attributes of the 6 representative NGCS designs in the Mothership epoch. (See Table 4-1 for units of measurement; negative attribute values are treated as 0.)

			Mothership			
	Blue Diam.	Red Square	Green Triangle	Purple X	Blue X	Yellow Circle
Design #:	1	2	3	4	5	6
Acquisiton	1274	1067	691	1115	1112	1356
Lifecycle	6167	5293	3934	5470	5464	6382
IOC	2018	2018	2018	2018	2018	2018
Crew Size	250	255	225	250	245	275
Displacement	7663	8143	4654	7734	8639	9800
Range	4500	6500	4500	6500	8000	6500
Speed	18	18	27	27	27	36
AIR CAP	0	-4	-6	0	0	4
ENDRNC	45	45	25	40	40	55
SB CAP	50	10	10	50	30	90
MAE:	0.47	0.36	0.21	0.39	0.38	0.51
MAU:	0.38	0.18	infeasible	0.54	0.53	0.75

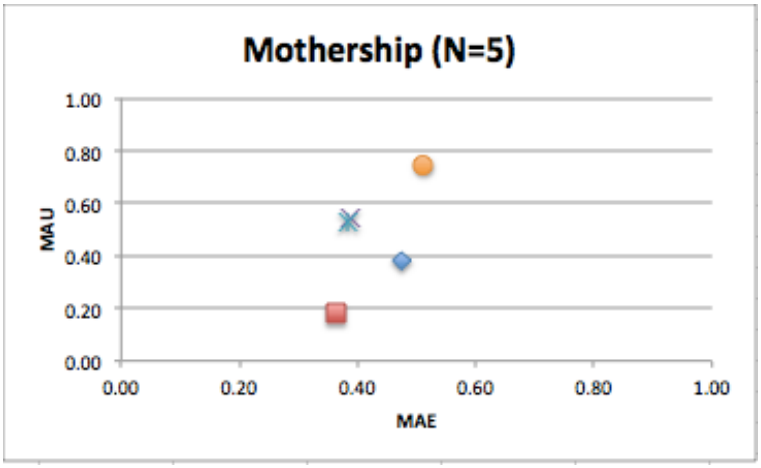


Figure 4-4. The 6 potential NGCS designs in the Mothership epoch. (One design is below minimum acceptable utility, leaving the 5 feasible designs.)

It is important to note that in general, the MAU and MAE values cannot be compared between epochs, since stakeholder preferences may change between epochs (i.e., and therefore change what the 0-to-1 scale represents, as it is not a universal scale). The yields of the tradespaces can always be compared, however, providing the number of designs at or above the minimum acceptable utility and at or below maximum acceptable expense. While six designs were evaluated in each epoch, only five are feasible in the Mothership epoch, indicating that this epoch provides some challenges for at least one potential design to provide minimum acceptable utility and/or operate below maximum acceptable expense. Table 4-5 shows that Design #3, being a smaller design, only provides room for 1 helicopter and has low levels of anti-surface and anti-air defenses. These design variable levels may provide adequate levels of satisfaction in the Baseline epoch, but when the

context and preferences change on attribute levels, this design no longer provides even minimally acceptable value.

Because acquisition and lifecycle (including operations) costs can be limiting factors compared to the other expenses (e.g., crew size) rolled up in the MAE metric, the monetary costs of each design are briefly observed in each epoch of interest. Design #6 has an acquisition cost of $1.3 billion and a lifecycle cost of $6.4 billion, while Design #2's acquisition cost is 30% less (~$1 billion), and its lifecycle cost is around 15% less ($5.3 billion). The other designs' costs are in the middle of these two designs. If budget levels were established for the stakeholder in this epoch, those considerations could aid in the comparison of these costs. In addition, Process 6 will examine more informative cost metrics over all epochs, with or without established budget levels.

The Fuzzy Pareto Number (FPN) is a metric developed to indicate a design's relative value in a given epoch, as it measures how far from Pareto-optimality that design lies in the tradespace of that epoch (Fitzgerald & Ross, 2012). Since it represents a percentage of deviation from cost/utility Pareto efficiency, it is measured from 0 to 100; in addition, since it is a percentage, it can be compared across epochs as an indicator of relative efficiency differences. In the Mothership Epoch, the FPN for 4 of the designs is zero due to their locations on the Pareto front. The remaining design, Design #1, has an FPN number of 20, meaning that it is 20% "inefficient" compared to the Pareto front in this epoch.

These analyses are now briefly discussed for the rest of the epochs of interest.

Sea Support is an epoch during which the NGCS would be required to support extended missions with very capable small boats over a wide range of global waters. It assigns the following values to the epoch variables (various possible levels described in Process 3): Small Boat Size of 35ft, Emissions standards at Level 3, Range Increase of 10%, and High Ice Region Use. The evaluated attributes of all designs are shown in Table 4-9 below; the tradespace of MAU vs. MAE follows in Figure 4-5.

Table 4-9. The evaluated attributes of each design in the Sea Support epoch. (See Table 4-1 for units of measurement; negative attribute values are treated as 0.)

			Sea Support			
	Blue Diam.	Red Square	Green Triangle	Purple X	Blue X	Yellow Circle
Design #:	1	2	3	4	5	6
Acquisiton	1274	1067	691	1115	1112	1356
Lifecycle	5887	5052	3755	5221	5216	6092
IOC	2020	2020	2020	2020	2020	2020
Crew Size	240	245	215	240	235	265
Displacement	7663	8143	4654	7734	8639	9800
Range	3600	5200	3600	5200	6400	5200
Speed	15	15	23	23	23	31
AIR CAP	2	-2	-4	2	2	6
ENDRNC	40	40	20	35	35	50
SB CAP	40	0	0	40	20	80
MAE:	0.70	0.60	0.44	0.62	0.61	0.75
MAU:	0.14	0.15	infeasible	0.46	0.46	0.71

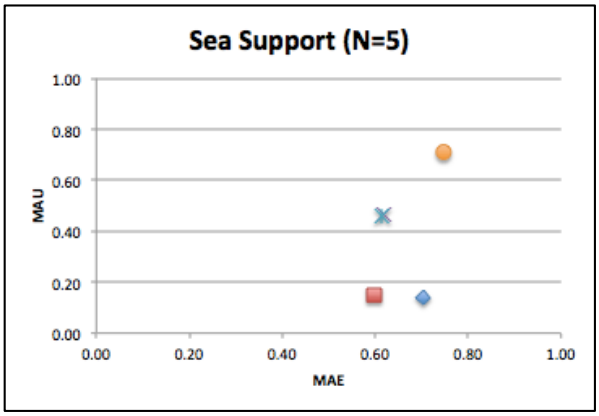


Figure 4-5. The 4 feasible NGCS designs in the Sea Support epoch.

The yield of this epoch is again five out of the six evaluated designs. From Table 4-9, we can see that the acquisition costs are similar to that of the Mothership epoch, but the lifecycle cost (and therefore operations cost) is slightly less in this epoch. We can also see that the IOC is pushed back to the year 2020 for all designs. The FPNs for Designs #2, #4, #5 and #6 are once again zero, while the FPN for Design #1 is 38.

The third epoch of interest considered, Sojourner, is defined by a Range Increase of 20% plus High Ice Region Use. The stakeholder's preferences in this epoch are agnostic of the air and small boat capabilities of each design. Table 4-10 displays the evaluated attributes of each design in this epoch, and the tradespace of MAU vs. MAE is shown below in Figure 4-6.

Table 4-10. The evaluated attributes of each design in the Sojourner epoch. (See Table 4-1 for units of measurement; negative attribute values are treated as zero.)

			Sojourner			
	Blue Diam.	Red Square	Green Triangle	Purple X	Blue X	Yellow Circle
Design #:	1	2	3	4	5	6
Acquisiton	1156	968	627	1011	1009	1230
Lifecycle	6141	5270	3917	5446	5440	6355
IOC	2019	2019	2019	2019	2019	2019
Crew Size	245	250	220	245	240	270
Displacement	7663	8143	4654	7734	8639	9800
Range	3600	5200	3600	5200	6400	5200
Speed	15	15	23	23	23	30
AIR CAP	2	-2	-4	2	2	6
ENDRNC	30	30	10	25	25	40
SB CAP	35	-5	-5	35	15	75
MAE:	0.51	0.41	0.27	0.43	0.42	0.55
MAU:	0.04	0.33	infeasible	0.23	0.27	0.59

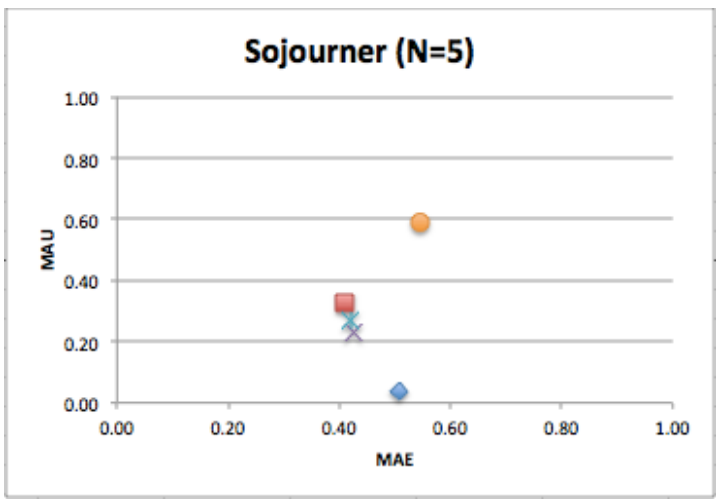


Figure 4-6. The 5 feasible potential NGCS designs in the Sojourner epoch.

Once again, the yield of this epoch is five of the six designs; only Design #3 is infeasible. From Table 4-10, we can see that the acquisition costs are similar to that of the previous epochs considered, but the lifecycle costs (and therefore operations costs) of all designs are around the Mothership epoch's levels. It is of interest to note

that only two designs remain on the Pareto front: Designs #2 and #6. The FPN for Design #4 is 4, the FPN for Design #5 is 3, and Design #1's FPN is 23.

4.6 Process 6: Multi-Epoch Analysis

The analyses of Process 5 can be enlightening regarding the behavior of individual designs and the impacts of individual epochs, but the observations can be time-consuming (and the resulting data overwhelming) for any significant number of designs and/or epochs. The analyses can also allow undue weight to be placed on the epochs of interest over those omitted from explicit consideration, even though all epochs represent possible operating environments. For this reason, Process 6 focuses on several summary metrics to gain a higher-level view of all designs' characteristics *over all epochs*. (Since only 6 epochs are used as representative of all epochs in this study, their tradespaces are shown side by side for easy comparison in the attached Appendix.) No weight is given to epochs based on likelihood of occurrence, as the purpose of this analysis is simply to cover all possible scenarios the system might encounter. The first of the metrics designed for this purpose is the Normalized Pareto Trace.

The Normalized Pareto Trace (NPT) reflects the percentage of all epochs for which a given design is Pareto efficient (Ross et al., 2009). A higher NPT indicates higher Pareto efficiency for a design over *all* epochs, rather than higher Pareto efficiency in only one epoch (like the FPN). It is calculated for any design by counting the number of epochs in which that design has an FPN of zero and then dividing by the total number of epochs. The six potential NGCS designs and their corresponding NPTs across the six representative epochs are listed below in Table 4-11.

Table 4-11. The NPTs for the 6 NGCS designs across the 6 representative epochs.

	DESIGN	NPT
BLUE DIAMOND	1	0
RED SQUARE	2	1
GREEN TRIANGLE	3	0.33
PURPLE X	4	0.5
BLUE X	5	0.67
YELLOW CIRCLE	6	1

Clearly Designs #2 and #6 would be good choices when high priority is placed on Pareto efficiency, as they remain on the Pareto front no matter which epoch encountered. Design #6 is the most expensive and brings the most utility in every epoch, while Design #2 is one of the lower expense designs and is closer to minimal acceptable utility in most epochs.

The somewhat simplistic measure of NPT can be extended by allowing some "fuzziness" threshold in the evaluation of efficiency. The fuzzy Normalized Pareto Trace (fNPT) is a metric that does precisely this: it applies a specified fuzziness percentage to the Pareto front in each epoch, where 0% is a normal Pareto front, and 100% includes the entire range of both the MAU and MAE of the designs in a given epoch (Fitzgerald & Ross, 2012). The fNPT of each design at several fuzziness levels is shown below in Table 4-12.

Table 4-12. The fuzzy NPTs of the 6 NGCS designs for several levels of fuzziness. A value of 1 represents fuzzy Pareto efficiency in 100% of the epochs.

	DESIGN	0% FNPT	5% FNPT	10% FNPT	20% FNPT
BLUE DIAMOND	1	0	0	0	0.17
RED SQUARE	2	1	1	1	1
GREEN TRIANGLE	3	0.33	0.33	0.33	0.33
PURPLE X	4	0.5	1	1	1
BLUE X	5	0.67	1	1	1
YELLOW CIRCLE	6	1	1	1	1

The results of the fNPTs show that with a small amount of fuzziness – around 5% – most of the NGCS designs have maxed out their fNPT number (due in part to the small number of designs considered). Design #1 remains inefficient even after the fuzziness level approaches 20%, decreasing this design's attractiveness compared to the others. With more designs under consideration, the benefits of the fNPT metric should be evident: helping identify designs that may be near-Pareto efficient in many epochs and resultantly missed by the original NPT (such as Designs #4 and #5 in this case).

4.6.1 Aspects of changeability

The metrics shown so far do not take into consideration the possibility of changing from one design to another in a given epoch. This possibility is usually present in some form, however, and as a result it helpful for Multi-Epoch Analysis to evaluate a design based on the strategies used for change – that is, an original design can be evaluated in Multi-Epoch Analysis by evaluating the *target design* (to which the original changes, if applicable) in each epoch. Strategies can be defined to guide the change behavior; for the NGCS case, the strategy chosen was "Maximize Efficiency" (i.e. move to the Pareto front if not already there). Transition rules were created such that any design could be discarded and any design purchased in any epoch. A transition matrix was then constructed to reflect the resulting target design (dictated by the chosen strategy) in each epoch from each original design in that epoch. With this additional information on the anticipated transitions and resulting designs in each epoch, modified forms of the previous metrics were constructed and are now discussed.

The *effective* NPT (eNPT) and *effective* fNPT (efNPT) metrics evaluate a design across all epochs in the following way: if the change strategy dictates that a (original) design changes to another (target) design in a given epoch, then the target design is evaluated; if the change strategy dictates that a starting design does not change in that epoch, then that starting design is evaluated (Fitzgerald, 2012). The eNPTs for the NGCS designs, generated by the change strategy "Maximize Efficiency" discussed above, are shown below in Table 4-13.

Table 4-13. The eNPTs for the NGCS designs with the change strategy "Maximize Efficiency".

	Design	eNPT	eNPT, with $ budget (notional)	eNPT, with time budget (notional)
BLUE DIAMOND	1	1	0.3	0.17
RED SQUARE	2	1	1	1
GREEN TRIANGLE	3	1	0.5	0.45
PURPLE X	4	1	0.85	0.75
BLUE X	5	1	1	.7
YELLOW CIRCLE	6	1	1	1

As the table shows, most designs' base eNPT is a great improvement over the NPT. In fact, it appears that every design can achieve Pareto efficiency in every epoch – which is true because of the transition rules defined and the transition strategy chosen. Recall that the transition rules defined for the NGCS case state that every design can be discarded and any other design purchased in its place in any epoch, while the transition strategy dictates movement to the Pareto front in every epoch.

If smaller-scale changes than discard/replace were available (e.g., "add UAV storage", "remove crew", etc), rules could be established to represent the feasibility/infeasibility of changing from one design to another in any epoch, thereby limiting the improvement of a design's eNPT over its NPT. Likewise, transition costs of money and time could be defined for any of the transitions between designs, and these costs could be used to limit feasibility (depending on the budget/goal of the transition strategy defined) as well as to track the total expenditures necessary to achieve a given eNPT. In the notional columns on the right of Table 4-13, it can be seen that budgets of time or financial costs would prevent Design #1, for instance, from improving very much. Likewise, the other designs may improve or not. (The effective fNPT (efNPT) metric can also be constructed, but is left out of the present study due to the already-maximized eNPT values above.)

4.6.2 Additional aspects of affordability

In addition to the efficiency of a design relative to other designs in each epoch, it can be useful to consider the resource expenditures of a design in various operating environments. Two ways of measuring expenditures across all epochs are applied to the NGCS case: one tracks the maximum amount required of each resource for a given design, and the other tracks the stability of resource consumption throughout all epochs. The first metric can help identify designs that would be unsustainable given the right conditions, while the second metric can identify designs for which allocating resources in the future (e.g., Congressional budget requests) may prove easier due to consistency through changing operational environments. In the case of the NGCS, the highest levels of expense for each design's Lifecycle Cost and Crew Size in all (6 representative) epochs are shown below in Table 4-14.

Table 4-14. The highest levels of Lifecycle Cost and Crew Size across all epochs.

Max Expense	Design					
	1	2	3	4	5	6
Lifecycle Cost ($ mil.)	7,093	6,087	4,524	6,290	6,284	7,340
Crew Size	260	265	235	260	255	285

Recalling the Normalized Pareto Trace from earlier, efficient designs #2 and #6 can be selected for comparison of their respective maximum expenses incurred. While Design #2 could cost a maximum of just over $6 billion in its lifetime, Design #6 could cost over 20% more in one possible scenario. In addition, Design #6 could require 25 more crewmen depending on the epoch encountered. Note that multiple designs' maximum expenses are not all necessarily from the same epoch, as epoch variables impact individual designs differently. (Likewise with a design's multiple expense attributes: each of a design's attributes are impacted differently by the epoch variables, so that the lifecycle cost may be highest in one epoch, while maximum crew size is required in another.)

The standard deviation of each design's lifecycle cost is shown below in Table 4-15. Continuing our analysis from the Max Expense, it can be seen that Design #2 has somewhat less variability across epochs than Design #6, by around $400 million. If budget forecasts were known at the time of this analysis, such information could be used effectively to choose a design that fit within the expected variability of the budget for the timeframe of the forecast.

Table 4-15. The standard deviation of Lifecycle Cost for each design across all epochs.

Expense Stability			Design			
(St.Dev. in $ mil.)	1	2	3	4	5	6
Lifecycle Cost	2,156	1,850	1,375	1,912	1,910	2,231

These considerations, when combined with previous analyses, help outline the impact of the risks involved for any initial concept selection, and they can help analysts and decision makers alike understand the traits behind design concepts that would be affordable in all futures.

4.7 Process 7: Era Construction

The analysis up to this point has evaluated NGCS designs in epochs, or short-run periods of fixed contexts and needs. By combining these short-run fixed periods into a sequence over a longer period of time, an *era* is created – allowing a practitioner to study the attributes of designs over one possible development of the operating environment, as well as the effects of path dependence through epochs (Ross & Rhodes, 2008). Two eras were manually constructed for the present study, each representing a ten-year sequence of epochs that the NGCS may encounter (taken from the six representative epochs from Process 4, c.f. Table 4-3). The first era

considered consists of the following 4 *frames*, where a frame comprises an epoch and a specified duration of time:

Frame Number	1	2	3	4
Epoch Name	Baseline	Sea Support	Baseline	Non-Polluting
Duration (months)	36	36	24	24

The second era considered also comprises 4 frames, but with different epochs and durations:

Frame Number	1	2	3	4
Epoch Name	Sojourning	Conflict	Mothership	Sojourning
Duration (months)	24	36	36	24

4.8 Process 8: Single-Era Analyses

The first era comprises epochs in which stakeholder preferences do not change. As a result, the MAE and MAU values can be compared directly. The era is shown below in Figure 4-7 (each tradespace) and Figure 4-8 (line graph view).

(36 months) (36 months) (24 months) (24 months)

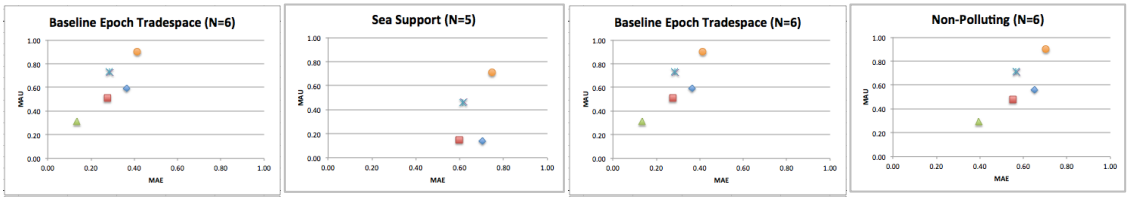


Figure 4-7. The four epochs of Era #1 and their durations. Stakeholder preferences remain constant; as a result, MAU and MAE values can be directly compared across epochs.

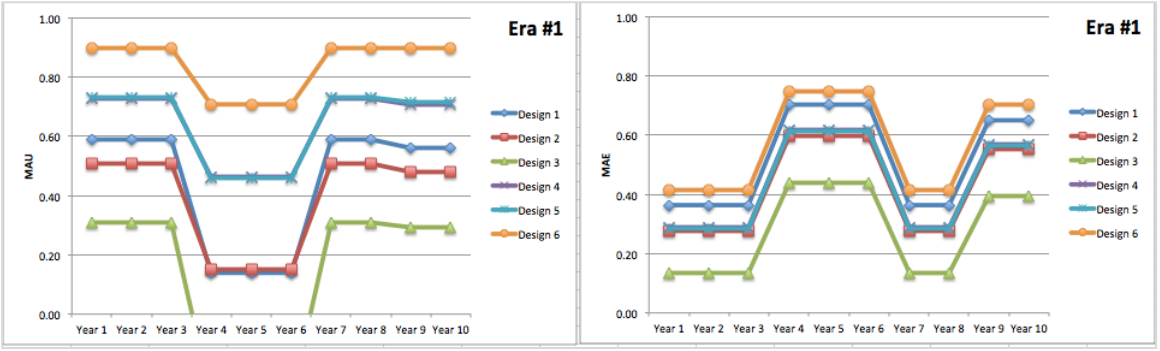


Figure 4-8. Left: the MAU values of each design during Era #1. Right: the MAE values of each design in Era #1.

Before proceeding, it is noted that Design #3 is infeasible in the Sea Support epoch, meaning that design cannot provide minimum acceptable utility throughout this era. If a decision maker believes this era to be one of the more likely narratives to play out for the NGCS, then that design could be removed from consideration.

Using the concept of time value of money, the NPV of each design's operations cost can be calculated for the entire era. (To obtain the operations cost for a given timeframe, the acquisition cost was subtracted from the 30-year lifecycle cost, and the result divided into the appropriate number of months.) For instance, Design #1's yearly operations cost in the Baseline Epoch is $131 million; this amount is used as the input to the final NPV calculation. This same calculation is performed for each design in each epoch in the era, and a 10% discount rate is assumed. The intermediate values and final calculation are shown below in Table 4-16.

Table 4-16. Calculation of the net present value ($ millions) of designs' operations costs for Era #1.

			Era #1			
($ millions)	**Design 1**	**Design 2**	**Design 3**	**Design 4**	**Design 5**	**Design 6**
NPV Ops Yr. 1	131	113	87	117	117	134
NPV Ops Yr. 2	119	103	79	106	106	122
NPV Ops Yr. 3	108	94	72	96	96	111
NPV Ops Yr. 4	116	100	77	103	103	119
NPV Ops Yr. 5	105	91	70	93	93	108
NPV Ops Yr. 6	95	82	63	85	85	98
NPV Ops Yr. 7	74	64	49	66	66	76
NPV Ops Yr. 8	67	58	45	60	60	69
NPV Ops Yr. 9	59	51	40	53	53	61
NPV Ops Yr. 10	54	47	36	48	48	55
NPV Total:	**929**	**803**	**618**	**827**	**826**	**953**

From these numbers, it appears that Designs #2, #4, and #5 form one group with similar NPV operations costs, and that Designs #1 and #6 form another similar group. While these numbers reflect the same grouping of designs as the MAE suggests in the tradespace plots, it is important to note that the MAEs could be similar for certain designs whose operations cost's NPV are vastly different. For this reason, it can be necessary to consider these costs separately from the MAE score itself if information about the operations budget is known at this stage.

The additional affordability considerations from Process 6 can also be repeated at this point: the maximum values of the operations costs of each design are shown below in Table 4-17, and the standard deviation of operations costs throughout the era is shown in Table 4-18.

Table 4-17. The maximum yearly operations cost of each design throughout Era #1, compared with the NPV maximum operations cost

($ million / yr)	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Max Ops Cost:	154	133	102	137	137	158
Max Ops Cost (NPV):	131	113	87	117	117	134

Table 4-18. The standard deviation of each design's operations costs throughout Era #1.

(St.Dev. in $ mil.)	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Expense Variability	11.0	9.5	7.1	9.8	9.8	11.4

In this case, the results from the era analysis reflect the same conclusion as Process 6 (over all epochs): the more expensive designs have slightly more variance than the less expensive designs. If more designs were under consideration, it may be possible to find a more expensive design that is more cost-stable across all epochs (and/or in a particular era), which may give incentive to include it when considering designs for final selection. Of course these metrics can be considered for any resources of particular concern, but are only applied here to operations cost for demonstration purposes.

The second era, shown in Figure 4-9 below, consists of epochs with changing stakeholder preferences, and so the MAU and MAE values cannot be compared directly across epochs.

(24 months) (36 months) (36 months) (24 months)

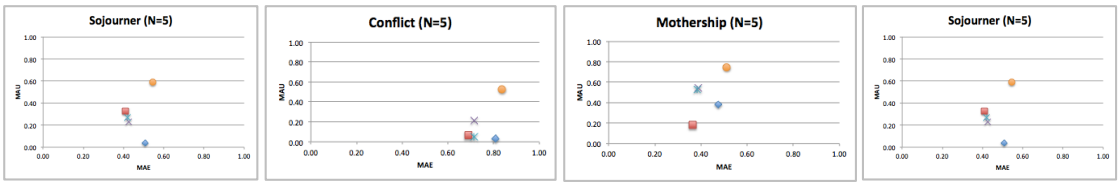


Figure 4-9. The four epochs of Era #2. The MAU and MAE values cannot be compared between epochs due to changing stakeholder preferences.

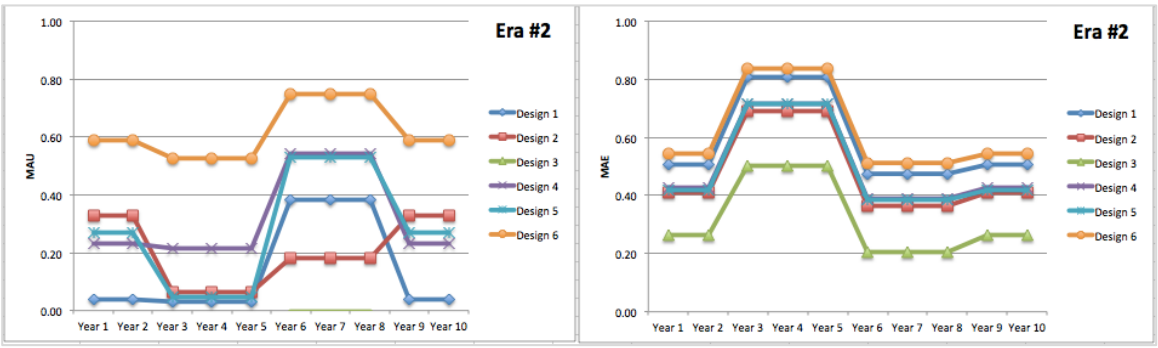


Figure 4-10. Left: the MAU of each design through Era #2. Right, the MAE of each design during the same ten-year period. Note that stakeholder preferences change between epochs.

It is again noted that Design #3 is infeasible in this era; in fact, it is not feasible in even one of these epochs. Design #2, while starting the era highly ranked in the MAU measurement, makes its way to the least favorable ranking for a short time. Its expenses remain least dissatisfying, however, among feasible designs in this era. The NPV calculations for all designs in this era are shown below in Table 4-19.

Table 4-19. Calculation of the NPV operations costs of each design in the 10-year Era #2.

			Era #2			
($ millions)	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
NPV Ops Yr. 1	166	143	110	148	148	171
NPV Ops Yr. 2	151	130	100	134	134	155
NPV Ops Yr. 3	157	135	104	140	139	161
NPV Ops Yr. 4	143	123	94	127	127	146
NPV Ops Yr. 5	130	112	86	115	115	133
NPV Ops Yr. 6	101	87	67	90	90	104
NPV Ops Yr. 7	92	80	61	82	82	95
NPV Ops Yr. 8	84	72	55	74	74	86
NPV Ops Yr. 9	78	67	51	69	69	80
NPV Ops Yr. 10	70	61	47	63	63	72
NPV Total:	**1171**	**1011**	**774**	**1042**	**1041**	**1204**

It should be obvious that this era represents a particularly challenging environment for the designs, with two of them exceeding one billion dollars for the NPV operation costs. The maximum values and the standard deviations of operations costs for all designs are shown below in Table 4-20 and Table 4-21.

Table 4-20. The maximum operations cost incurred by each design throughout Era #2, compared with the maximum NPV operations cost.

($ million / yr)	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Max Ops Cost:	190	164	125	169	169	195
Max Ops Cost (NPV):	166	143	110	148	148	171

Table 4-21. The standard deviation of operations cost throughout the 10 years of Era #2.

(St.Dev. in $ mil.)	Design 1	Design 2	Design 3	Design 4	Design 5	Design 6
Expense Variability	11.5	9.9	7.6	10.2	10.2	11.8

4.9 Process 9: Multi-Era Analysis

This process extends Process 8 by evaluating the dynamic properties of a system across many possible future eras, identifying patterns of strategies that enable affordability across uncertain long run scenarios. It incorporates all of the information generated up to this point in the method and can provide key insights as to *how* change options should be used to achieve maximum stakeholder value. As for the eras chosen to study, any number of eras can be generated through the means noted in Process 7. Alternative methods of generation can also be used to analyze shorter sequences of epochs. Chapter 5 of this thesis provides an in-depth explanation of the nature of this process, activities involved, and outputs provided. Chapter 6 demonstrates an application of this process to an NGCS-like ship system, with several change options added for more comprehensive analysis.

Discussion of the "Affordable" NGCS design

Since affordability has been defined in terms of feasibility, several observations can be made regarding affordable design choices for the NGCS system. First, Design 3 was consistently the lowest-resource design, and might have been the affordable choice if only the Baseline Epoch was considered. However, its lack of feasibility in many of the epochs makes it a poor design choice, since its performance does not enable the minimum required capabilities in those epochs. Second, the MAE metric shows that Designs 2, 4, and 5 all have similar resource expenditures in most epochs,

while Designs 1 and 6 require relatively more. These observations could provide good motivation for selecting the lower-expense designs (2, 4, 5) as designs of interest as early in the method as Process 6. The Max Expense metric applied in Process 6 reveals that Design 2 could require slightly more crew (265 crewmen) than Designs 4 and 5 (260 and 255, respectively); but Design 2 could also cost less (albeit not by much) over its lifecycle ($6.1 billion). Finally, when Expense Stability is considered, it shows that the designs currently under consideration all behave similarly throughout the epochs. Keeping in mind the definition of affordability as feasibility, it can be concluded that these three designs are indeed affordable choices. Accordingly, other measures (e.g., stakeholder satisfaction) can be used to down select from the initial designs of interest. Observing Design 5's decreased value in the Conflict epoch could result in selecting only Design 4, which requires similar expenditures but provides more stable value delivery throughout Conflict and other epochs. Designs 2 and 4 can be compared to identify the design variables and attributes that potentially enable affordability in these designs. For example, two common design variables are length (520–530 feet) and levels of Anti-Surface/Anti-Aircraft capabilities (both medium level). These common traits can be studied to provide the general interaction between these particular variables, the resulting ship attribute levels, and the stakeholder preferences on those attributes. The affordability analysis performed through the application of this method thus naturally leads into a study of the common traits of affordable NGCS solutions. Rather than concluding the study with one solution deemed "most affordable," the NGCS stakeholder can gain a better idea of early design decisions' impacts on the lifelong affordability of the chosen system design.

5 Multi-Era Analysis: An Exploration for Path Dependencies of Epochs and Designs across All Modeled Futures

> "…to a large degree, knowledge of the inevitable: of what, given our world order, could not but happen; and conversely, of how things cannot be, or could not have been, done; of why some schemes must, cannot help but, end in failure..."
>
> -Isaiah Berlin, *The Hedgehog and the Fox*

This chapter explores the nature of Multi-Era Analysis (MERA) and describes the various considerations required and activities involved. The chapter begins with a short discussion of the goal of MERA as contrasted with the goal of Multi-Epoch Analysis (MEA). The information requirements to conduct the study are then defined, along with the optional informational components. The variants of MERA, the major factors involved, and those factors' impacts on the study are discussed. Era-level strategies are defined and discussed, along with example algorithms for generating design trajectories through an era from a given strategy. Some tentative approaches to era-level path analysis are then outlined. The activities involved in the MERA process are described, with emphasis on the exploratory nature of the process. The Descriptive information outputs are then defined for the variants of MERA process.

5.1 Goals of Multi-Era Analysis

The goal of MERA is not strictly analogous to the goal of Multi-Epoch Analysis encountered earlier. Recall that the goal of Multi-Epoch Analysis is to apply descriptive metrics to designs over the entire space of epochs (i.e., the *tradescape*, defined as the set **V** below). There is neither any temporal aspect to Multi-Epoch Analysis, nor is there any consideration for transitions between epochs, making possible the prospect of *completely* exploring the uncertainty space that has been modeled as fixed periods of context and needs. Certainly a similar goal could be derived for MERA in its simplest forms – e.g., each epoch in an era lasts exactly for 1 year, and each epoch transitions to exactly one other specific epoch with 100% probability. In this (rather degenerate) case, the uncertainty space of modeled eras can be completely explored.

With all other cases of MERA, however, there can be both temporal and transitional rules and dependencies on both the epoch- and design-level, as well as combinatorial factors that enhance the difficulty of enumerating or even sampling the space. These factors create conditions under which the uncertainty space can

never be completely explored. The intent of Multi-Era Analysis aims not to cover or characterize the complete uncertainty space, which is intractable for a reasonably sized problem (see Section 5.4.5 for in-depth discussion of this point). Rather, Multi-Era Analysis aims to glean information from identifying relevant path dependencies that stem from two distinct factors:

1) The path dependency of perturbations, which include disturbances, and epoch shifts,

2) The path dependency of changes to a design as it progresses through its lifecycle.

By better understanding these path dependencies, the ultimate goal of MERA, then, is to aid in *identifying patterns of strategies* for value sustainment over many possible lifecycles, based on the particular path-dependent developments of designs, contexts, and needs. This goal can be conceptually broken into two distinct types of information[1]:

1) **Descriptive Information**, the primary goal of the process. The information produced should aid in evaluating and comparing some number of separate design alternatives, or evaluating and comparing some number of alternatives *with change options and strategies*.
2) **Prescriptive Information**, an optional goal of the process. This activity would consist of determining the optimal action (change vs. no change) when considering an individual design alternative in a given epoch with an unknown future, with multiple change options available to execute, and one or more era-level strategies present.

As will be discussed in further detail in Section 5.3.1.3, if change options are not present, then there is likely no need for the optional Prescriptive Information portion of the Multi-Era Analysis (with one exception being the determination of the timing of retirement/replacement of an existing system). Additionally, if change options are not present, the need for Descriptive Information from MERA implies either that there is some meta-information regarding epoch transitions, or that disturbances are being modeled. If no such information is present regarding epoch transitions, as will be discussed in Section 5.3.2, and if no disturbances are being modeled, then neither type of path dependence (of epochs or designs) exists in this case, eliminating the need for the MERA process altogether; the MERA process devolves into Multi-Epoch Analysis. If epoch duration meta-information is available, then the Multi-Epoch Analysis is simply one with a form of "weighting"

[1] Only the Descriptive information portion is studied in the present work. The Prescriptive portion is discussed in the Further Research section of Chapter 7.

on each epoch under consideration (see Section 5.3.2 for more discussion on this point).

5.2 Inputs to Multi-Era Analysis

The amount and variety of information that can be incorporated into MERA is significant. The *required* inputs to this process are:

1) A set **X** of design vectors, where each X is a unique configuration of design variable levels.

   Example: Design #4 = X_4 = [length: 500ft, beam: 30ft, draft: 40ft, …]

2) A set **E** of epoch vectors, where each E is a unique configuration of epoch variable levels (i.e., context variable levels and a set of stakeholder value models). For the present work, Single-Attribute Utility and Multi-Attribute Utility (and the corresponding Expense) functions are the stakeholder value models in use. The set **E** makes up the context + needs (i.e., epoch) space.

   Example: Epoch #2 = E_2 = [missionType: 3, fuelCost: 120%, Arctic: Low, …]

3) A set **Y** of attribute vectors, where each Y is an evaluated performance model output for each element in the cross product **X** x **E**.

   Example: Attributes for design-epoch pair <X_4, E_2> = $Y_{X_4E_2}$ = [performanceModel(X_4, E_2)] = [range: 4000nm, maxSpeed: 30kn, LCC: 240, …]

4) A set **V** of value vectors, where each V is a value model output for each attribute vector Y in **Y** (and therefore for each element in the cross product **X** x **E**).

   Example: Value vector for design-epoch pair <X_4, E_2> =

   V_{X_4,E_2} = [valueModel($Y_{X_4E_2}$)] = [MAU($Y_{X_4E_2}$), MAE($Y_{X_4E_2}$), SAU1($y^1{}_{X_4E_2}$), SAU2($y^2{}_{X_4E_2}$), …] = [0.24, 0.23, 0.6, 0.1, …]

Three *optional* inputs[1] to this process, of which *at least one must be present*, are:

5) A set **R** of eras, where each R is an n-tuple of *frames*. A frame r is a 2-tuple that comprises an epoch and a duration: <E, t>. The number of frames in an era R determines the size of n for that era's n-tuple.

[1] The author is grateful to N. Ricci for help in refining the descriptions of these inputs.

Example: Era #2 = R_2 = (<E_2, t_1>, <E_{16}, t_2>, <E_{11}, t_3>).
In this example, R_2 has 3 frames, and is $t_1 + t_2 + t_3$ time units in length.

Note: While eras are optional as inputs, they are required to conduct Multi-Era Analysis (as the name surely suggests). Eras can be generated for this purpose through various means, depending on the inputs to the process. See Section 5.4.5 for more on this point.

6) A set **P** of perturbations, where each *P* is defined as an operator on one or more of the following: 1) the design variable levels, 2) the performance model, 3) the context variables, and/or 4) the value model of the system. The Perturbation space is thus a set of *rules,* with each *P* pertaining to either a disturbance or an epoch shift.

   For example, a *P* could define a rule for:

   a. A transition of an epoch variable level to another epoch variable level (e.g., a probability value in a matrix for transitioning from element E_i to element E_j)
   b. The duration of a an epoch variable level (e.g., probability for or bound on the value of $t_k \in$ <E_i, t_k>, for some E_i in **E**.
   c. A short-duration change imposed on embedded performance model factors, causing reevaluation of *Y* on the cross product **X** x **E**.

7) A set **Δ** of change options, where each *Δ* in **Δ** represents a path enabler providing edges on a tradespace network. In the discrete change option case, these could include:
   a. a set of edges <X_i, X_j> comprising elements of **X** (i.e., an epoch-independent, discrete change path enabler on a tradespace network)
   b. a set of edges <EX_i, EX_j> comprising elements of the cross product **X** x **E** (i.e., an epoch-variable-dependent, discrete path enabler on a tradespace network)

   Each *Δ* has one or more associated change mechanisms, whose cost to execute defines the cost of traversing the associated path enabler's edge on the design tradespace network.

And one more optional input:

8) A set **T** of *tactics*, which are epoch-level transition strategies such as those found in VASC (e.g., maximize efficiency, survive), if era-level strategies will not be studied. For this thesis, "strategy" will refer to optimizations over an era, while "tactics" will refer to optimizations within a single epoch. ("Epoch-level strategies" may be occasionally used instead of "tactics" to help form consistency with the VASC literature.)

A representation of some of these sets of information in the larger context is depicted below in Figure 5-1. In the figure, "controllable" refers to modeled design

factors that are directly controllable for a given system, while "uncontrollable" refers to modeled factors that determine a system's attributes and value but are outside direct control. Uncontrollable factors may still be able to be influenced once a design is in further development and operation, but they are never under direct control.

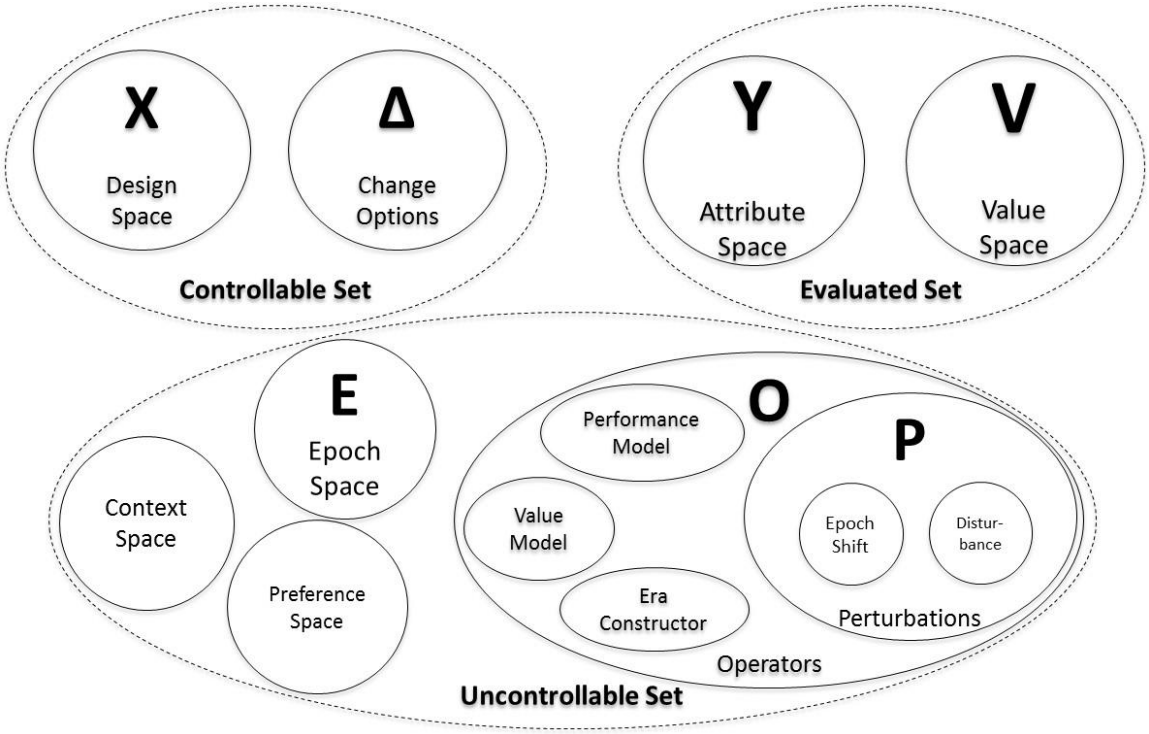


Figure 5-1. Sets of information potentially present as inputs to Multi-Era Analysis.

5.3 Components of and Considerations in Multi-Era Analysis

The form that the Multi-Era Analysis process takes depends on the inputs provided, the desired information output, and the time and computational resources available for the study. Some variants have been conducted in the past, though not with the same goals as outlined in this thesis. They are shown, along with the factors that can be included, in Table 5-1.

Table 5-1. Multi-Era Analysis (MERA) variants and their constituent components.

Variant	Design Changes Modeled		Epoch Information		Era-Level Information	
	From Exogenous Factors	From Change Options	Epoch Duration Rules	Epoch Transition Rules	Era-Level Strategies	Analysis of Path Dependency
Past variants:						
VASC (MEA)	No	Yes	No	No	No	No
VASC (Era-level)	No	Yes	No	Yes	No	Design- and Epoch-Level*
Fulcoly (2012)	No	Yes	Yes	Yes	No	Epoch-Level*
Richards (2009)	Yes	No	Yes	Yes	No	Design-Level
Presently:						
NGCS-like Case	No	Yes	Yes	Yes	Yes	Design- and Epoch-Level
Full Description	Yes	Yes	Yes	Yes	Yes	Design- and Epoch-Level

*Can be inferred from these publications, albeit not explicitly mentioned therein.

5.3.1 Consideration of Changes to Designs through Eras

Whether design changes are modeled is one of the defining characteristics in the variant of MERA being performed. Two sources for change can arise: the first is change imposed on the system from exogenous factors, while the latter is change that is executed by way of change options previously implemented in the system. For example, a naval ship that swaps out a modular weapon stack is taking advantage of a previously implemented change option (i.e., a modular bay and modular weapon stacks). On the other hand, the destruction of one of a ship's propellers due to ocean debris could be modeled as an exogenously imposed change to the system.

5.3.1.1 Changes Imposed by Exogenous Factors

Changes imposed (i.e., *design shifts*) from exogenous factors throughout an era are outside the scope of the present work, but have been covered in great detail by previous survivability-focused research, such as (Richards, Hastings, Rhodes, &

Weigel, 2007; Richards, 2009) and (Mekdeci, Ross, Rhodes, & Hastings, 2011; Mekdeci, 2013), and are formalized in concurrent work by (Ricci, 2014). They can be included as an input to Multi-Era Analysis in at least 2 ways: 1) by including them in the set **P** of perturbations as input to the MERA process; 2) including epoch variable levels that impose change on one or more of: a design *X*, or the value model of the system, potentially triggering new evaluations of **Y** or both **Y** and **V**. Such imposed changes can provide interesting path dependencies, as the survivability metrics of (Richards, 2009) demonstrate, but they will not be incorporated in the present work.

5.3.1.2 Changes through the Inclusion/Execution of Change Options

(Fitzgerald, 2012) considered changes to designs through eras by the execution of change options, with some simplifying assumptions. One such assumption was that a change mechanism could be executed any number of times. An additional assumption was that target designs were strictly confined to the set of nodes already in the tradespace network (i.e., only discrete change options were modeled, not continuous change options).

5.3.1.3 If Changes Are Not Modeled

If changes from neither options nor design shifts are modeled through an era, then the optional Prescriptive information portion of the study is no longer necessary (with the exception of determining the timing of system retirement/replacement); but the Descriptive goal of Multi-Era Analysis remains. Without either of these types of changes, though, little path dependency exists with respect to the trajectories of *X*s through eras. The only path dependency present is that of the developing epochs (and the resulting evaluations of the *X*s in those epochs) caused by transition and duration rules.

5.3.2 Consideration of Meta-Information on Epoch Durations and Transitions

Modeling epoch lengths and transitions is another defining characteristic of the variant of Multi-Era Analysis being performed, since better understanding the path dependencies of designs and epochs can provide great insight to present decision problems. (Fulcoly, 2012) describes the Epoch Syncopation Framework (ESF), in which epoch variable durations and transitions are modeled as Markov processes which can then be simulated to study the behavior of change options, choices of evolvability in a system, and strategies for the timing of future changes. Techniques such as these, or stochastically generated eras such as those found in (Richards, 2009) and (Fitzgerald, 2012), can be used to form the set of eras **R** as input to the MERA process.

When a set of eras **R** is not provided as input to the Multi-Era Analysis, eras must be constructed through the sequencing of frames: <*E, t*>, i.e., assigning a duration to each epoch that makes up the era, as well as determining which epoch comes next in the sequence. These rules are defined in **P**. It is often the case that knowledge exists about particular epoch variables that leads to reliable predictions on the duration of that epoch variable level anytime it occurs. It is also often the case that such knowledge can lead to reliable modeling of epoch transitions as well. If neither type of information is available or utilized, then the magnitude of Descriptive information produced by MERA is significantly lessened, since all of the path-dependent information on epochs remains un-modeled. In addition, if neither epoch transitions nor any other perturbations are modeled, and no changes are modeled to *Xs* through eras, then the MERA process devolves into Multi-Epoch Analysis.

5.3.2.1 Epoch Durations

The duration of an epoch *E* – and therefore the duration of frames containing that epoch – is dependent upon the potential durations of the constituent epoch variable levels that form the epoch. Whether the durations are modeled deterministically or probabilistically, external knowledge on epoch variable levels can aid in bounding the durations of epochs and their respective frames. (Fulcoly, 2012) and (Fitzgerald, 2012) both use memory-less processes for modeling epoch durations. In these cases, as with the present work, it is assumed that there exists no interaction between epoch variables with respect to the duration of any epoch variable levels, and it is further assumed that the constraints on each epoch's duration remain constant throughout an era.

Minimum durations from epoch meta-information

With these assumptions, it is the case that the minimum duration of an epoch (and therefore the minimum duration of a frame containing that epoch) can be strictly defined as the minimum of the duration of all of the constituent epoch variable levels, considering their previous levels and durations. This definition is reflected by the following formula:

$$\min\,[t_k \in \langle E, t_k \rangle] \;=\; \min\,[\text{duration}(E)] \;=\; \min\,[\text{min\_duration}(i)\;\; \forall\, i \in E],$$

where i is each constituent epoch variable level of epoch E, and
min_duration is a function that returns the minimum bound on i's duration,
given i's state in the previous frame(s).

For example, if the U.S. President were modeled as an epoch variable, with different candidates forming the epoch variable levels, then an epoch at the start of a Presidential term cannot last less than 4 years (deterministically, for present purposes). If the epoch had other variable levels that were known to possibly last less than 4 years – say, a political party controlling the House of Representatives – then any frames including that epoch would last at least 2 years (until the next election) before *potentially* transitioning to another epoch. The frame could last longer than 2 years, of course, which leads to the definition of the maximum duration of an epoch (and its associated frame).

Maximum durations from epoch meta-information

With the assumptions above, the maximum duration of an epoch (and its associated frames) can be defined in similar terms: it is the minimum of the maximum duration of the current epoch variable levels. The corresponding equation is then:

$$\max\ [\ t_k \in \langle E, t_k \rangle\ ] \ = \ \min\ [\text{max\_duration}(i)\ \forall\ i \in E],$$

where i is each constituent epoch variable level of epoch E, and
max_duration is a function that returns the maximum bound on i's duration,
given i's state in the previous frame(s).

To again use the example of the U.S. President: if the Presidential candidate were the only variable level in an epoch, then that epoch could not last longer than 8 years, since max_duration(PresidentInOffice) is equal to 8 years. If the House of Representatives was modeled as well, there would be no maximum time limit for a political party controlling the House. So max_duration(HouseControllingParty) would essentially be infinite (or could be modeled as a probabilistic duration, such as Poisson). However, in this case, the epoch (and associated frames) would still be limited to 8 years total, since the minimum of the two maximum durations is 8 years, and a new President must be elected after that time.

These examples, of course, ignore factors such as early succession of Presidents due to sicknesses, assassinations and impeachments, all of which are not that rare in Presidential history. Such exceptional factors can be examined by the analyst and modelers to determine relevance to the problem at hand, with perhaps associated probabilities being applied to accommodate these types of developments. In any case, epoch durations modeled from each of these types of additional knowledge –

minimums and maximums of epoch variable levels, whether deterministic or probabilistic[1] – can provide more realistic sets of eras to include in the study.

Constructing durations without epoch meta-information

It is likely the case that at least one epoch variable in any design study will have no minimum time bound modeled for any levels of that variable, meaning there will be no minimum duration for any frame in any era. In this case, if epoch transitions are not modeled in a way that takes into account durations, modeler judgment can be used to define a minimum timeframe of interest while keeping in mind the nature of the specific epoch variables modeled. To determine a maximum time bound on any frame only requires that one epoch variable level in that frame has a maximum time bound defined, which is likely the case for many combinations of epoch variables and their respective levels. However, as with the minimum bound, a maximum bound can be manually constructed in the absence of epoch meta-information, depending on how the epoch transitions are also implemented.

5.3.2.2 Epoch Transitions

It is often the case that, similar to durations of epochs, transitions between epochs can be reliably predicted through knowledge of epoch meta-information. (These transitions can be conceived as traversing edges on an epoch network, though such a description is not used in the present work). (Fulcoly, 2012) demonstrated modeling epoch transitions as Markov processes, and (Fitzgerald, 2012) modeled one-way transitions with the "Future" level of the Technology Level epoch variable, which could not return to "Present" once invoked.

To revisit the U.S. Presidential example: Assume that the President is modeled as an epoch variable, with different candidates forming each epoch variable level. Also assume that other epoch variables are modeled with variable levels that could last 1 to 3 years. Then an epoch at the start of a Presidential term could only transition to other epochs with that same President (with almost certain probability). After 4 years – i.e., after the Presidential term had been served – all other epochs would once again be available to transition to, since the "President" epoch variable could change at that time. (This example, as in the previous section, ignores factors such as early succession due to exceptional circumstances which turn out to be not that

[1] Probabilistic bounds on the durations of epoch variable levels are outside the scope of the current thesis, but provide a rich area for future research in formally describing the epoch and era spaces.

rare in Presidential history; but the point remains, as above, that transitions can be smartly modeled using this type of meta-information.)

As another example, if the Technology Readiness Level (TRL) of a particular technology were modeled as an epoch variable, the TRL level could be implemented as non-decreasing over an era. This type of relationship between epochs can be easily implemented to further help construct sets of eras that are most relevant to the questions at hand.

5.3.2.3 If Neither Durations nor Transitions Are Included

If no epoch meta-information exists or can be modeled, but *X*s are modeled to change through the era, then path dependence of designs can still be uncovered through the use of era simulation, even with all epochs having equally probable occurrence and lasting for an equally probable duration.

As discussed previously, if neither epoch transitions nor other perturbations are modeled, and *X*s are not modeled to change throughout an era, then no considerations apply for path dependencies of any kind. In this case, then neither the Descriptive portion of the MERA goal nor the Prescriptive portion applies, and the MERA process devolves into Multi-Epoch Analysis, with no need to consider long-term (i.e., era-level) paths and strategies.

5.3.3 Consideration of Era-Level Strategies, Trajectories, and Path Analysis

The stated goal of MERA is to uncover long-term strategies for value sustainment in the midst of developing path dependencies. One way that this goal is accomplished is through the use of descriptive metrics on the trajectories of *X*s through many eras. If information is provided for change options on designs, then the concept of *era-level strategies* emerges, the long-sighted version of the similar concept of strategies proposed in VASC. If such design options and strategies are provided, then descriptive metrics can be applied to the trajectories of *X*-strategy pairs throughout the era-space. If this is done, or if information is provided for epoch lengths/transitions, then path analysis can be performed through the use of descriptive metrics to identify and compare *X*s of interest. Each of these concepts will now be discussed in more detail.

5.3.3.1 Era-Level Notation

Strategies for change options in eras

Let the symbol Π denote the set of all modeled era-level strategies, where each Π is an *offline* planning algorithm that takes as input an era *R* and starting design *X*. The term "offline" with respect to algorithms comes from the fact that the algorithm is

given complete information of future events – in this case, an entire era – on which to make optimal plans. Offline planners do not adapt to changing circumstances, but rather optimize over the information available, which in this case is an entire era of epochs and durations. (It is possible to construct and use only "online" strategies, as will be discussed later in Section 5.4.3.2.) The function Π provides as output an action for each frame $<E,t>$ of the era. The action is defined as exactly one of the following: 1) no change, or 2) a list of change option Δ's to execute and each respective X_j to which the strategy changed in that frame. The list of change mechanisms reflects the (potentially multi-arc) transition paths, with each Δ and resulting X_j reflecting the changes and designs after each arc in the path.

Trajectories of designs through eras

Let the symbol $\boldsymbol{\psi}^{X,R,\Pi}$ denote a trajectory of an individual design alternative (subscript X) beginning in an era (subscript R), including any change mechanism execution required by the strategy (subscript Π, only required when options are included in the MERA process). In other words, $\boldsymbol{\psi}$ is an n-tuple (n being the number of frames in the era), where each element ψ_k in the set is a 3-tuple that represents a design-frame pair: i.e., $<X, E, t_k>$, where t_k reflects the duration of that frame. (Note that the pairing of X and E allow an alternate form of ψ_k: $<Y, t_k>$, where $Y = Y_{X,E}$ from the set $\mathbf{Y}$ of evaluated design-epoch pairs. This observation may be helpful to keep in mind, but will not be discussed further in the present work.)

When change options from the set $\boldsymbol{\Delta}$ are provided in the study, the outputs from a strategy Π above can be included in a hyphenated list preceding the target design X_j. These additions make the ψ_k an n-tuple, with n varying with the number of change mechanisms executed In that frame. For example, a ψ_k with one change could be $<X_i\text{-}\Delta_0\text{-}X_j,E, t_k>$, or with multiple changes in the same frame: $<X_i\text{-}\Delta_1\text{-}X_j\text{-}\Delta_2\text{-}X_l, E, t_k>$. Note that a more compact representation for multiple changes in the same frame could simply be to create a new change rule for each combination of existing change rules. In other words, if there were 3 total change rules defined for the study, then change rule 4 could be defined as an in-frame combination of change rules 1 and 2. This compact representation allows the ψ_k to always be a 5-tuple, since then only one rule needs to be referenced for any change. (It should be noted, however, that this method does remove the recording of intermediate designs that can prove to be important for some later types of analysis.)

As an example, take an illustrative trajectory $\boldsymbol{\psi}^{6,1,2}$. This notation would represent a trajectory through Era #1, which for this example, will comprise 4 unique epochs (#14, #25, #34, and #15) in 4 total frames of lengths 3, 2, 2, and 3 time units,

respectively. The trajectory starts in X_6. The strategy chosen to pair with X_6 is Π_2. The example trajectory is then represented by: {<6,14,3>, <6,25,2>, <6-5-8,34,2>, <8,15,3>}. As one can see, this design-strategy pair results in executing change mechanism #5 in the 3rd frame to transition to X_8, which the trajectory remains in for the last frame. The results stored in this way can easily be mapped to the other sets **Y**, **Δ**, and **Π** to evaluate any particular property of the designs, epochs, change mechanisms, strategies, and timings responsible for this trajectory.

Continuous-range change options present an additional challenge for recording these sequences. Such considerations are outside the scope of the present work; but for the purposes of the present discussion, it is assumed that any design *X'* created from a design *X* executing a continuous-range change mechanism is immediately evaluated by the performance model and then the value model. This newly evaluated point is immediately added as a new element *Y* to the set **Y** (i.e., for the evaluated performance attributes), and a new *V* to the set **V** (i.e., for the evaluated value vector). By so assuming, the previous definitions of **ψ** and **Π** can then be applied to continuous-range change options.

5.3.3.2 Strategies: Definition, Metrics, and Optional Components

As previously noted, the inclusion of change options in the MERA process requires a corresponding prescription for how the associated mechanisms will be executed. While the definition of these strategies may appear to be straightforward, the path-dependent aspects at the era-level add a unique challenge to strategy formulation due to the trade-offs between short-run and long-term results, as well as potential tradeoffs between stability and volatility. For this reason, strategy definition can be one of the most valuable activities of the MERA process, since it forces straightforward discussions about the value that stakeholders place on system attributes over the system lifecycle. As an example, simply stating "maximize efficiency" raises several questions: first, about whether (and how) to measure the preference of efficiency earlier in the lifecycle vs. efficiency later; and second, to what degree stability of the given metric is preferred over volatility. See Section 5.3.3.4, Path Analysis, for more discussion on this point.

A note on the paths that satisfy a minimally defined strategy

Similar to the strategy concept developed in VASC, an era-level strategy Π should return a single trajectory **ψ** for a given starting design *X*. However, it may be (and is in fact probably) the case that some set of trajectories **ψ** may satisfy a minimally defined strategy. When more than one **ψ** can be produced by a strategy Π, measures can be implemented to further reflect stakeholder preferences. Some example

measures can be seen in the meta-metrics presented in Section 5.3.3.4 below. If such measures are desired, they should be implemented in the strategy itself, since it is likely intractable to derive the set of all satisfactory **ψ**'s (which could lead to an examination of all possible paths of a design-strategy pair through the era). For the remainder of the present work, unless otherwise noted, an era-level strategy Π will return a unique n-tuple, not a set.

5.3.3.2.1 Strategy Definition: An optimization on metrics

Regardless of any other information provided, an era-level strategy Π must consider *all frames of an era*[1]. Example strategies and their objective functions (with n = number of frames in the era) are listed below.

1) Minimize lifecycle cost: $\min \sum_{k=1}^{n} opsCost(\psi_k) + changeCost(\psi_k)$

2) Maximize lifecycle efficiency: $\min \sum_{k=1}^{n} FPN(\psi_k)$

3) Survive: $\min \sum_{k=1}^{n} Infeasible(\psi_k)$
   Infeasible(ψ_k) := returns zero if X in ψ_k is feasible; otherwise, returns t_k

Each of these minimization functions must satisfy exactly one of the following conditions:

1) the X at ψ_{k+1} = X at ψ_k , hereafter denoted by: ${\psi^X}_{k+1} = {\psi^X}_k$
(i.e., no design change in the frame of ψ_k), or

2) the ${\psi^{X_j}}_{k+1}$ must be a target design X_j in the set **Δ** of change options for ${\psi^{X_i}}_{k+1}$
(i.e., a change mechanism is executed in the frame of ψ_{k+1} for X_i)

Notice that each example strategy has one or more corresponding Metrics Of Interest (MOI) on ψ_k by which it is defined. Of course metrics may be defined on any criteria of interest for the stakeholder, as long as the criterion is obtainable from any given **ψ**. (As was the case in VASC, the flexibility for creating metrics leads to an infinite number of strategies that could be created and compared within a MERA

[1] There may exist cases in which only certain portions of the future are of concern, and a strategy considers only those frames and no others. Generating such cases can prove to be an intriguing thought exercise but lie far outside the scope of the present work.

study.) Once a MOI is defined, it is applied to each ψ_k with a min/max function[1] to obtain an *era-level strategy* – the inclusion of every element of $\boldsymbol{\psi}$ is what differentiates an era-level strategy from an epoch-level one (such as those presented in VASC).

For example, the epoch-level Survive strategy discussed in VASC simply changes out of an infeasible design to a feasible one. But the era-level Survive strategy presented here may in fact *not* change out of an infeasible design if by doing so it would later make the trajectory infeasible for a longer period of time. In addition, the era-level Survive strategy as presented above *may change preemptively to another design* – i.e., in a frame in which the design is feasible – to prevent infeasibility in a future frame. Without any other constraint on the strategy, of course, it could even preemptively change to an infeasible design in the current frame in order to later prevent a longer duration of infeasibility. This point leads to additional discussion on the careful construction of era-level strategies by incorporating constraints on the execution of mechanisms and incorporating temporal aspects of measurement, which are now discussed. For an example of a fully crafted strategy incorporating constraints, see Section 5.4.3.3.

5.3.3.2.2 Optional Strategy Component: Constraints on Change Mechanism Execution

Several types of constraints may be placed on a strategy's ability to execute a given change option in a given epoch. The constraints can be formed from short- or long-term considerations, and they can be placed on monetary, temporal, or value-related metrics. For example, a strategy could be subject to the use of only options that require less than 6 months to execute (a short-term, temporal constraint). Alternatively, a strategy could be subject to a constraint that requires the lifecycle cost to be less than $5 billion dollars (a long-term, financial constraint). In general, short-term metric constraints on the attributes of change mechanism execution will be computationally more favorable than long-term, strategic constraints on design attributes. Long-term constraints have the potential to inadvertently over-constrain the space, leading to an examination of all possible paths through just one era, much less for many numbers of eras, which will be computationally intractable for any decently sized problem. Strategic constraints on execution are not modeled for the present work, although they can potentially contribute to the more concrete statements of stakeholder values (see Section 5.4.3.3 for more on this point).

---

[1] For the purposes of this thesis, it is assumed that era-level strategies are long-term optimizers, and as such can always be described in terms of minimization or maximization of metrics on frames.

5.3.3.2.3 Optional Strategy Component: Weighting of Metrics over Time

If a strategy Π is essentially defined by its application of metric(s) to all frames in an era, it should be obvious that a strategy could produce vastly different results if a discount factor is applied to monetary calculations (emulating financial NPV analysis). The merit of such discounting is not called into question here, although an epoch-based formulation can allow for a varying discount rate factor, helping address criticisms of constant discount rates. Certainly the use of discounting non-monetary expenses (e.g. ship crew size), benefits (e.g. utility attributes and SAU/MAU values) and other metrics (e.g., FPN) can be viewed with great skepticism. In these cases where discounting is not an obvious choice, consideration of the "now vs. later" preferences of the stakeholder prove to be a challenge, leading to the need for a more rigorous treatment of the path analysis and comparisons for many **ψ**'s through the same era. Such activity is begun below in Section 5.3.3.4.

5.3.3.3 Strategies: Implementation

The implementation of the above era-level strategies is challenging, even on a relatively small tradespace network, owing to the potentially large branching factor present in the midst of many change options on many designs. The nature of continuous-range change options also present their own rather obvious and daunting challenges when attempting to plan optimal trajectories through an era. Two efficient path planning algorithms are discussed here for these types of problems: the first, A*, is designed for optimal path planning over a discrete network of densely connected nodes; and the other, RRT*, is designed for optimal path planning over many-dimensional continuous spaces.

5.3.3.3.1 For Change Options On a Discrete Tradespace Network

Path planning over an era on a tradespace network follows the common practices of path-planning algorithms over networks. The input to the planning algorithm is a graph, made up of vertices and edges. Formally:

$$\mathbf{G} = \langle \mathbf{V}, \mathbf{E}, \mathbf{C}, \mathrm{S}, \mathrm{G} \rangle,$$

where $\mathbf{V}$ = a set of nodes,

$\mathbf{E}$ = a set of edges $\langle V, V \rangle$ between nodes,

$\mathbf{C}$ = cost function for each E,

and S and G are the start and end nodes, respectively.

In the case of the static tradespace network, each X in the tradespace forms a node V, and each Δ is a set of edges between nodes. The costs – whether temporal, monetary, or other – associated with each edge in Δ reflects the cost of executing the change mechanism that connects X_i to X_j.

A single design in a static tradespace can actually have its own single *design-tradespace network* – that is, its own set of nodes that it can reach over the existing edges in the tradespace network (see 6.1.4.2 for depictions of these in several case examples). In an era, many more nodes exist on the network, with each node V comprising a $<X,k>$ pair, where k is the frame number of the era. If only single-arc transitions were considered in the static design-tradespace network, then the nodes in the design-era network beyond k=1 (i.e. the first frame) will contain Xs not included in the original static design-tradespace network. In other words, it is possible that a X_i that transitions to X_j in the second frame of the era can reach additional Xs by the third frame of the era that the original X_i would not otherwise be able to reach. In this case where a single design has its own network through an era, it is defined by the problem description above.

On these tradespace networks, "blind" shortest path algorithms such as Breadth-First Search (BFS) or Incremental Depth-First (IDS) Search may be applied. Informed search algorithms, such as A* may also be applied. As noted previously, the performance of all of these optimal-path algorithms depends heavily on the branching factor of the design-tradespace network, as well as the number of change mechanisms that can be executed to reach the same designs. In a relatively large tradespace, even a small but consistent branching factor can prove to be very troublesome computationally.

In VASC, for example, the algorithm for creating the Full Accessibility Matrix was designed to find the shortest paths between all designs in a tradespace network in an epoch, given a set of available change options (i.e., edges). The time complexity of this calculation was assessed to be the $\#\text{mechanisms}^{\#\text{arcs}}$. In this particular algorithm's implementation, then, the d factor can be considered as the number of arcs (i.e., number of combined in-frame changes allowed to connect two designs in the tradespace network), with the branching factor b being the number of *reachable designs* (i.e., edges in the tradespace network) provided by each single mechanism to each single design. (For an alternate, more efficient approach, see Section 6.1.4.)

At the level of the era-strategy, of course, the branching factor b becomes the number of edges on the tradespace network provided by each mechanism *in each frame*, and the depth d is equal to n, the number of frames in the era. This causes concern for the complexity of the problem simply due to the size of the meta-

network resulting from the tracing of the tradespace over the era, much less the calculation of any path through the network. Three straightforward points should help alleviate dire concern. First, as discussed above, each design can have its own single design-tradespace network, meaning that the size of the era-level network under consideration is significantly reduced (likely by orders of magnitude). Second, the depth of such networks are limited to the number of frames in the era (see Section 5.4.5 for more discussion on this point). Third, real-life problems are rarely worst-case scenarios, as is shown below in Figure 5-2 from (Russell & Norvig, 2009), where the informed search algorithm A* is run on 12 variants of a network shortest path problem, each problem with a branching factor of 3 and a depth from 2 to 24 levels. Two different heuristics are implemented, with vastly different outcomes in the number of nodes considered, demonstrating the importance of the heuristic-construction process. The blue line on the left represents the performance of Incremental Depth Search (IDS), a combination of the (uninformed) Breadth-First and Depth-First search algorithms.

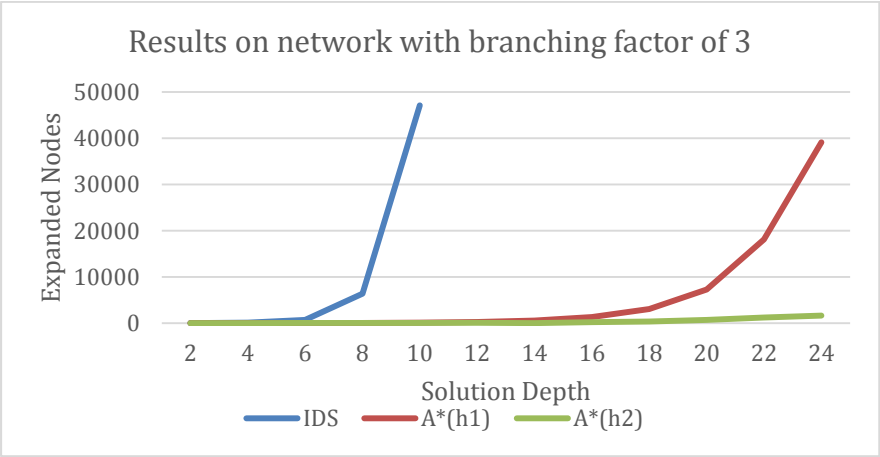


Figure 5-2. Results of A* search on problem with small branching factor. (Russell & Norvig, 2009)**, Sect. 3.6, Fig. 3.29.**

As one can see from Figure 5-2, A* algorithm is able to quickly find the shortest path in such problems by evaluating only a small subset of the nodes in the network. In other words, the algorithm's *effective* branching factor b^* is far less than the actual branching factor b of the network. It does this by using a heuristic to estimate the cost to the goal for any node whose children the algorithm is considering expanding (i.e., a node on the queue). If the network's path cost to a node on the queue is added to the heuristic cost on that node, and the summed amount is less than the shortest estimated path to the goal found so far, then A* will expand the node's children, adding them to the queue. If the heuristic is *admissible* – that is, it is optimistic on the distance to the goal – then A* is guaranteed to return an optimal path, with its

computational complexity then being determined by the complexity of the heuristic evaluation (as well as the quality of heuristic, as Figure 5-2 shows). The pseudocode for the algorithm can be found in the Appendix, from ("A* Algorithm," 2014).

The spatial complexity of A* is usually of concern in practical applications, since it considers exponentially more nodes with increasing depth. However, the longest conceivable eras as defined here will only consist of around 10-20 epochs, and most of the paths are relatively obstacle-free; both of these factors help reduce the spatial complexity that is normally a barrier to effective implementation of A* in real-world problems.

5.3.3.3.2 Example Heuristic for an A*-based Strategy Implementation

Assume that an era-level strategy Π desires to minimize Lifecycle Cost through an era with 20 frames, and the starting design X has its own design-tradespace network with a branching factor of 5 (each design in each frame can reach 5 unique designs in the next frame through the execution of one or more change mechanisms). Clearly by the end of the era, it is possible that an entire tradespace of thousands of designs will be reachable by executing change mechanisms. However, to limit the number of nodes and paths considered in the network (and therefore limiting the *effective* branching factor), a reasonable heuristic can be constructed for estimating the Lifecycle Cost of any node_j in frame $k+1$, given a starting node_i in frame ψ_k:

$$\text{ChangeCostFromTo}(\text{node}_i \text{ in } \psi_{k+1}, \text{node}_j \text{ in } \psi_{k+1}) \ + \ \text{OpsCost}(\text{node}_j)*t_{k+1}$$

This type of constructed heuristic can incorporate any metric, not simply OpsCost, and will always be admissible for strategies concerned with the durations of a metric. Construction of this type of heuristic does assume a metric that is concerned with duration; for example, this heuristic would not work well for binary metrics (such as "infeasible or feasible" with no regard for the amount of time a design is infeasible). In these cases (and even for the example heuristic shown above), meta-information on epoch transitions can be incorporated (though probabilistic information may lead to *inadmissible* heuristics, meaning the path may not be optimal). A demonstration implementation of A* in creating an era-level Survive strategy can be found in Section 6.2.3, with corresponding Matlab code in the Appendix.

5.3.3.3.3 For Continuous-Range Change Options

Continuous-range change options present challenges unique to those present in the discrete case. Any era-level strategy Π then needs to be able to quickly generate

paths to the goal regions while avoiding (physically and valuably) infeasible regions, which is an ideal problem for RRT-based algorithms to solve.

In implementing RRT-based algorithms in a path-planning problem for a <*X*, *∏*> pair through multiple frames in an era, several issue arise. First we consider the myopic, tactic-based questions regarding frame to frame transitions. First, when a <*X*, *∏*> pair transitions from frame *k* to the next frame *k+1* in the era, what is the pair's goal with respect to path planning; and second, what are its obstacles to that goal? An initial answer to the first question is that its goal region is simply the next frame (*k*+1). This answer implies that the cost of the path to the goal, then, is the Metric of Interest (MOI) evaluated on *X* in frame *k+1*. The obstacles, to answer the latter question, are the physically infeasible configurations of continuous-range *Δ*s available to *X*, and could also include the value-infeasible regions of **V** evaluated from physically feasible configurations of *EX*' (where *E* is that of the next frame *k+1*). The sample space is defined by all possible configurations of continuous-range *Δ*s available to *X*, with rejection of whatever sample points are found to be in the obstacle space. (Note that in this case, the cost of the path *lies in a different space than the goal region and samples*.) Because all possible configurations of continuous range *Δ*s lead to *EX*' that are neither evaluated with respect to performance nor evaluated with respect to values (i.e. no associated Y' or *V*' yet exist), tradespace approximation methods may be useful in constructing efficient predictions (of attributes and/or values) for any configuration *EX*' in order to approximately evaluate the sample (a new *X*') in the next frame's *E* and then the **V** boundary space as well. If a new path is found to the goal region through another set of changes, then the strategy's MOI can be evaluated on the new *X*'-frame pair, and the tree will now contain more than one path to the goal (which path is optimal depends directly on the strategy's MOI). In this way the algorithm can proceed, continually attempting to find additional paths (and hopefully more optimal values of the strategy's metric) as it transitions to the next frame.

Let us extend these concepts arising in the myopic tactics to the era-level strategy. In this longer-term strategy, the algorithm cannot proceed frame-by-frame, optimizing its path along the way; rather, the full space of possible transitions is explored over the entire era. In this case, the space of boundaries still exists (just as above) in the physically infeasible configurations of continuous-range *Δ*s available to *X*, and could also include the value-infeasible regions of **V** evaluated from physically feasible configurations of *EX*' ∀ *E*s in the era. (Note that whether to include value-infeasible regions is a modeling choice on the era-level, and may eliminate optimal paths to a strategy that does not specify value-feasibility as a

constraint.) The goal region, as in the discrete case, is now considered to be the last frame of the era. The sample space is then the space of combinations of frame and physically feasible configurations of the continuous-range Δs available to X. One modification then remains to be made to the algorithm in order to allow it to explore feasibly: each node on the random tree can only be extended to the same frame k or the next frame $k+1$ of the era. In addition, a branch and bound version of the RRT-based algorithm can be employed, which, once a feasible path is found to the goal region, simply prunes paths and new samples based on a comparison of their optimality with respect to the known feasible path to the goal region. One of the outcomes of applying an RRT-based algorithm to discrete frames in an era is that the branch lengths no longer are the same from each perspective; that is, the "length" of branches means something different in the MOI space as opposed to the boundary and goal spaces.

The first algorithm shown here is the classic RRT, based on the description in (LaValle & Kuffner Jr, 2000). It is as follows:

```
function BUILD_RRT(q_init)
      inputs: q_init, an initial node
      T.init(q_init)
      for k=0 to K do
            q_rand <- RANDOM_CONFIG()
            EXTEND(T, q_rand)
      return T
Algorithm 1: BUILD_RRT
(LaValle & Kuffner Jr, 2000)
```

The RRT algorithm begins with a tree consisting of one node; that is, the starting node. It then samples the free configuration space to generate a new candidate node, which it attempts to connect to the tree. This attempt is the EXTEND function, which is described below in Algorithm 2.

```
function EXTEND(T, q)
      q_near <- NEAREST_NEIGHBOR(q, T);
      if NEW_CONFIG(q, q_near, q_new) then
            T.add_vertex(q_new)
            T.add_edge(qnear, q_new)
            if q_new = q then
                  Return Reached
            Else
                  Return Advanced
      Return Trapped
Algorithm 2: EXTEND
(LaValle & Kuffner Jr, 2000)
```

The idea is quite simple, really: for any given randomly sampled point, attempt to extend the tree to that point by some fixed amount of distance. The fixed amount can vary depending on the environment and algorithm. Intuitively, one might expect that the optimal value for this distance would depend on factors such as: the size and shape of infeasible regions, the size of the bounding box, and the particular variant of the algorithm (RRT vs RRT* vs RRT* branch and bound, etc), as well as the desired performance (is an optimal path more important than the first feasible path?).

On the challenges in determining the nearest neighbor

These formulations require the Nearest Neighbor algorithm to determine which node in the tree to extend by the newly sampled point *X*'. This calculation is convenient for multi-dimensional Euclidean spaces, where the path cost metric and branch lengths in each dimension are measured in the same space and easily comparable. However, in a system with multiple continuous-range Δs, the calculation of the nearest neighbor is by no means straightforward. How does a change in one dimension compare to changes in the other dimensions, especially if such a change does not affect the metric used in the strategy? One continuous mechanism (e.g., increase orbit altitude) might have a range 10x the range of another mechanism (decrease communication intervals). It is possible that the percentage change of each individual range could be used, or perhaps some variant of the metric from the strategy instead, which means determining the nearest neighbor across the different spaces of samples and evaluated metrics. For a strategy that is strictly cost-based, the cost of each change could potentially be the determining factor. If costs other than monetary are included -- temporal, or other – then additional complications arise from identifying stakeholders' preferences on money and time. It should be obvious that each of these choices for determining the nearest neighbor could produce significantly different random trees, and therefore significantly different trajectories of designs through an era. The creation of the nearest neighbor function, then, turns out to depend directly on the strategy being implemented, with potential modifications required for any alternate strategies.

Combining Discrete and Continuous Change Options

At this point in the discussion, the question naturally arises of how to solve path planning when considering both discrete and continuous change options. Approaches to describing or modeling this type of problem, much less suggested approaches to solving it, lie outside the scope of this thesis. Further research in this area is suggested in Section 7.3.3.

5.3.3.4 Path Analysis, Era Metrics, and Meta-Metrics

The challenges involved with measuring and comparing trajectories of a *X* or a <*X*, *Π*> through an era primarily stem from two unique issues based on stakeholder preferences:

1) Whether a stakeholder prefers benefits and/or costs sooner rather than later;

2) Whether a stakeholder prefers stability (short-term and/or long-term) in the Metric of Interest (MOI) calculated across an era.

Ideally, any era-level metrics conceived should adequately capture the information necessary for any given stakeholder, regardless of their varying degrees of preference on these issues. Previously in era-level studies, (Richards, 2009) captured certain aspects of these concerns with survivability metrics such as median time-weighted utility loss and threshold availability, while (Fulcoly, 2012) and (Fitzgerald, 2012) simply used time-weighted averages to report metrics over time.

To demonstrate the multiple aspects of these temporal concerns, four example paths of a Metric of Interest (MOI) evaluated on a **ψ** through an illustrative 4-frame era are shown below in Figure 5-3, with the MOI evaluated on each of the design-frame pairs for all frames in each era.

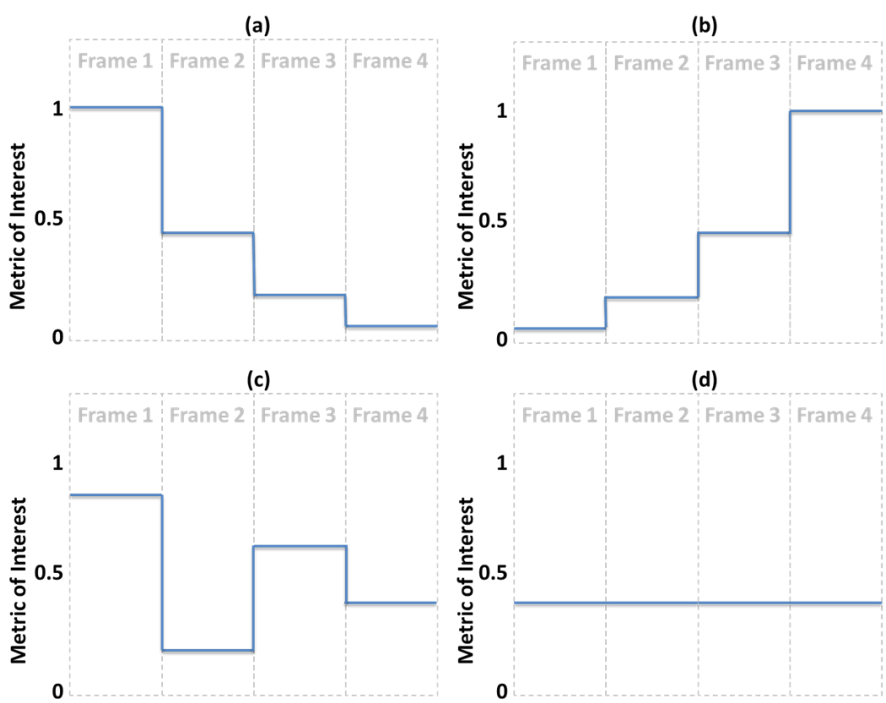


Figure 5-3. Four example paths of the development of a MOI over an era, exemplifying the time-related variations in values.

It is assumed that a given stakeholder may or may not prefer certain of these paths over others, and the preference may change depending on what MOI the path is based on (e.g. Utility, Ops Cost, FPN). Of course many function fitting and function approximation methods exist to describe these paths, such as those used in financial analysis and statistical regression. However, the non-uniform-length piecewise constant functions encountered in the present work pose a challenge in using established methods from other domains to analyze the discrete sections along an era, since there may not be any variance between two sample points. In addition, since Multi-Era Analysis is concerned with very large numbers of paths of potentially many MOIs, it is desirable to avoid optimization or iterative algorithms at the lower levels wherever possible. This reasoning leads to the construction of simple, special-purpose *meta-metrics* on the paths that a MOI might take through an era. It is proposed here that an era-level metric then be defined as follows:

> An *era-level metric* of a design or design-strategy pair is a set comprising a Metric Of Interest (MOI) in all frames and some subset of meta-metrics on that MOI in all frames.

With these considerations in mind, six separate meta-metrics for Metrics Of Interest (MOIs) are proposed below for present purposes that, when combined, attempt to efficiently capture the aspects above of measuring a path through an era comprising n frames. This list is not meant to be exhaustive of possible meta-metrics for era analysis, but rather to demonstrate the proof of concept for future research to build upon. For example applications of each meta-metric to the different metric paths shown in Figure 5-3, see Table 5-2 later in this section.

1. EXPEDIENCE

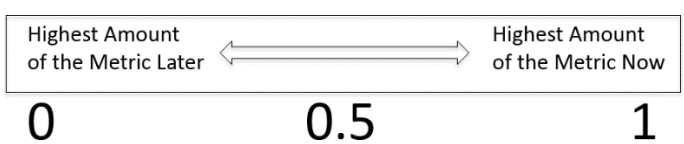
Purpose:

The first meta-metric, Expedience, captures the percentage of the MOI's total value that is delivered in the first half of the era. It is a first-order approximation aimed to address consideration #1 (i.e., whether stakeholders prefer the MOI sooner or later).

Equation:

$$\frac{\sum_{k=1}^{\frac{n}{2}} t_k \mathrm{MOI}(\psi_k)}{\sum_{k=1}^{n} t_k \mathrm{MOI}(\psi_k)}$$

Range:



2. VARIABILITY

Purpose:

The second meta-metric, Variability, aims to address consideration #2 from the previous discussion. It does not distinguish between one larger shift in the MOI value and many smaller changes in the MOI value, but it can provide additional information when combined with other meta-metrics such as Range and Greatest Change. It does not require calculation of the mean or time-weighted average.

Equation:

$$\sum_{k=2}^{n} \mathrm{abs}[\mathrm{MOI}(\psi_k) - \mathrm{MOI}(\psi_{k-1})]$$

Range:

The range of the Variability of a MOI is dependent on the global bounds of the MOI as well as the number of frames in the era in which it is measured (the only meta-metric described here that depends on the latter). The range of the Variability can exceed the global range of the MOI, since the differences between each frame are summed in calculating the Variability.

3. TIME-WEIGHTED AVERAGE

Purpose:

The third meta-metric, Time-Weighted Average, simply captures the average value of the MOI throughout the era. It is the only meta-metric not related to the 2 temporal considerations listed above, but has been applied in previous work by (Fulcoly, 2012) and (Fitzgerald, 2012). In the absence of stakeholder preferences on the temporal delivery of a metric, this meta-metric is still useful for comparing different paths.

Equation:

$$\frac{\sum_{k=1}^{n} t_k \mathrm{MOI}(\psi_k)}{\sum_{k=1}^{n} t_k}$$

Range:

The range of the Time-Weighted Average directly depends on the global bounds of the MOI being measured.

4. GREATEST INSTANT CHANGE

Purpose:

The Greatest Instant Change can be broken into Greatest Instant Rise and Greatest Instant Fall, which aim to address a short-term aspect of consideration #2 above. These meta-metrics can be useful in locating individual perturbations that are particularly beneficial or troubling to the metrics being applied over the era.

Equations:

Greatest Instant Fall: $\min\,[\mathrm{MOI}(\psi_k) - \mathrm{MOI}(\psi_{k-1})\ \forall\, k,\ \ k \in \{2,3,\ldots,n\}]$

Greatest Instant Rise: $\max\,[\mathrm{MOI}(\psi_k) - \mathrm{MOI}(\psi_{k-1})\ \forall\, k,\ \ k \in \{2,3,\ldots,n\}]$

Range:

The range of the Greatest Instant Change directly depends on the global bounds of the MOI being measured.

5. RANGE

Purpose:

The Range of a MOI throughout an era aims to capture a long-term aspect of consideration #2 above. It measures the difference in maximum and minimum values of a MOI over an era, regardless of when they happen and regardless of the order in which they occur.

Equation:

$$\max[\mathrm{MOI}(\psi_k)\ \forall\, k] - \min[\mathrm{MOI}(\psi_k)\ \forall\, k],\quad k \in \{1,2,\ldots,n\}$$

Range:

The range of the Range meta-metric directly depends on the global bounds of the MOI.

Table 5-2. The six meta-metrics evaluated for each of the 4 example MOI paths shown in Figure 5-4.

META-METRIC	FIGURE 5-3 (a)	FIGURE 5-3 (b)	FIGURE 5-3 (c)	FIGURE 5-3 (d)
EXPEDIENCE	0.8	0.2	0.45	0.5
VARIABILITY	0.95	0.95	1.2	0
AVERAGE	0.4	0.4	0.4	0.4
GREATEST RISE	0	0.55	0.4	0
GREATEST FALL	0.55	0	0.6	0
RANGE	0.95	0.95	0.6	0

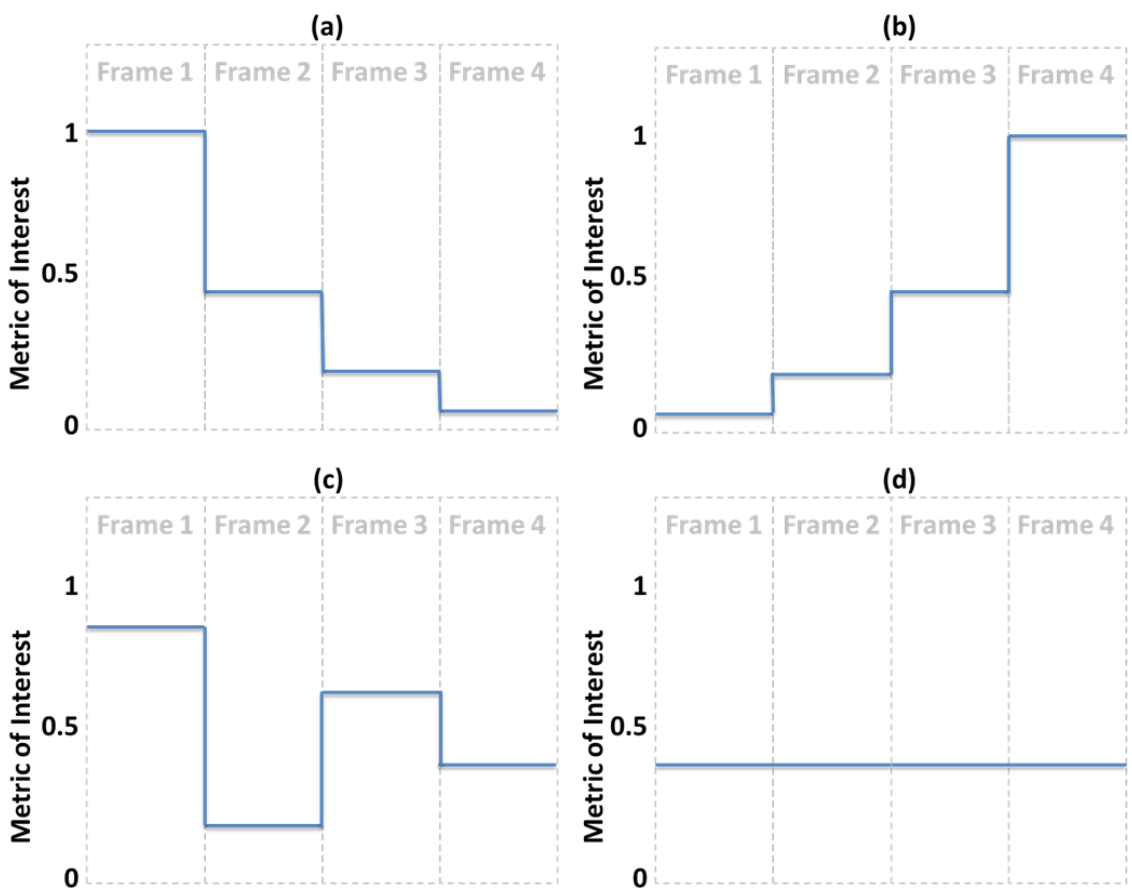


Figure 5-4. The four example paths of Figure 5-3, depicting developments of a MOI over time.

While basic in nature, these meta-metrics are efficient to compute and can help indicate some basic properties of metric paths[1] encountered during Multi-Era Analysis. They are expected to be of use in the construction of era-level strategies, as well as the detection of relevant path dependencies as part of era creation. Some existing statistical and financial time-series measurements were examined prior to the creation of these meta-metrics, but it is likely that promising measurements went unconsidered. Many such existing measurements require a number of assumptions to be made, including: 1) continuous functions with non-zero variance, 2) large number of sample points, 3) underlying static probability distributions, 4) ratio measurements on universally-ranged scales, 5) linear relationships, 6) large amounts of time available to analyze one time series, and/or 7) goodness-of-fit metrics needed for a single application. Some of these assumptions are appropriate for limited application of some system-level metrics across eras, but in general, system-level metrics encountered in methods such as RSC and RMACS are not strictly analogous to financial measures, nor are the artificial datasets of future eras analogous to the empirical datasets that are assumed for many financial and statistical metrics. The meta-metrics presented here are not meant as a comprehensive list, but rather a springboard for further development of thought in this area. Much opportunity exists for this area to be expanded and refined in future research, including potential modification of established time-series analysis methods to be made appropriate for inclusion in a design study.

5.4 Activities in Multi-Era Analysis

The process of Multi-Era Analysis can comprise up to 7 activities, each varying in length, type of activity, and output. All of the MERA activities are shown below in Table 5-3. The first and second activities help to scope the study by defining the ultimate goal and by limiting the number of designs to include in the consideration and analysis. The third and fourth activities are shaded to reflect their optional nature, as they can only be performed when change options are included as inputs to the MERA process. The fifth activity helps scope the study by only including the clips and eras that will initially be of interest. The sixth activity simulates the eras of interest and generates evaluation information on designs, design-strategy pairs, trajectories, and era-level metrics on those trajectories. The seventh and final

[1] Care must be taken when interpreting meta-metrics such as Expedience of MAU and MAE values across eras, as the zero reference point of these metrics may change between epochs. As later discussed, meta-metrics such as Greatest Instant Change can still be valuable in locating interesting path dependencies for these metrics in addition to other metrics such as operations cost, FPN, etc.

activity analyzes the results of Activity 6 with respect to the goals outlined in Activity 1.

Most of these activities depend on each respective previous output, but each activity is designed to be flexible with regard to the amount of information provided as input. A quick first pass through several of the steps may help orient the analyst in the general feel of the problem, highlighting areas of interest and inspiring additional questions. This first pass may naturally lead to feedback provided to the first activity, refining and modifying the inputs for the second pass of each activity. The amount of detail included in each iteration is up to the practitioner, but the minimum output expected from each activity is shown in Table 5-3. As an example of a quick first pass through the process, an analyst may choose only one or two Metrics of Interest, skip Activities 3 and 4, and only create and evaluate a few designs in all 3-frame clips. Preliminary analysis of the results may reveal potential areas of concern or possible improvement, and the first process can be revisited, with a potentially better understanding of the problem on a high level. Change options may be introduced on subsequent iterations, providing the means to create strategies and mitigate (or take advantage of) the uncertainties that present themselves in the eras being modeled.

Table 5-3. The activities of the MERA process, with inputs, description, and outputs. Activities 3 and 4 are shaded to indicate optional nature, only included when change options are included as inputs to the MERA process.

Activity Name	Possible Inputs	Activity Description	Outputs
1. Identify MERA Variant	All inputs to the MERA process.	Determine which path dependencies can be identified (whether epoch- or change-related or both) and which are of primary concern.	The overall goal for the process: which path dependencies are of concern? How will they be measured?
2. Select Designs and Metrics of Interest	A set X of designs, a set of metrics, results from previous Processes, and the goal from Activity 1.	Using results from previous processes as well as the goal from Activity 1, identify promising designs and the metrics of most concern. Create era-level metrics if desired.	*Required*: A subset of designs, and a subset of Metrics of Interest (MOIs). *Optional*: A set of era-level metrics.
3. Create Era-Level Strategies	A set **T** of tactics, and outputs from Activities 1 and 2.	Using the overall MERA goal as a guide, using MOIs and era-level metrics as enablers, create long-term strategies that reflect the desired tradeoffs on MOIs over time.	At least one era-level strategy description and implementation.
4. Identify Design-Strategy Pairs	A set **T** of tactics, and the outputs of Activities 1 to 3.	Using the goals outlined in 1, pair the designs identified with tactics and strategies created.	A set of design-strategy pairs to be evaluated in the eras.
5. Create Eras (required when a set **R** of eras are not input)	A set **E** of epochs, a set **P** of perturbation rules, and/or an era constructor.	Using Human-In-The-Loop, Breadth-First Search, and Random Sampling, sequence a large number of combinations of epochs and durations.	A set of eras in which all designs (and optionally strategies) of interest will be evaluated.
6. Design-Strategy-Era Evaluation	The outputs of Activities 2 to 5.	Simulate the designs (and optionally strategies) of interest through all of the eras created in Activity 5.	A set of evaluated trajectories for all designs of interest in all eras created in Activity 5.
7. Results Analysis	The outputs of Activities 1, 2, and 6.	Data-mining, correlation, interactive visualization, etc.	Potentially infinite.

5.4.1 Activity 1: Identify Desired Variant

The variant of MERA chosen, as well as the overall goals for the design study, will guide the rest of the process. Specifically, the analyst will need to consider which criteria are most relevant to learn about the path-dependent nature of the uncertainty space and/or change options in designs. Is a design's cost of most concern? Utility? Utility-cost efficiency? Previous metrics identified and favored in Single-Epoch Analyses and Multi-Epoch Analysis can aid in determining the aspects of the system with which the entire design study has been concerned up to this point. In addition, analysis of the inputs to the process can help, since lack of some of the optional inputs results in an inability to conduct certain variants of MERA. (For example, if change options are not provided as input, then the MERA variant cannot be concerned with path dependency of design-strategy pairs.) The amount and usefulness of information output at the end of the MERA process, as well as many assumptions made throughout the process, directly depend on the goals outlined in this first activity.

5.4.2 Activity 2: Select designs and epoch-level metrics of interest

Selecting designs of interest from a trade study before proceeding to Multi-Era Analysis is recommended, due to the large amount of computation and data required. However, as will be demonstrated in Chapter 6, simulating large portions of the tradespace may be informative as well. Depending on the implementation of the era constructor, era simulations may be relatively quick to run, since epoch-level evaluations are already computed. The primary driver of the amount of computation and resulting data is the number of eras simulated (see Section 5.4.5 for detailed treatment of this topic), but tradeoffs will exist between the number of eras created, the number of designs included in the study, the number of era-level metrics (MOIs and meta-metrics) created, and the number of strategies created and paired with the designs of interest. Once Activity 1 has been completed, the goal of the study can be a helpful guide in selecting metrics and designs using the criteria most relevant to the study's stated purpose. In addition, combining the epoch-level metrics of interest with one or more of the meta-metrics discussed in Section 5.3.3.4 will result in era-level metrics, which can then be incorporated to following activities.

5.4.3 Activity 3: Create era-level strategies

As previously discussed, if a set of change options are not included in the inputs to the MERA process, then era-level strategies are not needed to conduct any of the activities. In this case, the practitioner may proceed to Activity 5. It could also be

the case that a set of change options are provided as input to the study, and a set **T** of VASC tactics are provided as well. In this case, Activity 3 is optional, and the practitioner may proceed to Activity 4 using only VASC tactics; but era-level strategy creation remains highly recommended for the reasons outlined below. If a set of change options are provided as input to the study, and VASC tactics are not also included in the study, then this activity is required.

5.4.3.1 Creation of Offline Strategies

As discussed in Section 5.3.3.2, an offline strategy is essentially an optimization function on one or more metrics evaluated on every frame of an era. The metrics used in any strategy should reflect the aspects of a design that are relevant to the overall design study goals, which should be clearly laid out as part of Activity 1. Even still, careful consideration must be given to *all* of the aspects of a design that are deemed important, as is obvious from Section 5.3.3.2's discussion of the ramifications of the simply-crafted Survive strategy. Tie-breaking measures, such as meta-metrics (or other path analysis) of any kind, can further complicate understanding already-opaque outcomes of applying the strategy. For this reason, several "test" designs and "test" eras may be selected to help verify that a strategy's construction accurately reflects the priorities of the design study.

5.4.3.2 Creation of Online Strategies

One seeming alternative to the challenges of constructing offline strategies is to instead construct a strategy from *online* algorithms – that is, algorithms that do not operate on complete knowledge of the entire future. The VASC strategies were essentially online planners, with no consideration of future states. These could be augmented by one- or two-step look-ahead algorithms. More complicated algorithms could be used as well, such as those used in strategic game planning like chess or backgammon. If such an algorithm were given a starting design and large numbers of eras, it could dynamically learn to perform better with respect to the goals and constraints defined for it. The measurement of "better", though – the goals and constraints – lands such a planner squarely back in the same difficulties as the offline planner. That is, the goals and constraints of such a planner must be well-defined, since "winning" in complex systems is nowhere near as well-defined as "winning" in a strategic, zero-sum game. See Section 7.3.8 for more in-depth discussion of an online planner for Prescriptive Information after the completion of the Descriptive portion of the MERA process.

5.4.3.3 Era-Level Strategies as a Grounding of the Value Proposition

For the reasons discussed above, the creation of era-level strategies might be considered one of the more difficult portions of the MERA process. To be precise, it is the creation of era-level strategies *that accurately reflect stakeholder values* that might be considered one of the more difficult portions of the MERA process. However, this activity can also prove to be one of the most valuable activities in a design study, as it forces the analysts and decision makers to question and fully consider what they find most valuable in any system that is eventually developed. For example, the objective function chosen indicates by itself much about the way that a complex problem is being approached – if a stakeholder is primarily concerned with the efficiency of a system, then the objective function should explicitly consider the FPN in each frame. If the stakeholder is primarily concerned with satisfaction, on the other hand, then the objective function should reflect the utility in each frame. The constraints generated in this activity also provide additional information on how the problem is conceived – in other words, the constraints provide additional insight into the priorities of the system design process. If there is an existing budget to guide expenses, then they can be incorporated into the constraints on the objective function. A simple example of a fully-defined strategy is shown below.

$$\min \quad \sum_{k=1}^{n} Infeasible(\psi_k)$$

s.t. 1) ${\psi^X}_k = {\psi^X}_{k-1}$ or $< {\psi^{X_i}}_k, {\psi^{X_j}}_k > \in \mathbf{\Delta}$,

2) GreatestInstantRise[OpsCost($\boldsymbol{\psi}$)] < \$10 million/yr

3) 0.3 < Expedience[OpsCost($\boldsymbol{\psi}$)] < 0.7

4) Range[ChangeCost($\boldsymbol{\psi}$)] < \$30 million

This example strategy has several components that reflect the values present in this (hypothetical) system design problem. The first thing to note is that the objective function is with regard to the feasibility of the system; in other words, the stakeholder conceives of the problem in such a way as to be primarily concerned with the system's feasibility. A second note of interest is that the operations and change costs are explicitly limited by a budget. There are many ways to handle such constraints that may or may not nullify the space of solution paths: one way is to define preferences on the order in which they can be relaxed, while another is to simply have the strategy report a failure if it cannot find a path. The results of testing these techniques on sample eras will potentially give analysts and stakeholders a clearer picture of what is expected of the system, as well as indicate what is possible to achieve given the modeled context and preferences, and performance and value

models. For these reasons, this activity potentially requires multiple rounds of feedback and iteration, through which clarity in communication and a common understanding of realistic expectations can be achieved.

It should be clear from the many aspects considered in this simple example that the creation of an era-level strategy is intertwined with stakeholder values to such a degree that *such strategies can be considered concrete, context-centric definitions of the Value Proposition formed in Process 1 of the RSC-based Method for Affordable Concept Selection*.

5.4.4 Activity 4: Identify design-strategy pairs of interest

This activity is highly dependent on the output of Activities #1 through #3. Specifically, the process could consider one or more designs with no change options available, designs of interest matched with epoch-level tactics, or designs of interest matched with one or more variants of an era-level strategy. In the first case, the results would primarily be informative with respect to the path-dependent nature of the developing epochs. In the second case, the results would inform with respect to the path-dependent nature of designs as they change in the developing epochs. In the third case, the results would inform with respect to opportunities for further option creation and inclusion.

5.4.5 Activity 5: Create eras

With the completion of Activities 1 and 2, the desired variant of the MERA process has been identified, as well as the designs of interest in the trade study. In addition, the metrics of interest have been identified for the designs throughout the eras. For the process to proceed, the required information must be available to simulate designs' behaviors through a large number of eras. As discussed in Section 5.2, if change options are not included in the study, then either a set **R** of eras must be provided as an input, or a set of perturbations **P** must be provided as input to enable era creation. If change options are included in the study, then neither of these things is strictly necessary, since the set **P** can be created from uniform random distributions on **E** and durations, respectively, in order to study the path-dependence of changing designs through eras. If **R** is not provided as an input to the process, then it must be created through the implementation of **P** in some form.

5.4.5.1 The Purpose of Era Creation

The size of the era-space is infinite, due to the inclusion of various-length time durations for each epoch; but even if the durations are fixed for all epochs, there potentially remains a space of practically infinite discrete size for any reasonably-modeled problem. At least three issues arise in any discussion of probabilistically

characterizing such a space through sampling. First, it is not clear that the assumptions of independence and identical distribution hold for sampling portions of eras comprising (potentially) path-dependent design changes through a (potentially) path-dependent epoch space; for this reason, depending on implementation and sampling method, eras may be considered a categorical distribution in which each era is a potential outcome. Second, it is intuitive that most system-level metrics over many possible system lifetimes will likely have a large variance, meaning that any number of attainable samples might result in still very wide confidence intervals on any statistics, providing relatively little return on much computational effort. Third, it is not clear which statistics on the samples are helpful, as designing for an "average" future is not recommended (nor clear in meaning, since only one future ever happens), and neither is designing for "worst-case" futures (even if these could be guaranteed to be included in the samples). (Abbass et al., 2008) recommend refraining from averaging outcomes of scenario planning, since each scenario "represents a plausible future rather than noise". But as previously mentioned in the discussion of the purpose of exploratory models, the discretized models of the future inform decision makers about the nature of and approach to a complex decision problem (Bankes, 1993). It is understood that exploration of the space of possibilities will almost never lead to an optimal decision for any system, but rather a satisficing decision (Simon, 1996). To these ends, and to potentially increase the level of "good" in these "good enough" decisions, MERA aims to uncover effects arising from ordered epochs and design changes, which does not require probabilistically characterizing the uncertainty space.

5.4.5.2 Computational Complexities of Era Spaces

Let L = the number of epoch variable levels for each epoch variable, and v = number of epoch variables. Let U = the number of possible epochs, which can then be described as:

$$\mathrm{U} = \prod_{i=1}^{v} L_i$$

Assume that each epoch always lasts the same amount of time (i.e., every frame containing that epoch is always of the same duration). Recalling the "branching factor" from the discussion on computational complexity of networks, then in the case in which any epoch can transition to any other epoch, the branching factor at each frame of an era is U. Let n = the "depth" of the network (i.e., the number of frames in the eras modeled). If n is in the range of 15 to 20 for a particular study, then the number of possible eras is at most:

$$\mathrm{U}^n = \mathrm{U}^{20} = (\prod_{i=1}^{v} L_i)^{20}$$

A model of 5 epoch variables with 3 levels each would then result in a maximum of somewhere between 10^{34} and 10^{46} possible combinations of epoch developments over an era. As an example, the VASC application to the Space Tug case study produced a branching factor of essentially around 8, with 12 different durations; and the depth could be considered 20, since each epoch lasted 6 months on average, and the eras were 10 years in length. These parameters lead to a roughly estimated minimum of 10^{38} possible eras. In a similar application to the XTOS case study, only one duration was modeled, but the branching factor was 58, and the depth was 20, leading to around 10^{34} possible eras.

Even without considering each era-level strategy operating with $O(b^d)$ complexity in the worst case to evaluate the trajectories of design-strategy pairs, the prospect of reasonably exploring the era space is daunting. It is daunting even without considering multiple design-strategy combinations, and without considering storage of the associated metrics and meta-metrics. Varying temporal durations can also play a large part in the path dependencies of epochs and designs. Unfortunately, modeling durations from a discrete set adds significantly to the number of possible eras to simulate, since they directly increase the branching factor (as the Space Tug case demonstrates).

Of course decreasing the number of epoch variables and epoch variable levels can significantly decrease the size of the era space, and any epoch transition rules present will likely lower the size of the space by orders of magnitude as well. But numbers of this size should bring to light the potential difficulties in ever meeting some minimum confidence level through appropriately sampling the uncertainty space of system lifetimes. Combinatorial problems often require far more sophisticated techniques simply to ensure unbiased sampling (Erenrich & Selman, 2003; Lin & Iii, 2012). Such techniques could certainly be studied in future work to attempt to characterize the space in various ways, but they are considered to be unnecessary to the present purpose. The goal of MERA is not to cover the uncertainty space within a certain confidence level, but simply to *identify the relevant*[1] *path dependencies* of epochs and designs throughout the developing futures.

[1] "Relevant" with respect to the specific goals of the study, strategies created, and stakeholder values.

5.4.5.3 Methods of Era Creation

With the objective of identifying path dependencies in the era space, methods are now proposed for era creation. Each one is relevant to unique aspects of the computational problem of creating long-term eras made up of many epochs. As a result, methods can be combined to leverage the strengths of each, as is further discussed below.

a. Human-in-the-loop

In the first application of RSC to the Satellite Radar System, a group of experts created several different scenarios, which analysts used to then create additional numbers of eras from those epochs, durations, and transitions (Ross et al., 2008, 2009). This human-in-the-loop activity leveraged the cumulative experience of system experts to help create more relevant scenarios in which to examine the path dependencies. Such a hands-on approach is now described.

The problem of finding path dependencies in the era space is very similar to other well-known problems in the realm of discrete combinatorial environments. One of the most directly analogous environments is that of strategic games, such as backgammon and chess, where adversaries are continually trying to force a player into a path-dependent decision that will lead to that player's disadvantage. While computers have beaten human players in games like chess, such victories cost many hours of expert consulting and computer programming, in areas with very long histories of study, established thought and strategy, as well as large databases of previously-evaluated moves ("Deep Blue," 2011). In other, less structured games such as Go, humans remain dominant ("MoGo: A Grid5000-based software for the Game of Go," 2012). These facts demonstrate that human creativity is ideal for finding path dependencies and potentially hazardous (or beneficial) conditions in complex future scenarios with practically uncountable numbers of discrete outcomes. Engineers and analysts could potentially use information specific to the epoch and era constructs, such as the yield of each epoch, expertly generated scenarios, and information from design families (see Section 6.1.4) to assist in complex era constructions. Further research in this area could create a customized tool for era creation that could allow for and augment the natural creativity of human participants in the era exploration process, resulting in long sequences of (10 to 20) epochs that may be of interest in the search for path dependencies for a given system.

To this end, it is possible that epochs could be separated into levels of "difficulty" based on the yield of the epoch and/or any modifications to change options imposed

by an epoch. In other words, epochs with high yield (i.e., high number of feasible designs in the tradespace compared to the total number of designs in the tradespace) could be considered "easy", while epochs with varying amounts of low yield could be considered to have varying degrees "harder". Levels of difficulty could also be inferred from how many change options are available (or hindered) in an epoch, if epoch-dependent change options are modeled.

b. Breadth-First Search through Clips

A beginning approach toward finding path dependencies (of any type) might be to not attempt to sample the set of long-term eras, but rather begin with all possible single-epoch transitions (such as in VASC's Multi-Epoch Analysis) and then incrementally explore the era space by adding one layer of depth (i.e., one more transition) at a time. Such exploration of the era space reflects the activity of Breadth-First Search, the network search algorithm, in performing a one- or two-step look-ahead for decisions made at any given depth of the network. The number of epoch transitions (i.e., the depth) could be increased in this way, with complexity $O(U^{depth})$, until computing resources were depleted. This approach would have several direct benefits:

1) Analyzing shorter eras may enhance the clarity of era-level metrics and reduce complications in understanding results from longer sequences.
2) It is likely that suitably interesting first-order path dependencies would arise after only a depth of 2 or 3 epochs; these first-order path dependencies could then be easily recognized at deeper levels as constituent portions of longer eras and even higher-order dependencies.
3) Detection of second- and third-order path dependencies would be greatly enhanced, since metrics and meta-metrics could be compared for any two depths of simulation (depths 3 and 4 of the same era, for example).
4) Branches that display significantly interesting path dependencies at shallow depths may help inform decisions about which portions of the tree to prune[1] or explore first at later depths, significantly reducing the computational burden of simulating deeper levels.

This approach could also incorporate varying temporal durations. The upper bound on complexity of adding a set of durations size d is $O([d*U]^{depth})$. For example, if

[1] At present, pruning of the era tree is considered dubious practice, since slight alterations of an era can produce vastly different (and likely unpredictable) results. Further research could address this challenge in some way, either proving or disproving the legitimacy of a method for culling the era space, depending on the space described and the objective of pruning. See Chapter 7 for more in-depth discussion.

Space Tug were modeled with 1 epoch variable with 8 levels, with each level lasting 1 to 12 months, and with depth of 3 desired for simulation, there would be $(12*7)^3$ (i.e., ~0.5 million) possible simulations, which is a reasonable number of simulations for modern computing resources. If a depth of 4 was desired, and the number of durations was halved (to 6), then only ~3 million simulations would be required. To preserve the idea of an era as a length of time on the scale of system lifecycles, the term *clip* is introduced here for an era of arbitrarily small size, such as 3 or 4 epoch shifts. A clip can then be thought of as the bounds of rationality (Simon, 1996) on a non-human agent's attempt to optimize in an uncertain future.

c. Sampling

Random samples of the era space can always be obtained, but as noted above, unbiased samples may not be guaranteed. However, the goal of MERA is to identify path dependencies of various kinds, and it is likely that information relevant to these factors can still be gleaned from random samples. It is important to keep in mind, however, that the samples do not necessarily characterize the era space – important to remember since analysts may risk becoming anchored on specific outcomes and statistics (Kahneman & Tversky, 1979, 1984). One approach to near-optimal non-random sampling within a budget of finite sampling resources is suggested in Section 7.3.7.

d. Combination of the Above Methods

It should be obvious that the three methods for era generation listed above are complementary in almost every way. By first auto-generating many clips, analysts can build up eras of interest based on the path dependencies that are found in the initial results. These can be augmented with randomly sampled sequences of eras, providing further opportunity for exploration and feedback into the era generation activity. Clips can be combined[1] and/or built upon, potentially inspiring further ideas for uniquely impactful sequences of epochs, whether those sequences comprise shorter clips, multiple clips, or entire era-length sequences. Expert opinions on likely developments could be incorporated to construct eras of higher interest, as well as various degrees of difficulty of epochs and characteristics of the design families under consideration. In exploring the era space through these several methods, a satisficing determination can be made with regard to the areas

[1] It is important to note that sequencing clips in any way would require a recalculation of an era-level strategy's actions, if an era-level strategy is implemented as an offline planner.

of interest – that is, a determination of good enough quality to more confidently move forward in the evaluative and analysis activities.

5.4.6 Activity 6: Design-Strategy-Era Evaluation

This activity is analogous to RMACS's Process #4, Design-Epoch Tradespace Evaluation. It consists of simulating a design (and strategy, if applicable) through the eras constructed in Activity 5, Era Creation. The output of this activity is the multi-epoch metrics chosen in previous activities, any era-level metrics created in Activity 2, and the trajectories through each era evaluated. By storing each design's trajectory through each era, any metrics can be evaluated on the data without requiring further simulation. (This activity does need to be repeated for a Rule Removal study, however, which is demonstrated in the application of this process in the next chapter.) This activity can also be partially combined with the previous activity in the search for eras of interest, whether through the evaluation of clips before further generation of clips/eras, or through the evaluation of eras in the human-in-the-loop process.

5.4.7 Activity 7: Results Analysis

In general, there are many directions that analysis can take of the data produced by the previous activities. The information available from Activities 5 and 6 will depend, of course, on the epoch-level metrics, era-level metrics, and designs chosen to evaluate through the eras. Questions asked of the data could involve metrics on individual designs, metrics on design families, statistics on the use of change options, statistics on particular epoch variable levels, and any interactions of all of these factors. For example, one could inquire into the Range and Time-Weighted Average of the FPN of Design 32 paired with an era-level Survive strategy through all possible 3-frame clips. One could compare this information with all 3-frame clips where path-dependent changes to Design 32 are located[1] in order to learn more about the epochs and change options that potentially contribute to the path dependencies that emerge in these short sequences. Additional concerns and metrics may be identified for inclusion in another iteration back through some of the previous activities. This type of immersive experience in many futures for a given system may lead to additional insights in viewing a problem differently or

[1] In other words, where the epoch-level Survive strategy does not produce an optimal minimum infeasibility through the era.

generating a solution to particular scenarios of concern (Schön, 1993). The various kinds of analysis that can be performed are now discussed.

5.5 Outputs and Results of Multi-Era Analysis

As stated in the introduction to this chapter, the primary goal of Multi-Era Analysis is to obtain purely descriptive information about the behavior of the system across the era-space. Along the way, much information is generated by each activity, informing and helping clarify communication and expectations regarding many aspects of the system. In addition to the outputs of each individual activity, the outputs for analysis in Activity 7 are potentially limitless. The nature of the output depends on the inputs provided and the desired variant of MERA. A flowchart of the types of output achievable from each type of input provided to the process is shown in Figure 5-5.

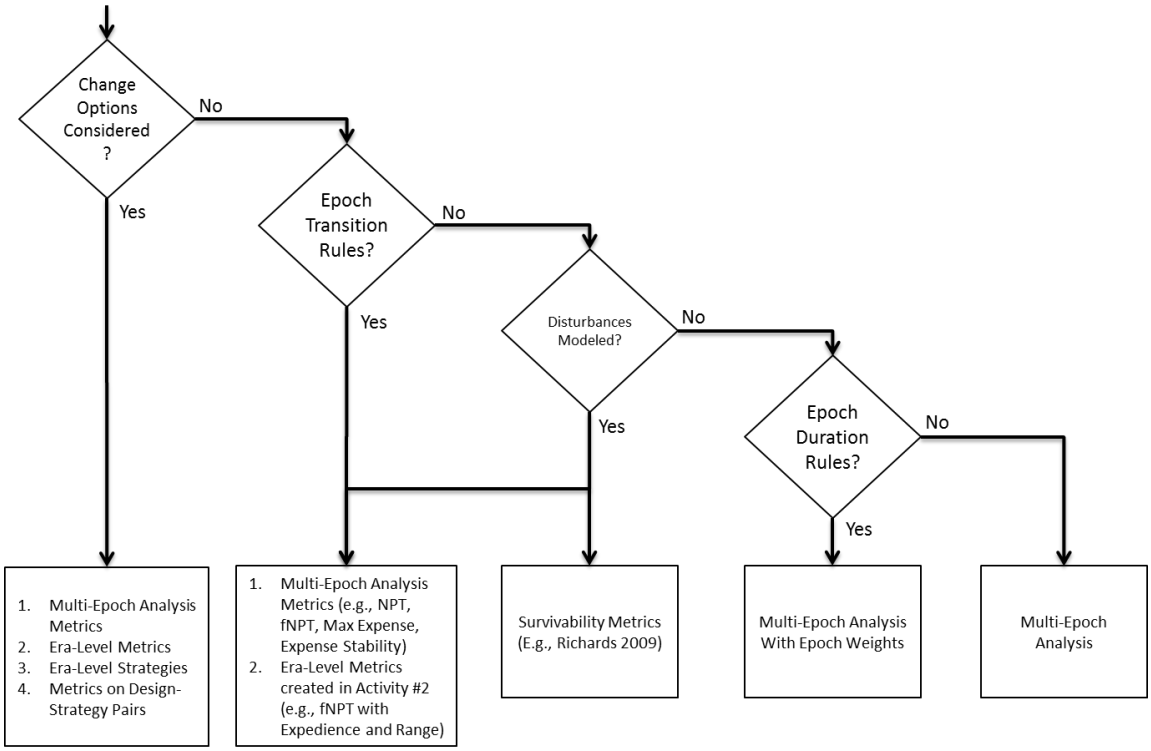


Figure 5-5. The outputs of Activity 7 from each type of input modeled.

The descriptive information provided by MERA is perhaps the richest source of data in the entire design method for analysts and decision-makers to consider when analyzing potential designs and strategies. First, identifying troublesome epoch transitions can be most useful to inspiring the creation and inclusion of new change options to the design study. Second, contingent on adequate input to the MERA process, almost any question of interest can be answered by mining the results of

Activity #6, Design-Strategy-Era Evaluation. The questions of interest can involve any combination of designs, metrics, tactics, strategies, change options, epochs, clips and eras.

5.5.1 Identification of Troubling Perturbations and Eras

The **Greatest Instant Change** meta-metric proposed in Section 5.3.3.4 can help quickly identify perturbations (disturbances or epoch shifts) that impart particularly drastic effects on designs and design-strategy pairs. In addition, the **Range** and **Variability** meta-metrics can quickly provide insight as to which sets of temporal developments should be more closely examined. This type of analysis can be used to generate additional change options for inclusion in the study, depending on the specific types of perturbation sequences that are discovered.

5.5.2 Without Change Options: Epoch- and Era-Level Metrics

If the MERA activities do not include the study of change options, helpful information can still be gained from the addition of MERA to the previously-completed epoch-level analyses (such as Processes 5 and 6, Single- and Multi-Epoch Analyses). While all of these cases can implement the same metrics as Single- and Multi-Epoch Analyses, the metric values produced in MERA will better represent the overall knowledge and intentions of the modeling process. Such Multi-Epoch metrics include the Normalized Pareto Trace (NPT), fuzzy NPT, Max Expense, and Expense Stability, among others. Each of these metrics can be recorded individually for every design-era pair, or aggregate measures can simply be recorded while evaluating all eras (such as maximum value in an era, minimum value in an era, and/or average value in all eras). In addition, era-level metrics can be applied to any design-era pairs, producing meta-metrics that can be further analyzed to determine the temporal effects of sequenced epochs and durations.

5.5.2.1 Modeling Disruptions and/or Disturbances

The first case in which MERA can be informative without change options is one in which small numbers of short-term effects such as disturbances are modeled. (Richards, 2009) simulated many eras expressly for this purpose. Each stochastically-generated disturbance in the Monte Carlo simulations comprised 3 epochs: one pre-disturbance, one during the disturbance, and one during recovery. By sampling the continuous probability space of perturbations, Richards produced enhanced "tear" tradespaces. These were based on metrics such as Time-Weighted Average Utility Loss and Threshold Availability, which were a type of era-level metric that required constant utility functions throughout the era. These era-level

metrics directly supported his studies of system survivability, which is one type of application of MERA without change options.

5.5.2.2 Modeling Meta-information on Epoch Transitions

The second case in which Multi-Epoch metrics can be helpful at the era-level is when epoch transitions predictably follow some set of rules (whether deterministic, probabilistic, or some combination of the two), as discussed in Consideration of Meta-Information on Epoch Durations and Transitions. In this case, it is likely that Multi-Epoch metrics applied (e.g., fNPT) will significantly differ from those in Multi-Epoch Analysis, directly reflecting how the path dependency of epochs affects the designs of interest in the study. Additional information may be gained through the application of era-level metrics to design-era pairs as well.

5.5.2.3 Modeling Meta-information on Epoch Durations

In the absence of meta-information on epoch transitions, epoch durations may still be modeled. However, if the durations are simply probabilistic and time-independent, then the MERA results will be no different than Multi-Epoch Analysis (simply using probabilistic weights on each epoch). Era-level metrics will provide no additional information in this case, since the meta-metrics measuring temporal and epoch-shift aspects will reflect the lack of path dependency of epochs and designs.

5.5.3 With Change Options: Metrics on Design-Strategy Pairs

5.5.3.1 Identifying desired era-level strategies

The first key piece of information produced by MERA when including change options is the identification of desired era-level strategies. The activity of defining these strategies within the context of the designs and eras modeled encourages decision makers and analysts to carefully consider the assumptions made as the long-term effects of strategies begin to emerge. The strategies resulting from this activity will be some of the most concrete statements of stakeholder/decision maker value corresponding to the Value-Driving Purpose Statement from Process 1 of RMACS.

5.5.3.2 Comparison of strategies

If more than one era-level strategy is created, multiple strategies can be compared by analyzing the results of those strategies when paired with designs of interest. These comparisons could include how many changes are required per strategy, and which change options are most used for each strategy, among other considerations.

Comparisons based solely on statistics from era samples, however, may run the risk of mischaracterizing the era space (see Section 5.4.5).

5.5.3.2.1 Defining and discovering path dependency of changing designs

Another benefit from defining era-level strategies is the ability to automate the discovery of relevant path dependencies present in the change of designs throughout eras. In the presence of constructed era-level strategies, one type of relevant design-change path dependency could be defined as *a sequence of perturbations in which an era-level strategy takes different actions than a corresponding epoch-level tactic*[1]. Whether the goal is minimizing or maximizing some metric (e.g. operating cost, FPN, etc), it is possible that an era-level strategy (with or without constraints on execution) will differ in its course of action during a certain frame when compared with an epoch-level tactic created to reach a similar goal. The differences in epoch- and era- level Survive strategies discussed in Section 5.3.3.2 form one example of the potential differences. Such differences in action make for excellent points of study into the assumptions behind any strategy, further generation of change options, and identifying potential pitfalls in the use of existing change options by the defined strategies.

5.5.3.2.2 Cost comparison of strategies

If more than one era-level strategy is created in Activity 3, then it can be informative to pair the same design with each unique strategy and note the different cost outcomes. Several cost factors could be examined, since many aspects of cost are present at this level of analysis. One type of analysis could simply compare the cumulative cost of changes prescribed by a strategy over the eras considered. If a strategy turns out to be more expensive than those it is compared with, adding certain constraints to (or even slightly altering the implementation of) that strategy could bring its change costs down to a more reasonable level. Alternatively, it is possible that a strategy accomplishing its goal with far less cumulative change cost than another strategy could be further constrained to incorporate aspects of the more expensive strategy, potentially killing two birds (i.e., meeting two strategic goals) with one stone (i.e., strategy).

Another aspect of the cost of a strategy has to do with the attributes of the trajectory it takes through the era. The MAE (as well as individual resource) levels of designs

[1] Certainly other types of relevant design-level path-dependencies exist, as Chapters 6 and 7 will discuss.

along the trajectory can be compared between two different trajectories produced by unique strategies paired with the same design. Meta-metrics can be recorded on these factors as well to help identify particular sequences of epochs that cause significantly different outcomes for the strategies under consideration.

5.5.3.2.3 Other areas of analysis

Another activity suggested in the VASC study by (Fitzgerald, 2012) is the Rule Removal study, which performs the same analysis again, but omitting a particular change rule of interest (as the name suggests). The differences in outcome can be informative with regard to the impact of including or excluding a change option, thereby providing further insight as to the nature of the tradespace itself, its connectedness, and the value of change options.

Of course many other potential areas of results can be examined for a specific problem. One small subset of the possible analyses of results is demonstrated in Chapter 6. With regard to path-dependencies of changes in eras, questions asked of the data can involve the correlation of epochs with execution of change options, correlation of change option execution in only path-dependent sequences, correlation of frames in path-dependent sequences, and least/most executed change options across clips. Comparisons of design-strategy pairs may examine the maximum and minimum costs incurred through clips, the distribution of meta-metric values over all evaluated clips, correlation of change option execution with particular frames and/or clips, and many other aspects, limited only by analysts' imagination with regard to what information is relevant to the study.

6 Application of Multi-Era Analysis to NGCS system

> "But the Mirror can also show things unbidden, and those are often stranger and more profitable than things which we wish to behold...For it shows things that yet may be."
>
> -Lady Galadriel (Tolkien, 1974)

The previous chapter described the nature of and various activities in Process 9, Multi-Era Analysis. This chapter demonstrates a simplified application of the MERA process to an NGCS-like case application very similar to Chapter 4. It begins with a description of the information inputs to the process, relating each set to those described in the beginning of Chapter 5. The chapter continues with a note on the explanation of the generation of change options. Each activity in the Multi-Era Analysis process is then described, with the accompanying outputs from the case application where appropriate.

6.1 Information Inputs to the Process

The information inputs to this application of MERA include most of the data generated in the first 8 steps of the RSC-based method. Since the MERA process can be more informative when considering designs with change options, the original dataset has been expanded from 6 point designs to include a tradespace network with change options connecting many of the designs. Though the underlying data comes from the same performance model (the MIT Math Model) for similar class frigates, the specific design variables and attributes included have been slightly modified from that of the 6 original designs in order to further accommodate the inclusion of change options over the design network. The epoch variables have been modified as well to better reflect impacts on the updated attributes. The new data and the modifications were made in conjunction with the naval subject matter expert referenced for the original 6 designs. This new design tradespace forms a portion of the dataset available in the application of MERA to the NGCS-like system.

6.1.1 Sets of Data

Recalling the inputs described in Section 5.2, the sets of information provided for this study are: 1) a set **X** of 134 feasible design variable configurations, 2) a set **E** of 96 feasible epoch variable configurations, 3) a set **Y** of 12,864 evaluated utility attributes and expense attributes for every design-epoch pair, generated from the performance model, 4) a set **V** of 12,864 MAU and MAE values for every design-epoch pair, generated from the value model. In addition to these sets, a set **Δ** of 3

unique change options is included to demonstrate the aspects of the process that deal with evaluating changing designs in many possible eras.

6.1.2 Generation of Change Options

Three change options were identified as of interest for this study, based on analysis by the subject matter expert and modeler. The options form three unique types of change, with varying levels of impact and cost. The first change option included is the Payload Upgrade, through which a ship may indirectly upgrade its Air, Weapons, and Communication systems. If a ship initially has the lowest payload level (Level 3), it costs $3 million and 1 month in port to upgrade to Level 2. To upgrade to Level 1 costs more, as the technology is more advanced ($10 million and 2 months in port). This change option provides 245 connections between designs in the tradespace, or a little over 1% of the space. The second change option included is the Add Engine option, which costs $5 million and 2 months in port if the ship initially has 2 engines, the lowest level. If a ship initially has three engines, then the upgrade to 4 engines costs $1 million and 1 month in port. This change option provides 372 connections between designs, or around 2% of the space. The final change option included is the Increase Length option, which represents adding a 20-30 ft. section to a ship's hull. This option is the most expensive of the three, but also potentially the most beneficial. The Increase Length option always takes 6 months in port and $20 million to $30 million, depending on the initial length of the ship. It allows the ship to have more capabilities, whether immediately (in some cases) or by allowing other changes that were previously unavailable. This change option connects the most number of designs, with around 1200 connections in total, or almost 7% of the space. Sparsity plots for each of the three transition matrices are shown below on the left sides of Figure 6-1 through Figure 6-3, where each cell in each matrix contains the cost of transitioning from Design *i* (row number) to Design *j* (column number). If a cell is blank, there is no transition available from that Design *i* to Design *j*.

Alongside each transition matrix, on the right sides of Figure 6-1 through Figure 6-3 is a force-directed visualization of the network created by that particular change rule. Each design alternative in the network is represented by a blue dot, with gray edges whose widths reflect the cost of traversing that edge. (Note that the edges are in fact directed, although the current version of this plot does not reflect the directionality of any edge, which can lead to egregious misinterpretations of the output.) The spatial position of the dots is only determined by that design's relation to all other designs (the "force-directed" portion of the name of the plot), with unconnected designs in the outer part of the plot; unlike the sparsity plot, the

position is not indicative of which design or transition is represented by a particular dot or edge. The plot is a static screenshot of an interactive visualization created through web technologies (HTML/CSS/Javascript/D3)[1], which allows a user to investigate points of interest and find specific designs and edges of interest.

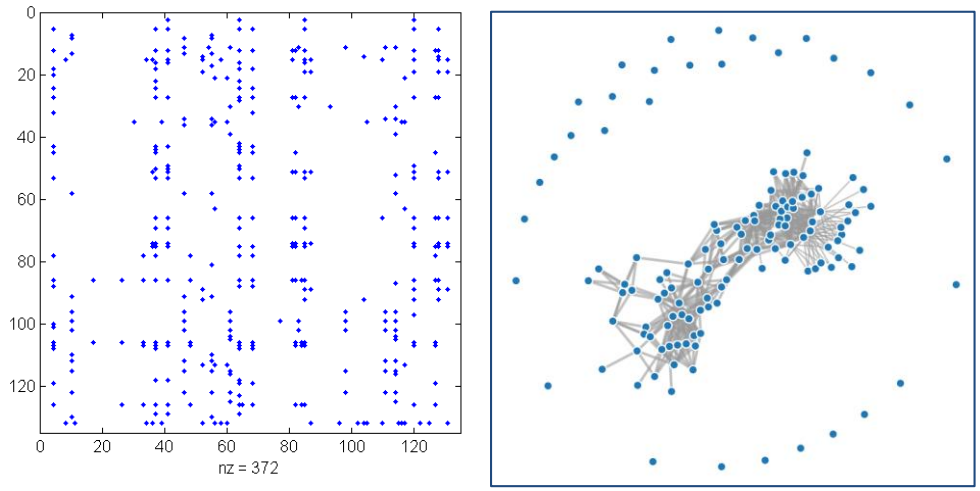


Figure 6-1. Sparsity plot of the transition matrix for Add Engine, alongside the corresponding force-directed network plot. The spatial position of each colored cell in the matrix on the left indicates which design numbers correspond to a transition, while spatial positions of each node on the right indicate a design's relation to other designs. Nodes in the central portion of the connected cluster may have a high degree of betweenness (Freeman, 1977) **relative to nodes on the outer portion of the connected cluster.**

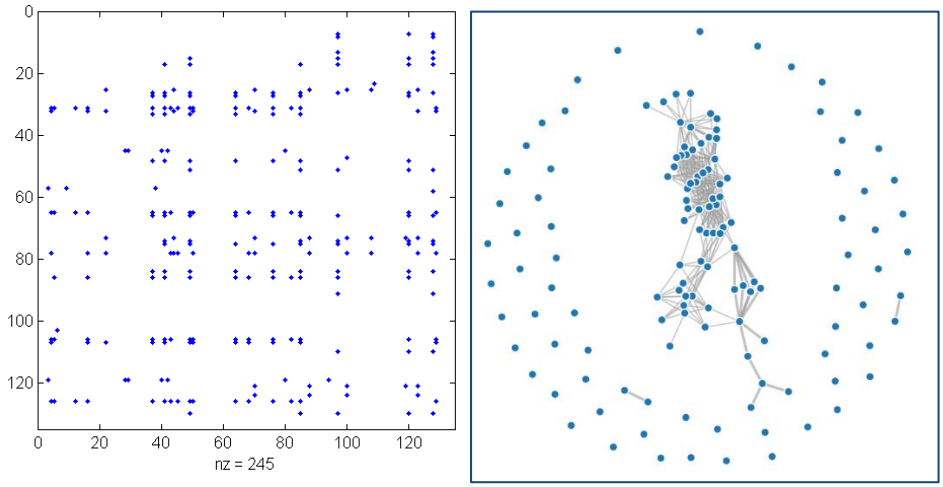


Figure 6-2. Sparsity plot of the transition matrix for Upgrade Payload, alongside a force-directed network plot. The spatial position of each colored cell in the matrix on the left indicates which design numbers correspond to a transition, while spatial positions of each node on the right indicate a design's relation to other designs.

[1] The Appendix contains the code for this tool, which is an example provided by http://d3js.org/.

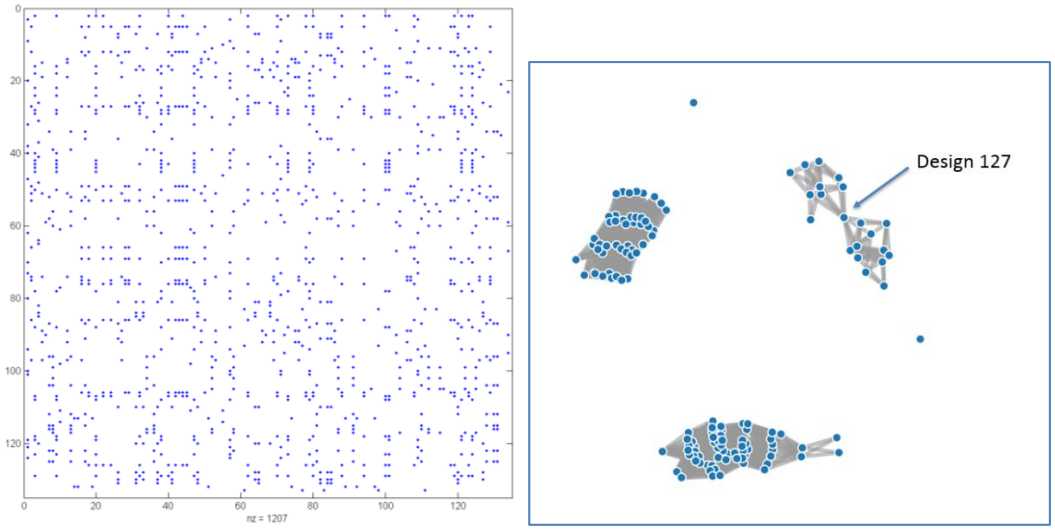


Figure 6-3. Sparsity plot of the transition matrix for Increase Length, alongside a depiction of the network created. The spatial position of each colored cell in the matrix on the left indicates which design numbers correspond to a transition, while spatial positions of each node on the right only indicate a design's relation to other designs. For example, Design 127 may have a high degree of betweenness relative to the other designs (Freeman, 1977)**.**

As the plots show, the options benefit each design in various ways. Some of the designs can use the Payload Upgrade, a few more benefit from the Add Engine option, and most designs can utilize the Increase Length option. (The code used to generate these change options and their visualizations is included in the Appendix.)

6.1.3 Epoch-Dependent Change Options

The financial cost of the change options described above are modeled to be directly dependent on the epoch variables in several ways. First, the cost to execute the Add Engine change option decreases slightly in epochs that have increased Automation Levels for ships, as the technology enabling autonomous systems is assumed to enable more efficient engines. Second, the cost of the Payload Upgrade option is modeled to be slightly more expensive in epochs with higher Fuel Price levels, as it is assumed that the costs of acquisition and installation of the new payloads depend somewhat on fuel prices. Third, the Increase Length option is also modeled to be more expensive in the higher Fuel Price epochs, since the cost of major construction depends on fuel prices (and on materials directly dependent on fuel prices).

In addition to financial costs, the availability of the change options is also modeled to depend on the epochs. In epochs that the NGCS-like system is part of a Battlegroup (i.e., it is a constituent system in an SoS), none of the above change options are allowed, as all changes would require returning to port and thereby discontinuing support of the SoS. In epochs with the most demanding missions

(resulting in demanding sets of preferences), the Increase Length option is not allowed due to the urgency of the missions.

With change options elicited and these transition rules established for the tradespace network, the MERA activities can begin.

6.1.4 On "changeability" provided by change options: the Full Accessibility Matrix

An approach to valuing the introduction of change options to a design study was explored extensively by (Fitzgerald, 2012). One of the topics reviewed therein is the Filtered Outdegree (FOD) proposed by (Ross & Hastings, 2006), as well as the several iterations that the later Value-Weighted Filtered Outdegree (VWFO) metric has gone through to capture various ways to measure the value of the changeability of systems (Hastings, 2010; Viscito, Chattopadhyay, & Ross, 2009; Viscito & Ross, 2009). While the FOD was simply intended to measure the number of changes that could be made to a system, all versions of the VWFO were attempts to capture the epoch-dependent differences in utilities of designs resulting from the changes made.

From the current work's emphasis on long-term strategies and long-term value to stakeholders, it should be clear that even change options that lower a design's utility in a current epoch might vastly improve its long-term value to stakeholders, given particular strategies and epoch developments. For this reason, it is here proposed that the FOD may be a more appropriate indicator of the value of changeability over the long-term in some cases, providing both a minimum bound on the *number of potential end states* and a minimum bound on the *number of ways* to get to those end states. The FOD traditionally only considered single-arc transitions, but with the creation of a full-accessibility matrix, it becomes possible to measure the true number of potential end states for any given design, also providing a minimum bound on the true *number of ways* that a design can change, considering all combinations of change options. The FOD applied to all arcs could be deemed the Fully Accessible Filtered Outdegree (FAFO) metric to differentiate it from the original FOD metric. Similarly, the unfiltered version could be denoted the Fully Accessible Outdegree (FAO) to differentiate it from the unfiltered outdegree.

6.1.4.1 Efficient creation of a full accessibility matrix

One version of the full accessibility matrix (including all combinations of change options) can be created simply by finding and storing the shortest path between all nodes. To efficiently compute the shortest path between all nodes for the present work, Djikstra's algorithm was implemented in Matlab without the use of a Fibonacci heap. With the implementation of a Fibonacci heap, the complexity of Djikstra's for the single-source shortest path problem is $O(|E| + |V|\log|V|)$, where

|E| is the number of edges in the network, and |V| is the number of total nodes. (Note that the upper bound could be much lower, as this complexity assumes that all nodes are connected to a single network, which may not be the case in a tradespace network created by change options, as the following section demonstrates in the Space Tug and X-TOS examples.) To solve the all-pairs shortest path problem, then, the complexity is $O(|V||E| + |V|^2 \log|V|)$. Again, this is likely a pessimistic upper bound in tradespace networks due to potentially disconnected regions. In two tests on small tradespaces with a few change options (the NGCS-like and Space Tug systems), the Matlab implementation took a few seconds to run on a workstation a few years old. It finished in 7 hours for a tradespace with around 3,400 designs and 36,000 initial edges provided by 8 different change options (the X-TOS satellite system), producing around 1.5 million edges total. It is expected that implementation of a Fibonacci heap would improve these times.

Djikstra's algorithm operates as follows:

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes unvisited. Set the initial node as current. Create a set of the unvisited nodes called the *unvisited set* consisting of all the nodes.
3. For the current node, consider all of its unvisited neighbors and calculate their *tentative* distances. Compare the newly calculated *tentative* distance to the current assigned value and assign the smaller one. For example, if the current node *A* is marked with a distance of 6, and the edge connecting it with a neighbor *B* has length 2, then the distance to *B* (through *A*) will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set*. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the *unvisited set* is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

("Djikstra's Algorithm," 2014)

Running this algorithm on transition matrices for both financial costs and temporal costs can provide lowest money- and time-costs for change paths for every combination of change options. A potentially more informative version of the

algorithm could be developed to store all of the possible non-dominated change paths in the course of running the algorithm, but such storage is not strictly necessary for simply counting the number of potential end states – only the shortest path needs to be kept for this latter purpose, which give the Fully Accessible Outdegree (FAO) for each design[1]. The results may then be filtered according to whatever upper bound on cost is desired, producing a matrix that corresponds directly to each design's FAFO. Multiple versions of this matrix can be generated in the case that epochs prevent change options or otherwise affect the cost of execution, such as those epochs modeled in the NGCS-like case. (Though since the resulting matrices are of size N^2, where N is the number of designs in the tradespace, multiple matrices in larger tradespaces can prove to be more challenging.)

6.1.4.2 Application of Full Accessibility Matrix to case examples

A full accessibility matrix was created for the NGCS-like tradespace network in the baseline epoch. The matrix and corresponding network depiction is shown below in Figure 6-4.

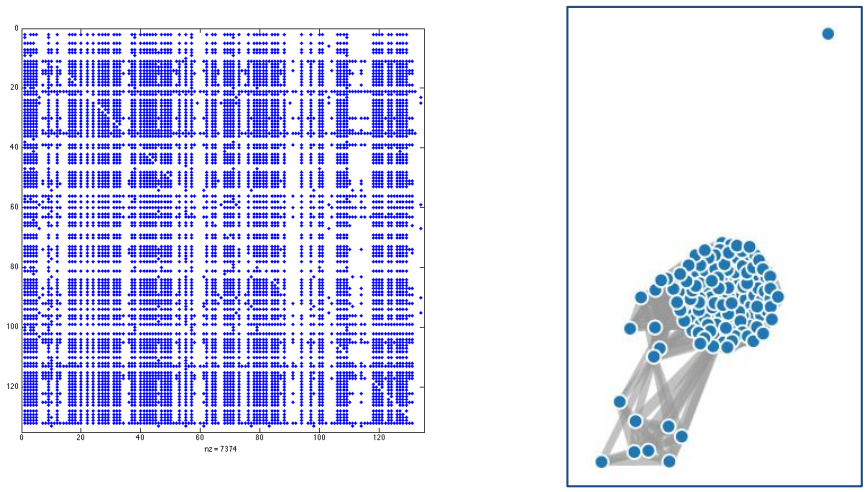


Figure 6-4. A full accessibility matrix of the NGCS-like system (left), with the associated force-directed network visualization (right). The nodes in the bottom left of the network depiction are those designs which cannot reach the central cluster of designs (all edges are directed, but directionality is not indicated by this particular visualization).

[1] Another potential modification to the algorithm, depending on problem type, may consider only specific sequences of change options and/or upper bounds on number of uses of a given change option. These modifications were not considered to be necessary for the present demonstration.

While the FAO is equal for most of the designs in the NGCS-like case, other cases have different results. The full accessibility matrix and network depiction for the Space Tug system are shown below in Figure 6-5.

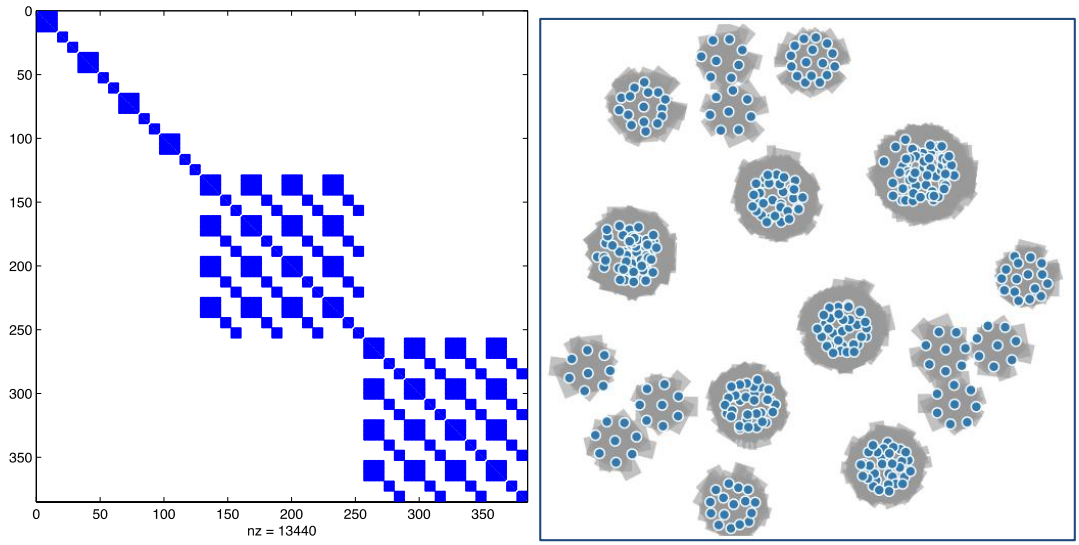


Figure 6-5. A full accessibility matrix of Space Tug (left), with the corresponding force-directed network visualization (right). Many designs have a FAO of 7 to 9, while many more have much higher FAO values.

The network depiction is more interesting in the case of Space Tug, due to the emergence of groups of designs. (Fitzgerald, 2012) discusses the prospect of "design families" emerging from particular sets of strategies, designs, and change options, as they encounter various eras. This network visualization can indicate the potential extent of such families, given any strategy that could be created. The FOD applied to all arcs (i.e., the FAFO metric) is a direct indicator, then, of the maximum size of a design's family under all changes under a cost threshold, with the resulting size of the family in practice being limited only by the particular strategy applied to the use of change options and the particular eras that the strategy/design pairs encounter.

Figure 6-6 shows the full accessibility matrix for the X-TOS satellite case study, containing a tradespace with over 3000 designs. In this case, the sparsity plot was made of only the first 500 designs to show the patterns of transitions. The force-directed network visualization was also only generated for the first 500 designs due to the computational burden of plotting the excessive number (~1 million) of edges in the full accessibility matrix.

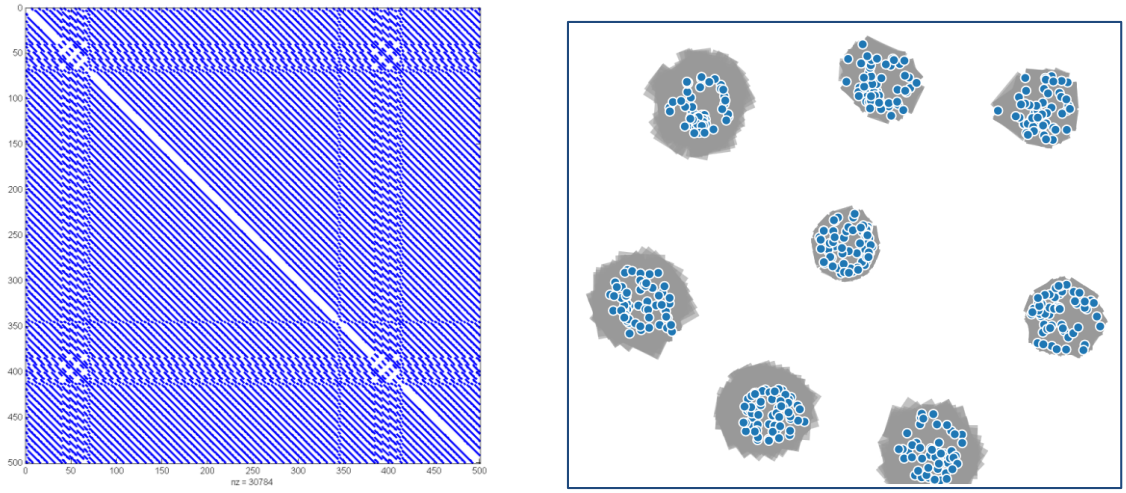


Figure 6-6. A full accessibility matrix of the first 500 designs of X-TOS (left), with the corresponding force-directed network visualization of the 500 designs (right). Gray areas imply many edges connecting the clusters of designs (blue dots).

The results of the X-TOS force-directed network visualization look similar to that of Space Tug, which is not surprising since they both are satellite systems with similar types of change available after launching into orbit.

6.1.4.3 Summary of Full Accessibility Matrix Generation

While at first glance it may appear to be tangential to the present application of multi-era analysis, the creation of a full accessibility matrix can provide much helpful information before beginning to create or simulate any eras. The emergence of the concept of general design families can inspire particular sets of epochs to be included in the creation of clips and eras, depending on which designs are in a general family as well as which change options connect a general family. In addition, the directionality of the connections is important, though presently not depicted in the network visualizations above. Future research could better incorporate all of this information in order to provide more effective tools for finding and representing the properties of the general design families encountered in design tradespaces, especially as they interact with the epoch space. Epoch networks could be visualized in the same way, further enabling the effectiveness of the human-in-the-loop portion of era creation discussed in 5.4.5.3 (see Section 7.3.5 for more discussion on this point).

6.2 Activities of Multi-Era Analysis

While all seven activities are performed in the application of MERA to the NGCS-like system, each activity is scaled down significantly for considerations of space and clarity. The first two activities connect this process as a continuation of the NGCS

design study of Chapter 4. The third and fourth activities, while optional in the general MERA process, are presently applied to demonstrate the results and analysis possible by including era-level strategies in the design study. The fifth activity demonstrates ways to down select epochs to include in the eras based on epoch yield and impact on change options, and the sixth activity evaluates the designs and strategies in the eras of interest. Finally, the seventh activity conducts a small portion of the analysis possible, primarily using interactive visualizations, in order to demonstrate ways of contextualizing the results and locating path dependencies of interest.

6.2.1 Activity 1: Identify Desired Multi-Era Analysis Variant

In this first activity, the desires of the overall design study are considered, and incorporated with the information inputs available to this final process of the method. The stated desire of the NGCS design study is to develop an affordable naval ship system. Recalling the definition of affordability as "remaining or becoming feasible", then, the analysis will be related primarily to the feasibility of the system (with respect to minimum utility and maximum expense levels) and also to the actual expense levels of the system.

Considering the inputs to the process, change options are available to the designs, meaning that the study will attempt to identify the path-dependence of changing designs. Limited information is available on epoch durations, but will be modeled as well. No information on disturbances or disruptions will be included, meaning the desired variant will be Safe MERA. Although the Prescriptive Information (i.e., online planner) portion is possible to include anytime change options are provided, such activity will be omitted due to the representational nature of the NGCS study.

Activity Outputs: Feasibility is of primary concern, with MAE levels secondary. The Descriptive portion of MERA without disturbances will be conducted, with the primary goal of identifying path dependencies in the use of change options.

6.2.2 Activity 2: Select designs and epoch-level metrics of interest

Now that the goals of the study have been established, designs and epoch-level metrics of interest can be selected accordingly. From the tradespace of 134 feasible ship designs, the fNPT (with 5% fuzzy level) was calculated for all designs across all epochs, producing a list of 17 designs with fNPT value of 1 (i.e., within 5% of the Pareto front in all epochs). From this list of 17 designs, 3 designs were selected based on the design variable levels of the final designs from the first 8 processes in the NGCS study. The three designs and their design variable levels are shown in Table 6-1.

Table 6-1. The design variable levels of the NGCS-like designs of interest A, B, and C for the MERA process. Lower payload levels indicate higher levels of capability.

Design	Length	Beam	Draft	Payload	Engines
A (id 68)	527	55	17	2	3
B (id 35)	450	51	15	3	2
C (id 11)	457	51	15	3	3

Design A reflects designs 2 and 4 from the NGCS study, with similar length and payload options. Designs B and C reflect design 3 from the NGCS study. Recall that design 3 was removed from consideration in the course of that study due to its infeasibility in several epochs. With the addition of change options in the Multi-Era Process, these designs may provide ways for variants of design 3 to become feasible, potentially making it an affordable alternative in the course of the study.

While these three designs would continue the study of the previous 8 processes of RMACS, many more designs will be evaluated in the present application in order to better illustrate the wide variety of analysis that can be performed and the many aspects of results that may be of interest in the course of general design studies. For this reason, Designs A, B, and C will not be referenced after this activity.

Activity 1 stated that the primary aspects of the analysis regard feasibility and expense levels. The epoch-level metrics of interest, therefore, will include both the MAU and MAE values, as well as the financial and temporal cost of executing change options. Consideration of these three metrics in any epoch should then adequately capture the aspects of concern throughout the process. It is assumed that the stakeholder has no preference on the path analysis metrics of Section 5.3.3.4, and as a result the creation of era-level metrics in this activity is unnecessary.

Activity Output: Designs A, B, and C are noted to be most similar to designs resulting from the first 8 processes of RMACS, with respective design variable levels given in Table 6-1. Although these three designs would be selected to continue the previous case application in Chapter 4, many other designs in the tradespace will be evaluated in order to better demonstrate the variety of analysis that can be performed. The metrics of interest will be MAU, MAE, and financial and temporal costs of change options. No era-level metrics will be considered, as it is asserted that the representative stakeholder has no preferences over the temporal aspects of any of these metrics.

6.2.3 Activity 3: Create era-level strategies

Given the goals of the analysis described above, the strategies created during this activity should reflect the desire for feasibility and lower expense levels. For clarity and simplicity in demonstrating the results, only one strategy was created: Survive. This strategy is similar to the epoch-level Survive tactic; that is, when a design is infeasible in an epoch, the strategy will use its shortest and cheapest change option to change to a feasible design (if possible). This goal well mirrors the goal from Activity 1, which is to find feasible (and therefore affordable) design-option pairs. One of the key differences of the era-level Survive strategy from the epoch-level tactic, however, is that the era-level strategy seeks to minimize the time spent in an infeasible state *over the entire era*. The epoch-level Survive tactic only considers the duration of the current epoch, and if a design becomes infeasible, the tactic executes the cheapest change option to reach a target design. The target design may become infeasible in the very next epoch, but such information is not considered in the epoch-level tactic. The era-level strategy, on the other hand, navigates all possible paths of a design and its change options through the design-tradespace meta-network over the era, and chooses the cheapest path with least infeasible time. This consideration of all paths might mean preemptively changing to a design which does not become infeasible, or it might mean executing a more expensive change option in the present epoch to prevent having to change again in a future epoch. (It is important to note here that time spent changing is assumed to be time spent infeasible, since in the NGCS-like case, time spent changing requires being docked in port and doing nothing else.) One of the assumptions for this case application included that the strategy could never change to an infeasible design in minimizing the total infeasible time spent through an era.

6.2.3.1 Strategy Implementation

The era-level Survive strategy was implemented using the A* algorithm discussed in Section 5.3.3.2. The A* algorithm explores the design-tradespace meta-network over the era by only considering the nodes with promising (i.e., heuristically evaluated to be low) "costs" to traverse. The cost of being in X_i in a given frame of the era is essentially defined by its feasibility in that frame – if it is feasible, then there is no cost for that node. The cost of X_i in one frame of the era traversing the network to X_i in the next frame of the era is also zero, since no change is executed. The heuristic cost of X_i in the next frame depends on whether it is infeasible or feasible – if infeasible, then the heuristic cost is the cheapest change to execute (if change is available; otherwise the heuristic cost is whatever penalty is applied for being infeasible multiplied by the duration of the frame). If X_i is feasible in the next frame,

then its heuristic cost is again zero. The cost of X_i changing to a X_j in the same frame is simply the cost of the change option. Since only feasible X_js are considered by the era-level strategy, their heuristic cost will always be zero. The decision logic for the dynamic construction of this heuristic is shown below in Figure 6-7.

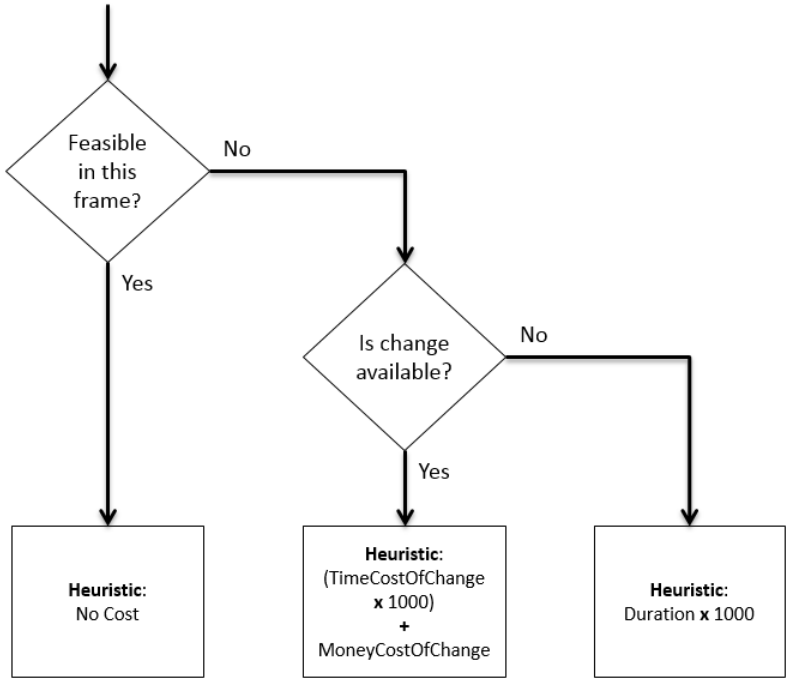


Figure 6-7. Construction of an admissible heuristic for the A* implementation of the era-level Survive strategy. An admissible heuristic optimistically estimates the cost of a design. The penalty applied for a time unit of infeasibility is 1000x more than the base unit of financial cost of change options.

A heuristic estimated in this way is admissible, since it will never overestimate the cost to get to the goal region, which is the end of the last frame of the era). An admissible heuristic results in the strategy always finding the optimal path to the goal region – in this case, the (financially cheapest) minimally infeasible path.

6.2.3.2 Strategy Verification

This strategy implementation was coded in Matlab, with verification of desired output and benchmarks run on the number of nodes explored by the algorithm. The verification portion consisted of examining eras in which the path could be verified to respond appropriately to durations of infeasibility. One such era comprising 3 frames (i.e., a *clip*) is shown below in Figure 6-8, with the starting design 32 initially infeasible in the beginning epoch of the clip.

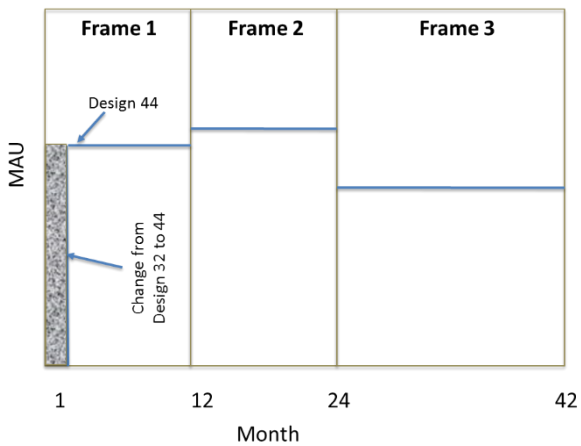


Figure 6-8. The path of the MAU of design 32 paired with the era-level Survive strategy, in a sample 3-frame clip of length 42 months (X axis). Time spent changing is shown as the gray shaded area under the line; in this case, equal to 1 month. (Only one change executed in this era, from 32 to 44.)

The rule executed for design 32 to change to design 44 was Rule 1, the Payload Upgrade rule. This execution cost $10 million and 1 month of time, resulting in design 32 upgrading its weapons to Level 2 (and thereby becoming design 44). It remained design 44 through the era, which was feasible in the next two epochs of the clip. This type of verification in several eras and various designs provides assurance that the algorithm is behaving as desired[1]. The epoch-level Survive tactic was also verified in a similar manner, with the results shown in Figure 6-9 for the same starting design in the same clip.

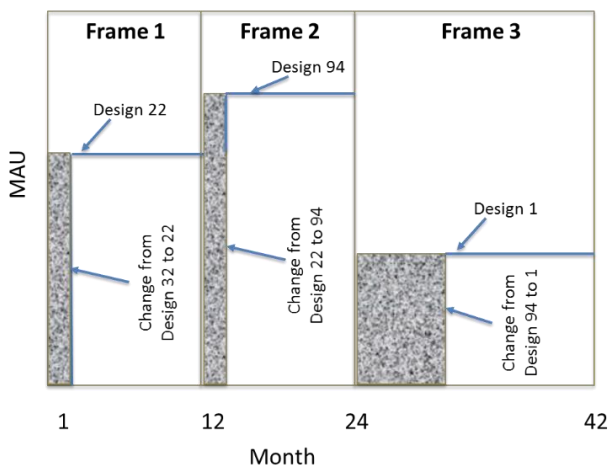


Figure 6-9. The path of the MAU of design 32 paired with the epoch-level Survive tactic, in the sample 3-frame clip of length 42 months (X axis). Time spent changing is shown as the gray shaded areas under each line. (Three changes executed in this era: 32 to 22, 22 to 94, and 94 to 1.)

[1] Only verification was performed; validation was not performed due to the representative nature of the study and consequent lack of actual stakeholder values on the outcome of the strategy.

Although the point of examining this single design in a given clip was to verify the proper operation of the coded tactic and strategy, it serves as a pointer to the comparisons of later activities. One can immediately see the obvious differences between the epoch-level tactics and the era-level strategy with these two plots. The epoch-level Survive tactic does successfully change from design 32 to design 22 using Rule 1, preserving its feasibility in the first frame of the clip. However, it requires changing again in the second frame of the clip, where it again becomes feasible for a short time, until encountering the third epoch of the clip, where a major change (Rule 3, Increase Length) from design 94 to design 1 is required to preserve its feasibility for the rest of the clip. The transition in the first epoch (to design 22) was one of the lowest cost changes available, and as such it was a natural choice for the Survive tactic. However, as the era-level strategy shows, changing to design 44 was the same cost and would have eliminated the need for further changes throughout the era. This difference in "optimal" actions is an example of identifying one form of relevant path dependence in the use of change options.

6.2.3.3 Strategy Benchmarks

For the algorithm benchmarks, one thousand simulations were performed for a random design in eras from length 3 to 20 epochs, and the number of nodes considered in each run was averaged. (One thousand samples for each length was determined to be sufficient due to the lack of "obstacles" in the network and knowledge about the quality of the heuristic.) The results are plotted alongside the number of estimated nodes in the entire network for each length of era, shown in Figure 6-10. The number of estimated nodes in the entire network was obtained using a branching factor (average number of reachable designs) of 4, which is a low estimate of the average branching factor considering the three transition matrices shown in Figure 6-1 through Figure 6-3 and the number of epochs that enable/disable change. Note that a breadth-first search (or depth-first search) for an optimal path would consider the entire number of nodes in the network.

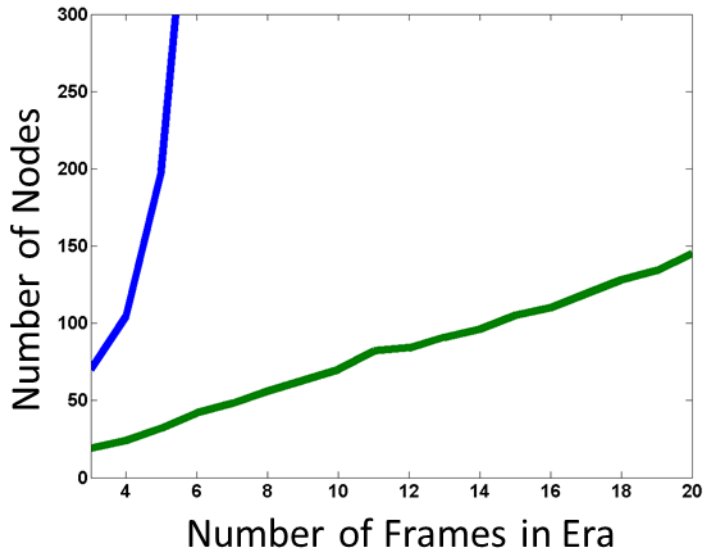


Figure 6-10. Estimated number of total nodes in the design-tradespace meta-network (exponential line in blue) compared with the number of nodes considered by the A* implementation (constant-slope line in green), in 3- to 20- frame eras, average of 1000 samples in each era length.

The remarkably low average number of considered nodes compared to the number of nodes in the design-tradespace meta-network is not surprising in this particular case, given the structure of the network. Many designs can access change options if need be, and many designs are also feasible in many of the epochs. These factors lead to relative ease in finding a low-cost path through the era to the end frame, meaning that the worst-case complexity $O(b^d)$ is not experienced here. The code used for the implementation of the A* algorithm can be found in the appendix.

Activity Output: This activity incorporated the previously identified metrics of concern in creating the era-level Survive strategy (with some assumptions) and epoch-level Survive tactic. The strategy was then implemented in Matlab as a form of the A* path-finding algorithm for discrete networks. It was tested on a sample era to ensure correct actions, and benchmarks were tested for its performance relative to the worst-case complexity. The epoch-level Survive tactic was created in Matlab and verified in a similar manner.

6.2.4 Activity 4: Identify design-strategy pairs of interest

This activity combines the designs identified in Activity 2 with the strategies created in Activity 3. Since only one strategy was created and implemented for the NGCS-like case, that strategy (era-level Survive) can be combined with each design of interest (designs A and B from Activity 2). Each design will also be paired with the

epoch-level Survive tactic in order to locate path dependencies arising from the execution of options as the designs traverse the eras created in the next activity.

Activity Output: Designs A and B were both matched with the era-level Survive strategy and epoch-level Survive tactic.

6.2.5 Activity 5: Create eras

6.2.5.1 Description of the epoch and era spaces of the NGCS-like system

For the NGCS-like study, 96 epochs were provided as input to the process. The epoch meta-information modeled was that each epoch would last either 6 months, 9 months, or 12 months. Eras are considered to be of length 20 years, meaning the minimum number of frames in an era is 10 and the maximum number of frames would be 40. The total size of this era space, then, is a minimum of $(95*3)^{10}$ eras (95 since each epoch is assumed not to transition to itself), or in other words a minimum of 10^{24} possible eras. If only clips are considered, there are over 80,000 possible clips of length 2 frames, and there are over 23 million clips of length 3 frames.

6.2.5.2 Human-in-the-loop

To help bring these numbers into a more tractable realm of exploration, the epochs were broken into groups based on various degrees of difficulty, which is taken as a combination of the yield of each epoch with its impact on the use of change options. The yield of each epoch was plotted on a histogram, shown below in Figure 6-11, where each epoch is represented by a different bar, and the yield of epochs 1 through 97 is represented by the height of its corresponding bar.

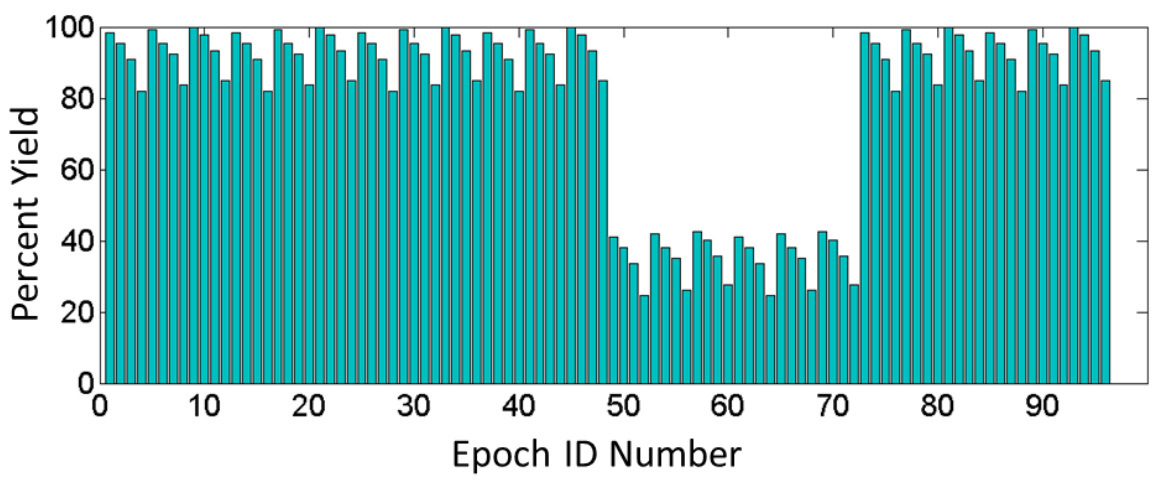


Figure 6-11. Percentage (along the Y axis) of the yield of the 97 epochs (each a bar along the X axis).

Based on this result, the epochs were broken into 4 levels of yield: Very High (yield >= 120), High (100 < yield < 120), Low (40 < yield =< 100), and Very Low (yield =< 40).

The epochs were then analyzed with regard to their impact on change options, with a similar plot shown below in Figure 6-12, which depicts both the availability and cost of change options across the epoch space.

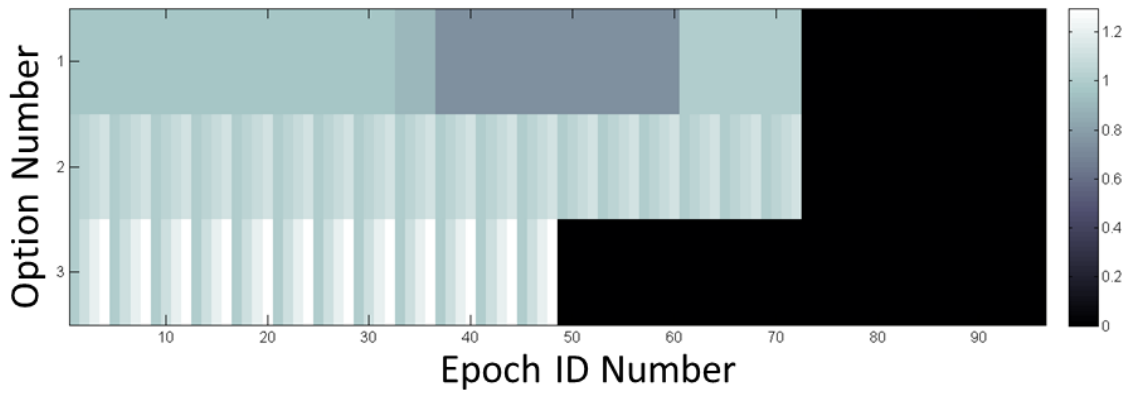


Figure 6-12. Change options (Y axis, descending order: Add Engine, Upgrade Payload, Increase Length) across all epochs (X axis), with cost multiplier indicated by shading. (Multiplier of 0 indicates no availability of a change option in that epoch.)

To adequately represent the epoch space in Multi-Era Analysis, proportional numbers of each type of epoch – with respect to yield and impact on changes – were chosen for inclusion in era creation. The selection included 8 epochs total: one Low- and one Very Low-yield epoch, one High- and one Very High-yield epoch each from the "no changes" epochs, and four total High- and Very High-yield epochs from the "all changes available" epochs.

6.2.5.3 Creation of Clips

Using the 8 different epochs chosen in the human-in-the-loop portion of this activity, clips comprising more frames than in the initial problem formulation can now be created. If one the four categories is sampled for each frame of an era, and each category is allowed to transition to itself, then there are only $(8*3)^2 = 576$ possible clips of length 2 frames, and $(8*3)^4 = \sim 330{,}000$ clips of length 4 frames. This activity then consists of creating 2-frame clips, 3-frame clips, and 4-frame clips by producing all possible combinations of category and duration for each length.

No other methods of era creation were performed for the present application. Random samples were generated in preparation for finding longer forms of path dependence, but most of the results mirrored the shorter-sequence clips generated here or else added significant complexity in understanding, presenting, and differentiating the results from those of the shorter sequences. As a result, only the shorter sequences are presented in this thesis.

Activity Output: Four categories of difficulty for all epochs based on yield. Creation of all possible combinations (of category and duration) of clips length 2, 3, and 4 frames.

6.2.6 Activity 6: Design-Strategy-Era Evaluation

This activity involved evaluation of each design-strategy pair of interest with the eras provided by Activity 5. The trajectories were stored in the format described in Section 5.3.3.1, with designs (and changes) paired with each frame of an era: $<X_i\text{-}\Delta_0\text{-}X_j, E, t_k>$. Since only single-arc transitions were allowed, the trajectories always comprised 5-tuples such as the one shown, with Δ = zero and $X_i = X_j$ if no change was executed in that frame.

Activity Output: All evaluated design-strategy pairs in all eras, stored in the form of their trajectory through each clip and/or era. A list was generated of clips and eras where the epoch- and era-level strategies differed in the execution of change options, helping identify sequences that exhibit path-dependence in the use of change options.

6.2.7 Activity 7: Results Analysis

As discussed in Chapter 5, there are many directions that analysis of the results could take. The present analysis will focus on a few demonstrative results, which should not be taken as comprehensive but rather examples intended to inspire additional ideas for further analysis. Rather than focus on one design, the present analysis attempts to illustrate the various levels of interesting behaviors and properties that designs and strategies can exhibit as they progress through clips and eras. Several different types of visualizations[1] are presented to help locate relevant features of the evaluated data.

6.2.7.1 Design Family Behavior

One of the first results that can help contextualize subsequent analysis is that of the design families resulting from the particular options and epochs present in the era evaluations. Recall that the naval ship tradespace did not exhibit separate "families" of designs like previous satellite case studies did (see Section 6.1.4.2). However, after the application of the epoch- and era-level strategies in all 3-frame clips, unique sets of designs begin to emerge. The results of evaluating the first 100 designs in all 3-

[1] The code for several of the visualizations used in this activity – such as the force-directed graphs, parallel sets, and parallel coordinates – can be found on http://d3js.org. As an example, the Appendix contains code from the D3 website for generating the force-directed graphs.

frame clips are shown in Figure 6-13, alongside the Full Accessibility Matrix previously generated before the era evaluations.

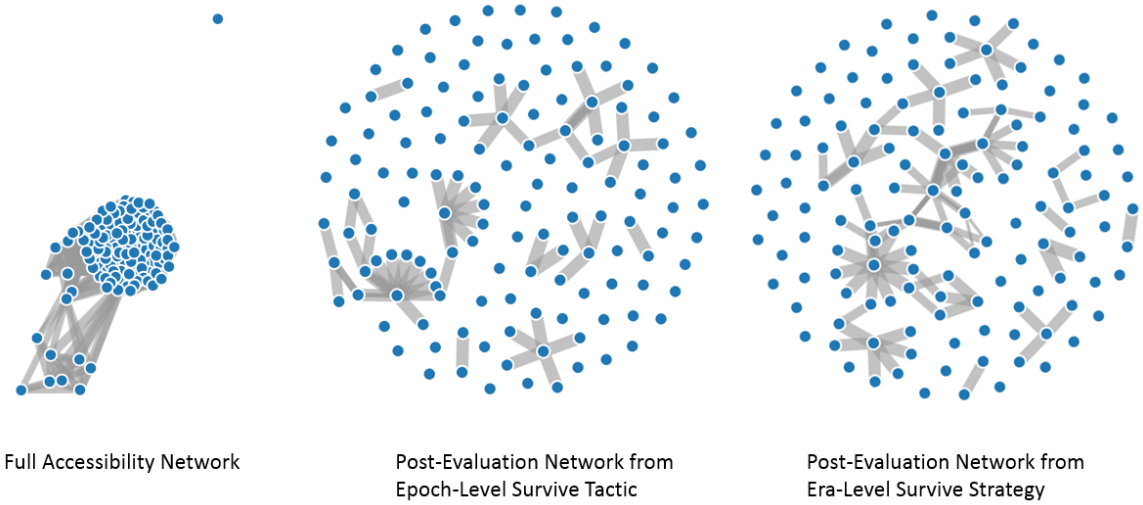


Figure 6-13. Resulting networks from the Full Accessibility calculation of Section 6.1.4.2, Epoch-level Survive tactic, and Era-level Survive strategy. (Tactic and strategy evaluated on 100 designs of the tradespace through all 3-frame clips.)

These plots show that although most designs can reach one another in the network, the strategy used and the epochs encountered strongly determine which designs rely on one another to cope with the changing circumstances throughout the short clips. The network resulting from the Survive tactic is much less connected than that of the network resulting from the Survive strategy. This difference may be due to the fact that the era-level strategy is able to anticipate changes necessary to avoid infeasibility, which provides it greater incentive to potentially use more nodes of the network in achieving minimum infeasibility over the era. Using an interactive visual tool like the force-directed graph of Figure 6-13 can allow quick identification of patterns and investigation into many other potential questions of interest. The current tool could be vastly improved by indicating the directionality of edges, as well as the particular change option(s) used to form an edge between designs.

6.2.7.2 Distributions of Design Changes through Clips

Another area of potential interest when evaluating all clips is the distribution of time spent in a given design. One simple approach to visualizing this information is shown in Figure 6-14, a bar chart of the proportion of time spent in each design through all 3-frame clips, when starting with Design 14 paired with the epoch- and era-level Survive strategies. Design 14 was selected based on the initial exploration showing that it was well connected to other designs.

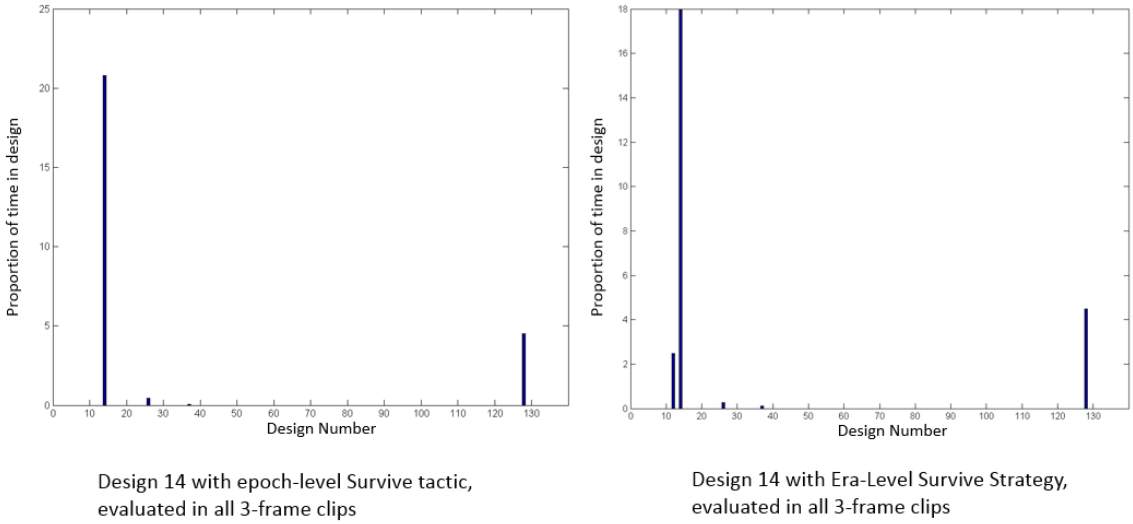


Figure 6-14. The distribution of time spent in each design, when Design 14 is paired with epoch- and era-level Survive strategies and evaluated in all 3-frame clips (around 14,000 total clips).

As the charts show, the tactic and strategy both use some common designs, such as 26, 37, and 128. However, the era-level Survive strategy also uses Design 12, which remains unused by the tactic. In addition, the strategy spends less time overall in Design 14 than the tactic does, which could indicate that there are many frames in which Design 14 is infeasible but cannot use change options (possibly due to the epoch blocking use of one or more change options). Also notice the curiously small amount of time that Design 37 is used by both the tactic and strategy, which might reflect a very specific sequence of frames in which Design 37 is very valuable to Design 14 (or one of its "siblings") in order to escape some particular episode of infeasibility. All of these observations can be used as starting points for further investigation into the nature of relationships between designs and into particular sequences of frames that are of interest due to their potential hazard or benefit for a design and/or strategy in question.

An alternate way to visualize some other aspects of this data is through an interactive visualization of parallel sets, which not only visually indicates the proportion of a design's occurrence in each frame but also the relationship between designs in all frames. A visualization of this type, based on web technologies[1], is shown in Figure 6-15. Note that the temporal element of how long each frame lasts is omitted from the diagram. (In the present case example, some frames last as little as 6 months, while some last 12 months.) This is just one example of how

[1] Code to generate this visualization may be found on http://d3js.org/.

visualizations that are particularly good at conveying certain aspects of the data can actually skew other aspects that may or may not be important depending on the question at hand. For this reason, a widely-varying visualization approach is recommended to be taken in the analysis activities, since so many aspects of the data can be presented and studied.

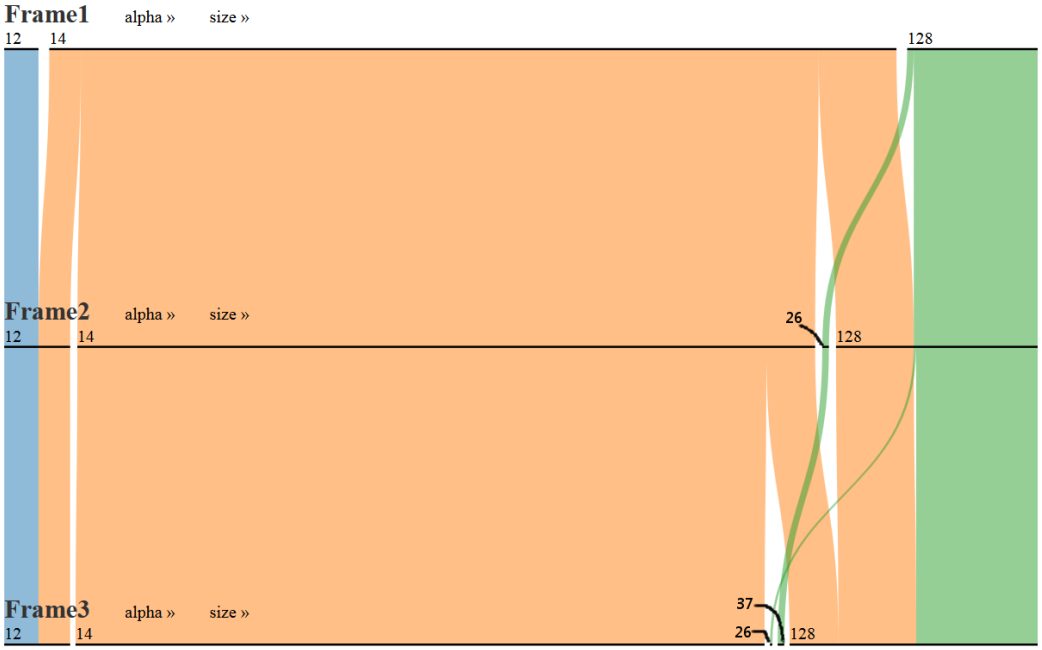


Figure 6-15. A parallel sets visualization of Design 14 paired with the era-level Survive strategy in all 14,000 clips of length 3 frames, effectively conveying different aspects of the same data as Figure 6-14. Color indicates which design number the strategy uses in frame 1 (either 12, 14, or 128), while the size of each horizontal line reflects the proportion of clips in which the corresponding design number appears in that frame.

The parallel sets visualization can be quite powerful since it simultaneously shows the breakdown of all designs in all of the frames in all of the clips (around 14,000 trajectories are displayed here). From this presentation, it is easy to observe that the Survive strategy often stays in Design 14 in the first frame – much more often than not, in fact. It is also easy to observe that Design 128 does not often change to other designs, only changing to Design 26 a very small percentage of the time. The answer to the question posed earlier regarding when Design 37 was of use is now apparent: Design 26 switches to Design 37 most of the time upon encountering the third frame of the clip. No other designs appear to use Design 37 at all. The interactive portion of this visualization adds to its ease of use, in that by hovering over any particular trajectory the user can obtain the raw number of times that the trajectory happened, as well as the percentage that number represents out of all of the data points. It is important to keep in mind that no information is presented on frame length in this diagram, which potentially misconstrues the proportional amount of time spent in

each frame. (In the present case example, however, all frames are equally either 6, 9, or 12 months due to the simple epoch duration rules modeled.)

6.2.7.3 Distributions of Metrics on Designs through All Clips

In addition to observing the behaviors of designs and strategies through the eras, it can also be constructive to apply the metrics that are of interest to the study, identified in Activity 2. In the present case, this means the MAU, MAE, and costs of changing. The parallel sets visualization presented above works well in cases where there is limited changing between values, such as in design trajectories. But when metrics such as MAU and MAE that consistently vary with each frame are plotted in a similar way, the results can be quite overwhelming. An attempt to plot this information is shown below in Figure 6-16, which even in its interactive form proves difficult to navigate and understand.

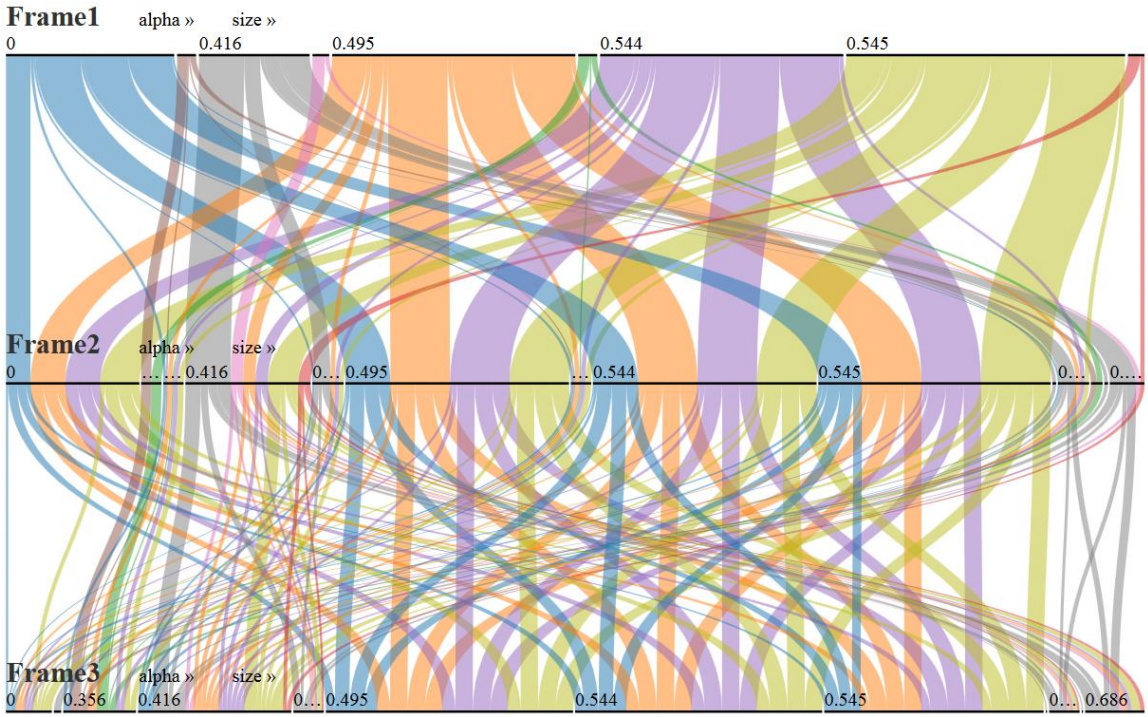


Figure 6-16. A parallel sets view of the Multi-Attribute Utility paths produced by the trajectory of Design 14 paired with the Survive strategy in all 14,000 clips of length 3 frames.

There are several factors that make this plot difficult to interpret, including the number of unique values in each frame, the lack of scale on each horizontal axis relative to the actual MAU values, and how frequently values change between frames. Even though these factors may complicate the interpretation of the plot, one interesting observation should be noted: those designs with one of the highest beginning MAU values (i.e., 0.545) have the highest proportion of zero value (a stand-in for infeasibility) in the following 2 frames. This observation may be a good

starting point for further investigation into contextual developments of interest or concern. This type of observation may take much more effort or go unnoticed with traditional statistical methods, but is easily found by experimenting with different visualizations of the data.

An alternate way to plot many paths of a metric over time is with a parallel coordinates chart[1], which more intuitively presents the numerical values on a number line for each frame. A visualization of this type built on web technologies is shown in Figure 6-17, in which darker, more solid color indicates a larger number of paths tracing the same line through the clip. This type of visualization allows interactive filtering as well, by which individual regions of each axis can be selected to only plot certain paths of interest. Because a user may interact with all of the paths at once, unusual paths may present themselves quite naturally by diverging from the central mass of the surrounding paths, and paths of interest can be easily located and separated by filtering out the paths of lesser interest. While Figure 6-17 looks perhaps less intimidating than the previous chart of MAU values, it also displays potentially much less information, since proportion of number of paths is only indicated by the opacity of each line. This observation should reinforce the idea that no one visualization is "the right one" to use for these types of activities. Rather, there are many ways to present, examine, and interpret complex data, and more comprehensive analysis can presumably lead to resultantly higher levels of comprehension of the complex problems and systems at hand.

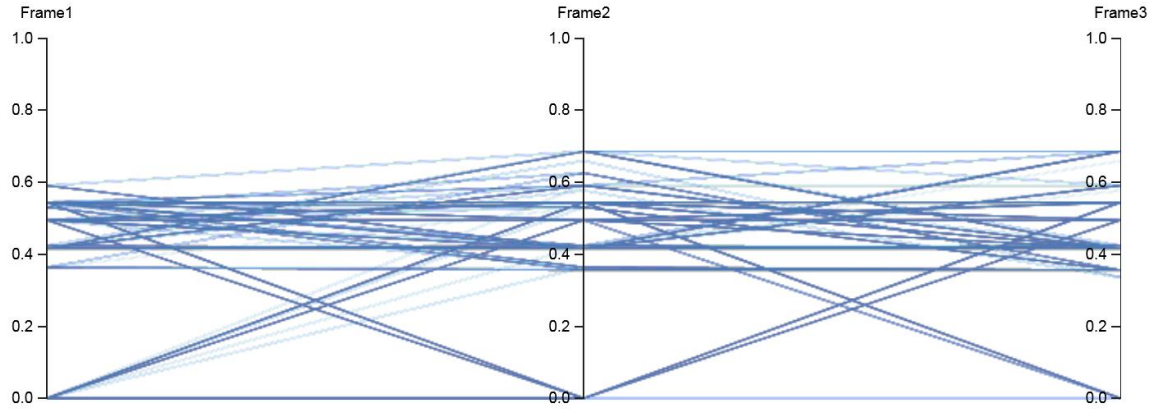


Figure 6-17. All MAU values in all 14,000 clips of 3-frame length for Design 14 paired with the Survive strategy. Darker color indicates more paths tracing that same route through the clip.

[1] Code to generate this visualization may be found on http://d3js.org/.

Two cautionary notes are in order for this visualization. The first is that the temporal element of each frame is omitted just as with the parallel sets, since omitting that aspect allows easier presentation of the data. It should be recognized, though, that this could potentially cause confusion or misunderstanding of the actual time a design spends at a particular level of a metric. The second caution is with regard to the paths as displayed: in "reality", all metrics on designs through a clip or era are piecewise constant functions, not sloped lines from one level to another (see Figure 5-3 for four examples of "actual" paths). Unfortunately, overlaying many piecewise constant functions presents its own unique challenges in effective display of the individual paths as well as the potential change in value occurring at the beginning of each frame. Both of these areas provide room for further research into better visualizing these different aspects of paths.

6.2.7.4 Usage of Change Options

An additional aspect of potentially useful information can come from examining the use of change options throughout the eras. Investigating the differing uses of options between strategies and tactics can lead to a better understanding of the relationship between individual options and the developing epochs that are encountered. For this analysis, a matrix can be created that stores the number of times a change option is used in each epoch throughout all simulated clips and eras. A Matlab visualization of the results from the Survive strategy paired with Design 14 is shown below in Figure 6-18.

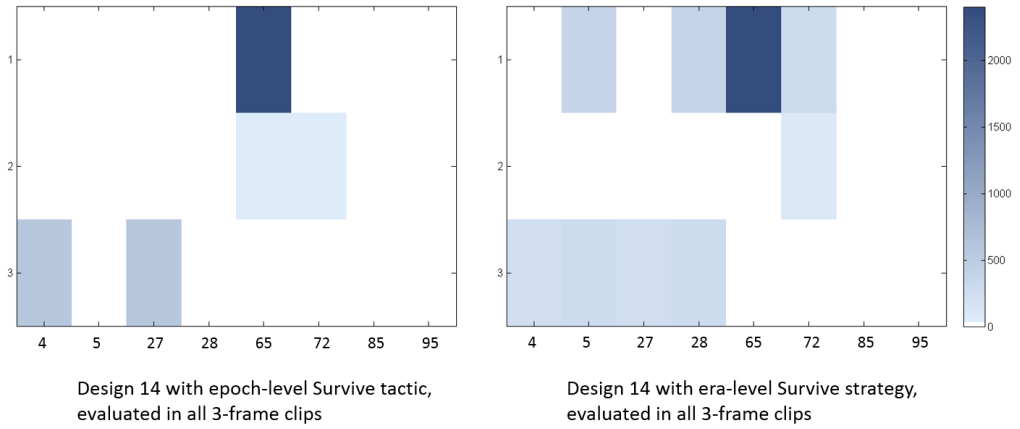


Figure 6-18. Usage of options (Y-axis: 1, 2, and 3) in all evaluated epochs (4, 5, 27, 28, 65, 72, 85, 95) by Design 14 paired with epoch- and era-level Survive strategies in all 14,000 clips of length 3 frames.

The figure shows that change option #1 (Payload Upgrade) appears to be used much more often than the others, by both the tactic and the strategy. The figure also shows that change option #2 (Add Engine) is used the least frequently over all of the clips, by both the tactic and the strategy. The use of the temporally expensive change

option #3 (Increase Length) seems to indicate that the cheaper change options must only lead to infeasible designs in those epochs. But this brings up the importance of investigating which designs are using each change option, and when they are using each change option. For this analysis, the change options can be plotted against all designs to compare the amount of usage by each design.

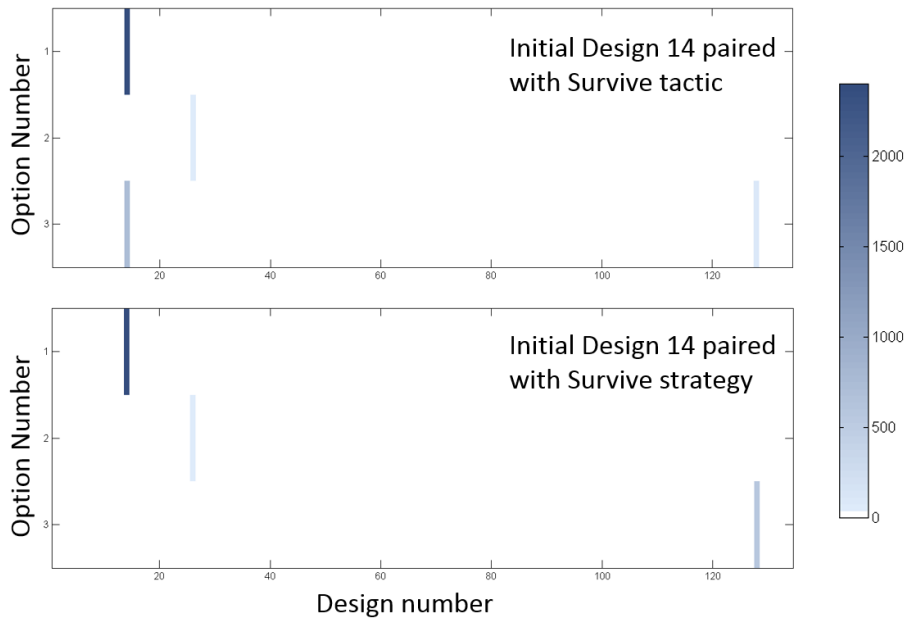


Figure 6-19. The usage of change options by each design in all 3-frame clips, from the trajectory of Design 14 paired with the epoch- and era-level Survive strategies.

When combining the information in Figure 6-19 with that of Figure 6-18 and Figure 6-15, a more holistic understanding of the designs' relationships with change options and with each other begins to form. For example, it is clear that the long-term strategy prevents Design 14 from ever having to use option #3. Instead, only Design 128 uses option #3, and only when it switches to Design 26, which it does in Epochs 4, 5, 28, and 29. In addition, Design 26 uses option #2 to switch to Design 37 whenever it encounters Epoch 72. An important note here is that these observations are the results from the output of the long-term strategy. To dig further into why the long-term strategy is using the options in this particular way, the epochs involved in only the sequences beginning with Design 128 can be plotted on a parallel sets visualization[1], shown below in Figure 6-20.

[1] Code may be found on http://d3js.org/.

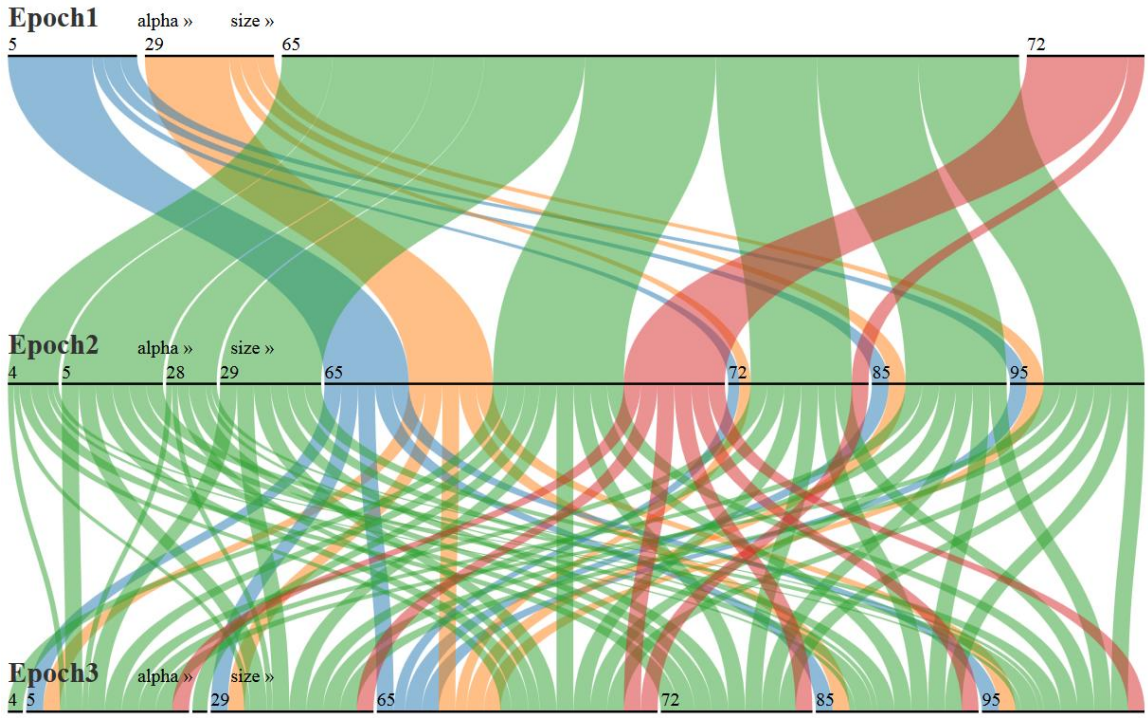


Figure 6-20. A parallel sets visualization of the epoch numbers in each frame involved in the trajectory of Design 14 paired with the Survive strategy, considering only the clips where Design 14 changes to Design 128 in the first frame. (Epoch1 corresponds to the EpochID in frame 1, etc.)

Beginning with the epochs encountered in the first frame (i.e, Epoch1 in the figure, including EpochIDs 5, 29, 65, and 72) of the clips in this set, and combining with information from the previous plots, several observations can be made. The first is that the Survive strategy for Design 14 uses option #1 to change to Design 12 some of the time, and to change to Design 128 all other times. The second is that, when Design 128 is used by the strategy in the first frame, and Epochs 4, 5, 28, or 29 in the second frame are followed by Epoch 72 in the third frame, the strategy chooses to use option #3 in the second frame to change to Design 26 and then option #2 in the third frame to change to Design 37. For this particular sequence of frames, then, the minimally infeasible trajectory utilizes two additional options that it otherwise does not use when beginning with Design 14. To conclude that options #2 and #3 are somehow less essential because they are not used in as many futures as option #1 is ignores the fact that each of these trajectories is a plausible future, not "noise" that deviates from some average. Only one of these sequences (and resulting trajectories) will actually occur, and in the case of building more than one copy of a system (such as building many copies of the NGCS-like naval ship), *all* of the ships encounter the same epoch developments. If options #2 and #3 were not included in the system design, then *all* of the ships may become infeasible in one of these particular epoch sequences – there would be no average case, only failure of all.

6.2.7.5 Activity Summary

Hopefully it is clear that the analysis at this stage could take countless directions. Simply presenting the results in a somewhat concise manner can take significant effort, since there are so many aspects of the data that can be included or excluded depending on the goals of the analysis. There is no single way to sum the results, obtain a numerical value, or otherwise even display the data; much less is there a "best" measure or visualization to use. Since every visualization technique is good at presenting some aspects of data and poor at presenting other aspects, it is intuitive that many different types of visualizations should be used in this process to foster as many questions and observations as possible in the course of the analysis. By interacting with the results in this sort of immersive experience, analysts can begin to form a clearer understanding of the ramifications of decisions made earlier on, whether in the modeling and analysis of the first 8 process of RMACS, or in the earlier activities of MERA. The goals elicited in Activity 1, of course, should be used to give a clear direction during this activity, since otherwise much effort can be spent for relatively little return of relevant information.

It bears repeating here that the goal of the exploratory analysis in this last activity is not to determine a single "optimal" solution (Bankes, 1993). Nor is it to design for an average or worst-case future (Abbass et al., 2008; Simon, 1996). Rather, the goal is immersive experience (Schön, 1993), which can lead to deeper insights, consistent communication, and better mental and constructed models with regard to the many aspects presented by a complex problem and its potentially complex solutions.

Activity Output: Demonstrative plots and visualizations on several levels of analysis, including network depictions, frequency of design occurrence in frames, interactive visualization of design trajectories through all evaluated clips, metric paths through all evaluated clips, and preliminary analysis of the use of change options throughout various epochs for one particular design-strategy pair.

7 Discussion

"For even the very wise cannot see all ends."

-Gandalf the Grey (Tolkien, 1974)

7.1 Discussion of Process 9, Multi-Era Analysis

The results of Chapter 6 demonstrate a very small portion of the analysis that can be performed on the results of simulating many designs through many possible clips and eras. Almost any inquiry into a system's behavior can be answered by the data produced in Multi-Era Analysis, and much more data can result from the practically infinite number of remaining possible simulations (regardless of how many eras are ever actually simulated). These abilities raise at least two important questions.

First, what are the aspects of most concern in the design of the present system? The many choices made in the modeling efforts of previous processes in RMACS reflect the answers to this question, and can be used in the determination of which details to record and which further data to generate. The first activity of MERA aims to address this question, since the aspects of concern directly influence all following activities of metric creation, strategy creation, era creation, and design-era evaluation. It is highly probable that some of these latter activities will inspire additional questions and aspects of concern, which can be incorporated through feedback to later iterations through the process.

The second question, equally important as the first, is: What are the stopping conditions for the analysis? Quantitative requirements on confidence levels are generally unsuitable to answer this question, considering the size of the era space and the types of uncertainty that the analysis is attempting to address. As stressed throughout the present work, modeling complex future systems is not an exercise in optimization, but rather an exercise in thinking through, planning, and formalizing a human approach to a complex problem (Abbass et al., 2008; Bankes, 1993; Sarewitz & Pielke Jr, 1999; Simon, 1996). This central point, of course, leads to the idea that a human must be involved in making the judgment as to when "better than nothing" is "good enough". On the surface this answer may seem somehow less rigorous than a contrived, advanced technique for assigning numerical values to the quantity and quality of products from Multi-Era Analysis; but it is nonetheless ubiquitous practice at all levels of decision-making in open systems. Animals, humans, and organizations all have some budget of resources (whether temporal or physical) for gathering information and analyzing it, and at some point they are

required to act on the limited (with respect to all available) information that they have gathered and analyzed. It should not be surprising, then, that the stopping criterion for a comprehensive study of all possible relevant futures exhibits this property.

7.2 Discussion of the RSC-based Method for Affordable Concept Selection (RMACS)

As proposed in Chapter 3, the nine processes of RMACS aim to address the need for methods and metrics for early-lifecycle conceptual design of affordable systems. By modifying the original Responsive Systems Comparison (RSC) through the incorporation of the Multi-Attribute Expense function, metrics for affordability, and Multi-Era Analysis, RMACS is able to identify properties of system concepts that can enable affordable solutions to the complex problems facing many organizations and decision makers today. Though designed to work with fully enumerated tradespaces, the application of the first 8 processes of RMACS demonstrated its ability to bring insight into a study of only a handful of point designs. The final process was demonstrated on a much larger dataset with change options to show a small portion of the comprehensive analysis that RMACS with Multi-Era Analysis can enable for a full tradespace study.

7.2.1 The Importance of Feedback and Iteration

Though all of the processes of RMACS were presented in a linear fashion, the feedback from each process can be a valuable source of modifications to future iterations. As with the original RSC method, each process is designed to accommodate varying levels of depths in the inputs and activities presented. Since the purpose of these design methods is primarily exploratory, some of the key results can be the "Aha!" moments that analysts and decision-makers have when presented with unexpected results of a process. Iterating back through previous processes and incorporating the knowledge gained can greatly enhance the clarity and relevance of later results.

7.2.2 Integration with Existing Practices

One of the questions that invariably arises with a new design method is the question of how to integrate the new method into existing practice. In the case of RMACS, such integration is presumably not that difficult, since tradespace exploration has been growing in its use in various governmental and commercial organizations in recent years, albeit without the use of attributes-and-utility as a constructed value model. In addition, system models and simulations are becoming more interoperable due to DoD standards and due to the growing popularity of Model-

Based Systems Engineering (MBSE). One of the major challenges that remains is the characterization of epochs and eras in addition to (and sometimes instead of) the standard models of uncertainty that are used in the performance models of existing systems. The process of eliciting the epoch variables is relatively straightforward, only requiring a small team of analysts and experts. Baseline performance models can be incorporated relatively easily into a "baseline" epoch, with modifications needed for each epoch variable that has been elicited. Unfortunately, though, it is often the case that legacy models and simulations have probabilistic assumptions "cooked in" from the lowest levels to try to characterize future uncertainty. For example, in a simulation of a naval ship's performance in an open-water skirmish, a simulated ship could have a high chance of successful attack due to the current model of an adversary's capability. But when epochs are introduced to an existing trade study, there may need to be multiple models of the adversary, one for each level of an epoch variable that represents the adversary's capabilities. In the simulation, chance of a successful attack would still be determined by the model of the adversary, but which adversary model to use would depend directly on the epoch variable level during the simulation. In many existing-use models such as this example, modifications to these low-level factors would need to be carefully considered in order to accommodate the varied conditions and changes that the creation of epochs may impose on such probabilistic assumptions.

7.3 Future Research

It should be clear from Chapters 5 and 6 that there are practically limitless areas for future research in the newly introduced process of Multi-Era Analysis. These areas include identification of additional types of relevant path dependencies, types of era-level metrics, modeling of change options with finite number of uses, creation of era-level strategies (i.e., path planning) for continuous-range change options as well as mixed discrete-and-continuous-range change options, methods for analysis of families of designs and epochs, tools for effective human-in-the-loop creation of eras, culling techniques for the era space, sampling techniques for the era space, and statistical and visually interactive methods for navigating the outputs of the process, and creation of autonomous recommendation agents for strategic use of change options. Each of these areas is now briefly described.

7.3.1 Identification of Additional Types of Relevant Path Dependence

As Chapter 5 describes, the goals of Multi-Era Analysis include identification of path dependence in the developing epochs and changing designs over time. Only two aspects of relevant design-level, change-related path dependencies were identified

for the present work. The first was that of the differing use of change options by epoch-level strategies (optimizing over single frames) and era-level strategies (optimizing over the system lifetime). The second was that of the sequential use of change options in a limited set of future trajectories discovered in the results of Section 6.2.7. The algorithmic implementation of tactics and strategies allowed automatic locating of the first aspect of design-level path dependence in simulated eras. Interactive visualization allowed discovery of the second type. It is certainly the case that additional types of relevant design-level path dependence exist, with or without the inclusion of change options. For example, perturbations may also strongly affect a design's evaluated performance throughout an era, depending on when they happen, as (Richards, 2009) demonstrated. Other era- and meta-metrics could certainly be used to find and describe other design-level considerations.

Epoch-level path dependencies were not demonstrated in the present study, but the relationships between epochs modeled can provide key insights into how the unfolding future maps to the behavior of designs over time. Many of the design-level metrics used in Multi-Epoch Analysis may take on quite different values when applied to eras constructed from highly path-dependent epoch transitions. Future research could look into identifying relevant epoch path dependencies by examining different epoch variables and/or their impacts on the attributes of the systems being modeled.

It is essential to note here that the term "path dependence" is somewhat broad, since every era demonstrates path-dependence by the very definition of sequencing epochs and durations together. This inherent vagueness is the reason for this thesis's use of the word "relevant" or phrase "suitably interesting". It is understood that what is relevant or suitably interesting path dependence to one system (or stakeholder) may not be relevant to another system (or stakeholder). Further work could attempt to classify the various types and aspects of path dependencies that arise in era-level studies, potentially suggesting conditions under which a particular type becomes relevant, as well as effective methods for locating that particular type throughout many eras (e.g., application of metrics, algorithmically, visually interactively).

7.3.2 Era-Level Metrics

As discussed in Section 5.3.3.4, stakeholders and decision makers may have preferences on temporal aspects of various system metrics over an era. The meta-metrics proposed in that section aim to begin capturing the short- and long-term considerations related to many of the metrics used throughout the RMACS method. The list is by no means intended to be exhaustive. Future research could identify useful metrics from financial and statistical analyses in order to leverage a large

amount of existing literature and practice on time-series analysis. Unfortunately, many such measurements encountered in the course of the present work require a number of assumptions to be made, including any of the following: 1) continuous functions and/or non-zero variance, 2) large number of sample points, 3) underlying static probability distributions, 4) ratio measurements on universally-ranged scales, 5) linear relationships, 6) large amounts of time available to analyze one time series, and/or 7) goodness-of-fit metrics needed for a single application. These assumptions work well in cases where empirical data from one long development of a single metric is of interest, but they tend to poorly translate into the realm of various system design metrics over many thousands (or millions) of potential future developments. Modification and demonstrated application of established time-series analysis methods could be a profitable endeavor, though, since it would allow much existing research to be leveraged in analyzing the temporal aspects of designs through many eras.

7.3.3 Creation of Era-Level Strategies for Continuous-Range Change Options

The strategy implementation of the present work was limited to discrete change options on a tradespace network. A beginning approach to implementation of an era-level strategy for continuous-range change options was proposed in 5.3.3.2, with the key challenge being that of modifying the nearest-neighbor algorithm to incorporate the particular strategy, preferences, and constraints on change option usage. Future research could potentially demonstrate such a viable path planner (whether or not with an RRT-variant algorithm) for the continuous-range space to enable analysis of change options without an accompanying tradespace network. This would likely require some sort of tradespace approximation technique, construction of which requires multiple configurations of a design to be evaluated in each enumerated epoch (the interested reader is directed to (Fulcoly, 2012) for an application of such approximation techniques in a tradespace). Once constructed, the approximation model would then need to be linked in some way to the path planning algorithm, so that the path planner could obtain computationally cheap estimates of whichever metrics are used in the calculation of the nearest neighbor.

If tradespace approximation techniques are used, an alternate approach might be to simply approximate several levels of a continuous-range change option, thereby creating a type of tradespace network for each epoch. Since this would result in a discrete tradespace network, a network search algorithm such as A* could then be used to navigate the approximated network in the same manner as the era-level strategy implementation of Chapters 5 and 6 of this thesis. One of the benefits to doing so would be potentially simplified combination of continuous-range change

options with discrete change options, since it may be straightforward to combine both types of networks resulting from the differing change options. One of the drawbacks, of course, is the potentially much larger sizes of the resulting networks to store and search. Combining continuous-range and discrete change options presents a unique challenge but is a valuable area for further exploration.

7.3.4 Methods for Analysis of Design Families and Epoch Networks

The study of design and epoch networks is an area ripe for formalization and future research into leveraging methods and metrics from existing network analysis literature. Much existing research in general networks assumes fully connected graphs, equality of nodes, and/or goals specific to electronic communication over nodes and links, but many of the concepts could potentially be brought over to the analysis of design and epoch networks and families.

7.3.4.1 Design Families

As demonstrated in Section 6.1.4.2, the idea of general design families can be a useful concept in describing groups of designs that are accessible by change options, without requiring a major redesign. The analysis and comparison of these groups could provide an alternate approach to a changeability-enabled trade study: instead of evaluating which individual design is "best" when paired with a strategy and options, design families could be evaluated based on their ability to succeed in many possible futures. The change options linking these design families could also be compared in this way, as it could be the case that particular families depend largely on one or more particular change option(s). By evaluating entire design families over only a few frames, the patterns of change could be identified and metrics could be evaluated to help better extrapolate families' behavior in the long run. In addition, the Full Accessibility network could be compared with the resulting network from the evaluations of a strategy in many clips and eras, providing information on which "sub networks" of a family may be of most interest.

7.3.4.2 Epoch Networks

Another area of analysis that remains relatively unexplored is that of the networks created through the modeling of epoch variable levels that have duration and transition constraints. (Fulcoly, 2012) began some of this activity with the Epoch Syncopation Framework, with the goal of linking evolvability-related changes to somewhat reliable knowledge of the rate of future changes in context and needs. There are many more levels of analysis that could be performed on epoch networks, including the analysis of densely vs. sparsely connected regions, as well as potential identification of epochs that exhibit a high level of "betweeness" (Freeman, 1977),

connecting otherwise disparate futures. Interactive network graphs such as those presented in Section 6.1.4.2 could be one method for beginning this type of analysis, but it is certainly possible that other established research in network flow could be brought over, allowing some type of characterization of the era space for a given study. This result in turn could lead to more effective means of era creation, era space culling, and era space sampling.

7.3.5 Interactive Visual Tool for Effective Human-In-The-Loop Era Creation

One of the central portions of Multi-Era Analysis is Activity 5, Era Creation, in which eras of interest are selected to include in the study. Several methods for this activity are proposed in Section 5.4.5, including a human-in-the-loop software tool for constructing longer sequences of frames. At present, it is envisioned that such a tool would include components for the analysis of designs, epochs, and frames and clips, as well as a component for accepting as input any expertly-generated eras and using those as potential starting points for suggested eras of consideration.

On the design level of analysis, information of interest could include design variable levels, attributes and metrics, change options, networks and design families. On the epoch level, the tool would have components for analyzing epoch yields, epoch descriptor impacts on design attributes, epoch impacts on the use of change options, and epoch network nodes and edges, among other information. On the frame and clip levels, information of interest could include evaluated designs in frames and clips, including metrics, meta-metrics, and change option usage.

In addition to these analyses components, the tool should allow the input of expertly-generated eras, as in the RSC application to SRS (Ross et al., 2008, 2009), and provide analysts with helpful means to use such eras in generating many variants based on those described by experts. This component could incorporate many of the design- and epoch-level considerations, and potentially provide some sort of automatic recommendation system based on characteristics of the expertly-generated eras and the structure of the design and epoch networks. Through this type of approach, an interactively visual tool would enable the natural abilities of human judgment and pattern recognition to be harnessed in determining which scenarios to consider for a given system.

7.3.6 Pruning the Era Trees for Era Creation

The unwieldy size of the era space most likely removes the possibility of enumerating a large fraction of the potential futures for a given system. One of the ways to potentially deal with this problem is the pruning of the network that results from frames transitioning to one another. One form of pruning was demonstrated

in the NGCS-like application, which selected 8 representative epochs (out of the 96 available) based on their estimated difficulty (determined from an epoch's yield and its impact on the execution of change options). Unfortunately, due to the discrete nature of the space and the subjective nature of the relevant characteristics, there are no guarantees on whether important information could have been omitted along with the excluded epochs. Preemptive pruning was also conducted by limiting the durations allowed for each frame. While such pruning is likely necessary to conduct any meaningful foray into the era tree, methods for efficient and benign pruning remain elusive at present. Further research could examine specific cases where a specific pruning method demonstrably cuts out information of interest, including identification of the type of information desired and the features involved, such as: change options that are modeled to be discrete or continuous, durations of a particular discretization, epoch-design interactions with respect to some metric, and many other possible features.

7.3.7 Sampling Techniques for Era Creation

Another immediate response to the vast number of eras is the idea of sampling the space. Unfortunately, sampling combinatorial spaces for specific properties proves to be challenging in cases such as sampling solutions to Boolean satisfiability problems (Erenrich & Selman, 2003; Lin & Iii, 2012). However, if the goal is not to characterize the space, but rather only augment the other approaches to era creation, a beginning approach to sampling the era space may simply consider it as an *information gathering problem*, in which functions such as coverage, entropy and mutual information might be applied. These functions all exhibit the property of submodularity, which essentially means that there are diminishing returns in change of function value for each additional sample. This property leads to the prospect of applying proven, efficient approximation algorithms to determine a near-optimal set of eras to sample, given a cost to obtain each sample and some budget of finite resources. As a beginning step, this approach would likely cover as many unique combinations of frames as possible before depleting the budget of resources. The implementation may be more challenging (and potentially more informative) in problems where there exist tightly constrained epoch relationships, meaning that certain epochs and frames are prevented or alternatively happen more often. Much further research could be done in this area, possibly incorporating the results into the interactive visual tool discussed in Section 7.3.5.

7.3.8 Prescriptive Information from Autonomous Planner

As demonstrated in chapters 5 and 6, when change options are included in the MERA study, the concept of era-level strategies emerges. An era-level strategy Π is an *offline* decision-making algorithm, since it takes as input an entire era, which gives the strategy complete knowledge of the future on which it decides optimal actions. The actions chosen by the strategy (and the resulting trajectories of designs through the eras) aid in generating information in the Descriptive portion of MERA.

But what about the case when a X finds itself in an epoch with an unknown future ahead of it? Which change options, if any, should it implement or execute *right now, based on the past and present*? This type of decision planner based only on past and present information is known as an *online* algorithm, with online era-level strategies represented here by Π^*. Certainly such a planner can reference the knowledge generated from the Descriptive portion of the MERA study, as the offline results of simulated tactics and strategies provide valuable information on potential "dead-ends" or "traps" in the execution of change mechanisms. If the system and epoch models are updated as the phases of system development progress, an online planner could be built that leverages such information through each phase of system development. In this way, system changes in operational phases could be recommended by an advanced autonomous planner that continually monitors the current state of the world, and the planner itself would represent the accumulation of information tracing back to the conceptual development of the system.

The creation of such an online planner would have several stages. First, priorities and constraints would be set by an analyst to specify the conditions that are preferred/allowable and those that are not. The implementation of these conditions would be analogous to setting up the conditions of the era-level strategies of Section 5.3.3.2. The conditions could include Boolean criteria (e.g. feasibility/infeasibility), efficiency criteria (e.g. FPN), or real-valued attributes (Speed, Ops Cost), and the criteria could consider the short-term (e.g. do not become infeasible in the present epoch) or long-term in its evaluation. Once these priorities and constraints are set, the online planner can be constructed, incorporating some scoring method for the priorities and constraints, as well as incorporating all previously generated Descriptive data from MERA as a reference dataset for the planner. Then the planner would participate in either supervised or unsupervised learning, simulating thousands or millions of eras, and storing its performance with respect to the "scored" priorities/constraints. Once the planner has learned to a satisfactory level (though it never needs to "stop" learning), it may then be referenced for any specific combination of epoch developments in an era, prescribing the optimal current

action Π^*_k with respect to past events, all simulated possible futures, all simulated possible future trajectories, and the priorities and constraints defined by the analyst, stakeholders, and decision maker. For more information on this topic, the interested reader is directed to the examples of TD-Gammon (Tesauro, 1995) and Deep Blue (Campbell, Hoane Jr., & Hsu, 2002).

8 Conclusion

> "…when the results of their labours, even when they are not nugatory, tend to take unpredicted, often catastrophic directions. Better to use in each context the methods that seem to fit it best, that give the (pragmatically) best results; to resist the temptations of Procrustes…"
>
> -Isaiah Berlin, *The Hedgehog and the Fox*

As stated in the introduction, this research was motivated by three related factors, including the state of the practice of systems engineering in increasingly costly defense systems, the current downward budgetary trends of the DoD, and the state of the art in systems engineering research for early-lifecycle, conceptual development of system design. The confluence of these three factors resulted in the main research questions surrounding an early-phase design method and analysis for changeable systems in many possible futures. Each of these questions and their answers are presently reviewed.

8.1 Answer to Research Question #1

The first research question was:

1) Can the affordability of complex systems be better enabled through an early-phase design method that incorporates Multi-Attribute Tradespace Exploration and Epoch-Era Analysis?

The answer is affirmative, as demonstrated by the development of RMACS and its subsequent application to the NGCS naval system. RMACS extends the original RSC method through the incorporation of the Multi-Attribute Expense (MAE) function, affordability metrics such as Max Expense and Expense Stability, and the addition of Multi-Era Analysis (MERA) as the ninth process. The MAE metric helps mitigate the perceptual complexity of preferences on resource consumption, and those preferences then define the conditions in which a design can be considered feasible (and therefore affordable). The Max Expense and Expense Stability metrics help capture additional short- and long-run considerations of the usage levels of specific resources. When MERA is included in the method's application, it can yield insights into path-dependent relationships among designs, epochs, change options, and strategies, and the insights gained can help establish the patterns of more affordable designs as well as strategies and change options that enable designs to be more affordable throughout their lifecycle. The overview of the RMACS method was described in the content of Section 3.3, and is shown below in Figure 8-1.

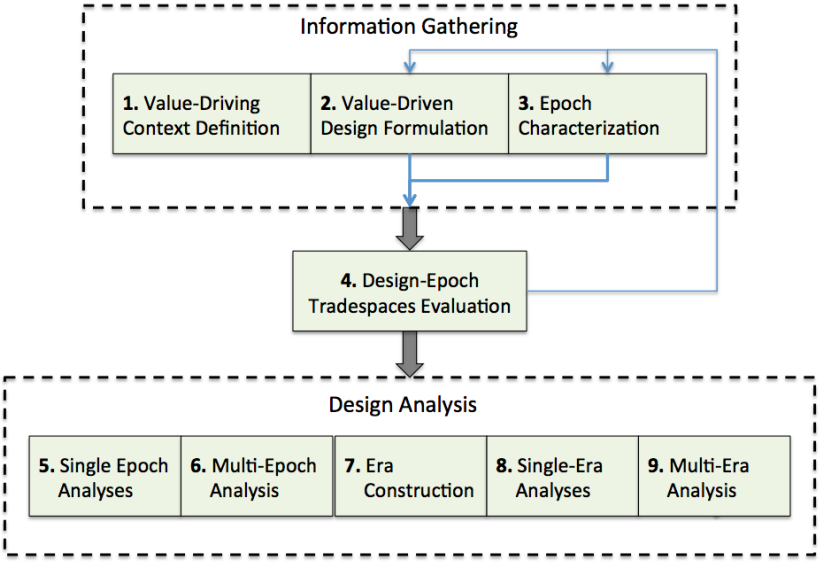


Figure 8-1. An overview of the RMACS method.

8.2 Answer to Research Question #2

The second research question was:

2) Can the affordability of changeable systems be better enabled by leveraging existing path planning approaches to efficiently evaluate long-term change strategies for systems in many possible futures?

The answer to this question is affirmative, as demonstrated by the implementation of the informed-search A* algorithm in the NGCS-like design study, providing the ability to automate changes guided by era-level strategies in many possible eras. One example result from this implementation is shown below in Figure 8-2, which shows a design trajectory (and resulting MAU path) produced by an A*-based era-level Survive strategy.

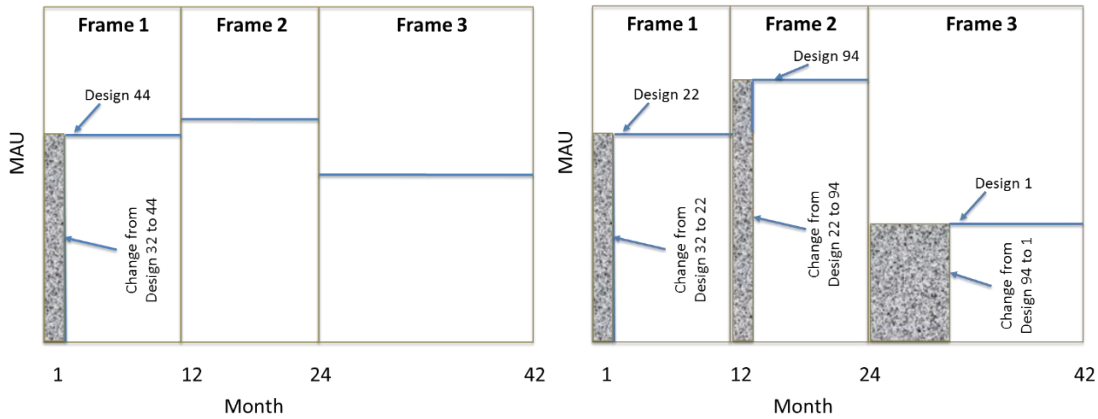


Figure 8-2. Design trajectory resulting from the A*-based era-level Survive strategy implementation (left), compared with the results from the epoch-level Survive strategy in the same era (right).

An affirmative answer to question #2 is also demonstrated by the implementation of Djikstra's algorithm to efficiently create the Full Accessibility Matrix, resulting in the Fully Accessible Outdegree (FAO) and the Fully Accessible Filtered Outdegree (FAFO) metrics, which both reflect potential long-term effects of change. Example network visualizations resulting from Djikstra's algorithm are shown below in Figure 8-3.

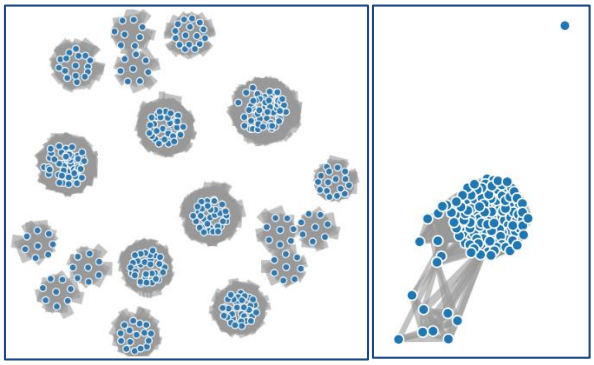


Figure 8-3. Full Accessibility network visualizations resulting from the implementation of Djikstra's algorithm to include all arcs (Left: Space Tug design study, Right: NGCS-like case study).

8.3 Answer to Research Question #3

The third research question, which was more open-ended than the first two, was:

3) How can we better capture the considerations related to long-term, path-dependent properties of changeable systems in dynamic operating environments?

The answering of the open nature of this question is begun by the activities included as part of RMACS' Process 9, Multi-Era Analysis. This process potentially includes the creation of *meta-metrics* to help measure Metrics of Interest (MOIs), such as system resource usage and other relevant metrics – and the temporal development of each – over the entire system lifecycle. The example meta-metrics given by the study are shown below in Table 8-1 and Figure 8-4.

Table 8-1. Six example meta-metrics resulting from the current work.

META-METRIC	FIGURE 5-3 (a)	FIGURE 5-3 (b)	FIGURE 5-3 (c)	FIGURE 5-3 (d)
EXPEDIENCE	0.8	0.2	0.45	0.5
VARIABILITY	0.95	0.95	1.2	0
AVERAGE	0.4	0.4	0.4	0.4
GREATEST RISE	0	0.55	0.4	0
GREATEST FALL	0.55	0	0.6	0
RANGE	0.95	0.95	0.6	0

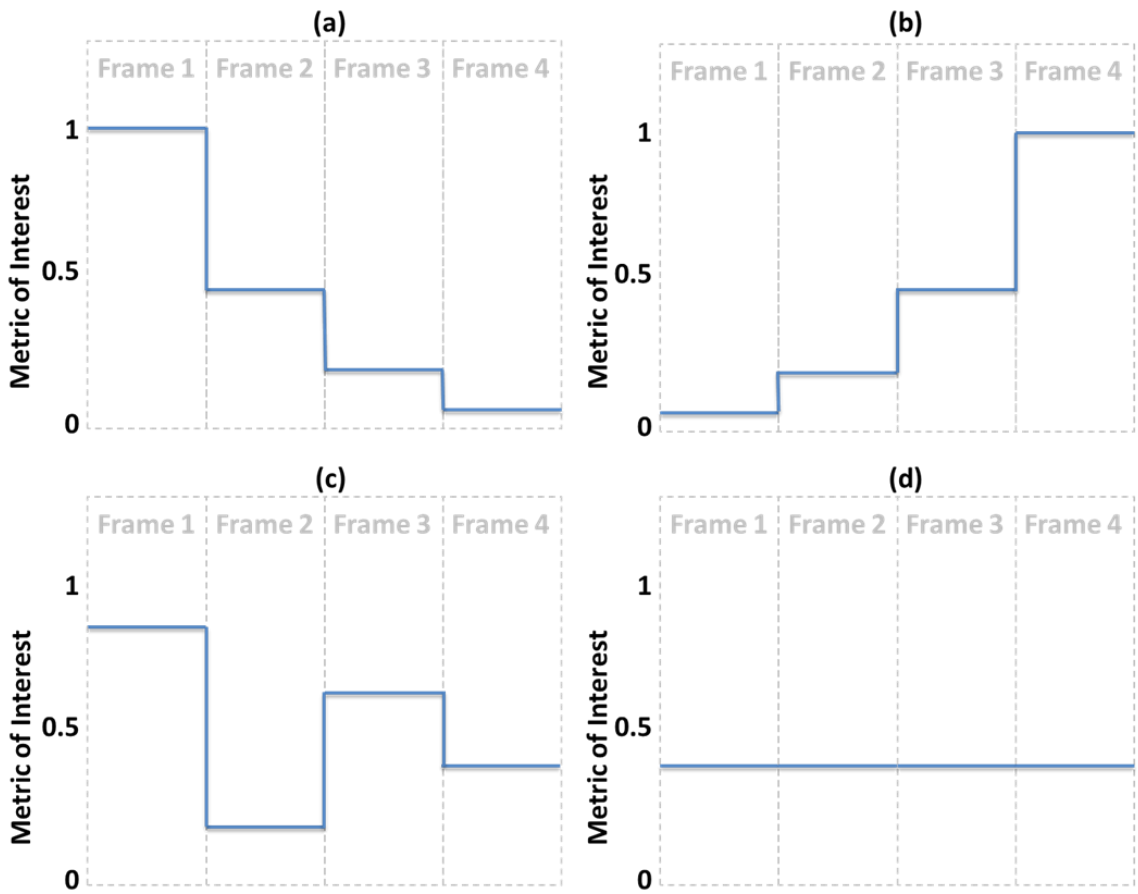


Figure 8-4. The four example paths from Figure 5-3 of the development of a MOI over time.

The answer to this third research question also included the definition and demonstration of *era-level strategies* to enhance the understanding, communication, and expectations of stakeholders and analysts in design studies of changeable systems. An era-level strategy can be minimally defined as minimization

or maximization of an objective function, or it may include an arbitrary number of constraints, including constraints on metrics within an epoch, or era-level metrics that include considerations for stakeholder preferences on metrics over time. An example of an era-level strategy created in the present work is:

$$\begin{aligned} \min \quad & \sum_{k=1}^{n} Infeasible(\psi_k) \\ \text{s.t.} \quad & 1)\ \psi^X{}_k = \psi^X{}_{k-1} \quad \text{or} \quad < \psi^{X_i}{}_k,\ \psi^{X_j}{}_k > \in \boldsymbol{\Delta}, \\ & 2)\ \text{GreatestInstantRise[OpsCost}(\boldsymbol{\psi})] < \$10 \text{ million/yr} \\ & 3)\ 0.3 < \text{Expedience[OpsCost}(\boldsymbol{\psi})] < 0.7 \\ & 4)\ \text{Range[ChangeCost}(\boldsymbol{\psi})] < \$30 \text{ million} \end{aligned}$$

When era-level strategies are created, they can be implemented as demonstrated by the informed-search A* algorithm applied in the NGCS-like design study, providing the ability to automate changes guided by era-level strategies in many possible eras. An example result from this activity is shown above in Figure 8-2.

The answer to this third question also includes the identification of *methods for creating eras of interest* that provide information on relevant path dependencies. The first method proposed is the construction of a human-in-the-loop tool that leverages interactive visualization and established methods of analysis to provide analysts with the information necessary for creation of longer sequences of frames. The second method proposed is that of Breadth-First Search through clips, or short sequences of frames representing portions of eras. This idea is depicted in Figure 8-5 below. The final method proposed is that of sampling, which may be useful when combined with the other two methods.

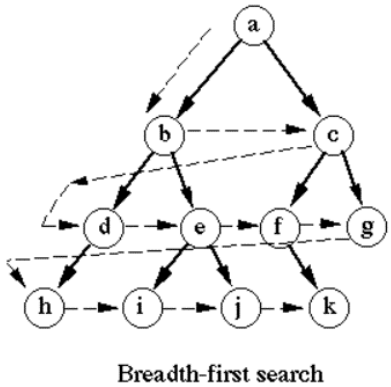


Figure 8-5. One of the methods proposed and demonstrated for era creation in Multi-Era Analysis.

Finally, the answer to the third research question included *the use of interactive visualizations for results analysis*, for the analysis of design and epoch networks pre- and post-evaluation, as well as parallel coordinates and parallel sets demonstrated

in the last activity of Process 9 in Multi-Era Analysis. Some of the visualizations used[1] in this activity are shown below in Figure 8-6.

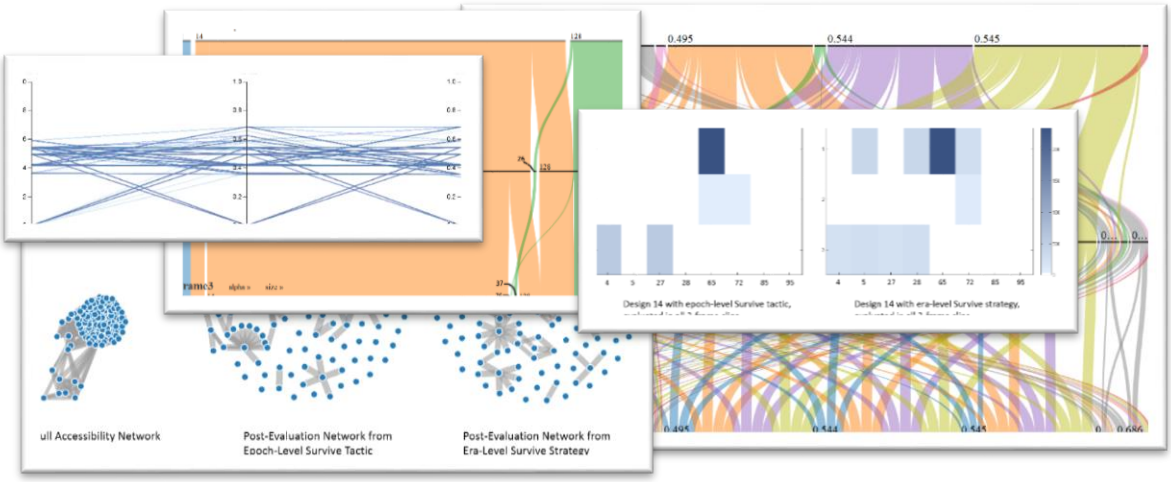


Figure 8-6. Some of the visualizations used in the Results Analysis portion of Multi-Era Analysis.

As noted in Section 6.2.7, each type of visualization captures different aspects of the results produced by the Design-Strategy-Era Evaluation activity of Multi-Era Analysis. Some of the visualizations are more helpful for locating specific forms of path dependencies, and some of the visualizations help contextualize the behavior of designs and strategies throughout the clips and eras. Each visualization has its strength and weaknesses, and none are able (or intended) to give "the whole picture". They are all intended to augment the other approaches to *exploratory modeling*, which is at the heart of the entire RMACS method.

8.4 On the Challenges and Benefits Associated with Discretization of Futures

As is the case with visualizations, in general every tool at once enhances some ability while limiting some other ability. The first 8 processes of the RMACS method demonstrate how the analytical tools of Multi-Attribute Tradespace Exploration and Epoch-Era Analysis provide a much-enhanced ability to capture many of the aspects of complexity inherent in the design of modern, large-scale systems. The 9th process of Multi-Era Analysis may indicate the limitations imposed by these analytical tools – that is, the computational and other difficulties of comprehensive analysis of sufficiently many long-run, discretized futures. For practical purposes in modern

[1] For a description of the software tools used to generate each visualization, see Section 6.2.7.

scenario planning, however, these limitations are not very limiting, since accurate prediction of long-term future scenarios is not possible regardless of the type of analytical tools used. Modeling the future as nicely distributed continuous random variables simply for computational ease of scenario planning cannot help stakeholders and analysts better understand the perceptual complexities associated with the value delivery of systems over time. Nor can it help them better understand and communicate their own mental models of and approaches to mitigating the uncertainties of contextual variables and potential changes in preferences. When these latter purposes of exploratory modeling (as Section 3.1 describes in detail) are brought into consideration, the true benefits of the analysis of discrete futures can be fully appreciated.

8.5 Final Thoughts

The rabbit hole goes deep. Modern system architects face uncertainties, complexities, and constraints – with respect to cost, schedule, and organizational factors – at almost every level of system design, especially in the early phases of system engineering for large-scale, complex systems. And yet cause for optimism remains with respect to incremental improvement on past and existing practices. Glimmers of hope are provided by the exploratory constructs and methods prescribed by the state of the art in systems engineering. These exploratory tools allow a clearer perception of the complex domains that motivate the creation of modern systems, enhance the understanding of value-delivery considerations, and provide a basis for clear, consistent communication among stakeholders, analysts, and decision makers. The way forward is seen as yet through a dark glass, but the research presented here sheds some additional light on the path. All that remains is to take the first steps in the transformation of current practice. There are, perhaps, no more fitting words than Winston Churchill's:

> "Now this is not the end. It is not even the beginning
> of the end. But it is, perhaps, the end of the beginning."

Glossary and Acronyms

A*: A best-first network search algorithm that uses a heuristic to estimate the distance to the goal, thereby exploring more promising paths first. See Search Algorithms for Discrete Networks, for more information.

Affordability: In this thesis, affordability is defined as "The property of becoming or remaining feasible relative to resource needs and resource constraints over time" (Wu et al., 2014).

BFS: Breadth-First Search, a "blind" or uninformed network search algorithm. See Search Algorithms for Discrete Networks, for more information.

Clip[1]: A subset of an era, a clip is an ordered sequence of a small number of *frames*, where "small" is relative to the total number of epochs in an era. Whereas eras reflect timescales on the order of a system's lifecycle, clips only capture a portion of the lifecycle's future. Clips can be used to reduce the computational burden arising from Multi-Era Analysis. See Section 5.4.5.3 for more information.

DFS: Depth-First Search, a "blind" or uninformed network search algorithm. See Search Algorithms for Discrete Networks, for more information.

EEA: Epoch-Era Analysis, an approach to capturing the contextual and temporal aspects of complexity in system design. See (Ross & Rhodes, 2008) for complete description.

Epoch: Period of fixed context and needs.

Era: Ordered sequence of frames.

Era-level Metric: A Metric Of Interest (MOI) and some subset of meta-metrics applied to that MOI over all frames in an era.

Epoch-level Strategies: Also referred to as *tactics*, or epoch-level tactics. Decision rules that only consider the present frame (and potentially past frames) for purposes of optimization of some objective function. The epoch-level strategies created in VASC, for example, reacted to present epochs with no regard for what the future may hold.

---

[1] The author is grateful to M. Fitzgerald for helping identify suitable technical terms such as this one.

Era-level Strategies: Also referred to simply as *strategies*. Decision rules that consider an entire era over which to optimize some objective function. When given complete eras on which to operate, era-level strategies are offline planners.

Frame: The basic unit of era analysis, a frame is an *epoch* matched with a duration of time. An *era* can then be defined as an ordered sequence of frames.

MATE: Multi-Attribute Tradespace Exploration. See (Ross, 2003) for complete description.

MEA: Multi-Epoch Analysis, the analysis of designs, metrics, and epoch-level strategies (i.e., tactics) across the evaluated epoch space (i.e., *tradescape*).

MERA: Multi-Era Analysis, the analysis of designs, metrics, epoch- and era-level strategies, and perturbations across the (practically infinite) era space.

MOI: Metric of Interest, used in the construction of era-level metrics and strategies. Reflects measures of system properties relevant to a study – e.g., MAU, MAE, FPN.

NGCS: Next-Generation Combat Ship, a hypothetical naval system that builds off of the Littoral Combat Ship, Schofield's Offshore Patrol Cutter (OPC) study, and the MIT Math Model used in naval ship design courses at MIT. Used for the application of all 9 processes of RMACS (process 9, MERA, uses an NGCS-like system with discrete change options to form a tradespace network).

Offline Algorithm: An algorithm that requires all of its input before it begins to operate. An example offline sorting algorithm is Selection Sort (http://en.wikipedia.org/wiki/Selection_sort). The era-level strategies discussed in Section 5.3.3 take in an entire era before choosing what to do in any frame of the era. In this way an offline era-level strategy can globally optimize over an era.

Online Algorithm: An algorithm that operates on information as it becomes available, reacting to present and past states of a serialized input. An example online sorting algorithm is Insertion Sort (http://en.wikipedia.org/wiki/Insertion_sort). Epoch-level strategies (i.e., tactics) created in VASC reacted to each epoch as it happened, leading to local but not necessarily global optima.

Path: The sequence and durations of the levels of any metric applied to a trajectory throughout an era.

Path Dependence: Discussed in the present work to consist of two types, Epoch- and Design-level path dependence. Design-level path dependence can be related to the use of change options, of which 2 aspects are identified in the present work. See Section 7.3.1 for more complete description.

RMACS: RSC-based Method for Affordable Concept Selection. The title of this thesis.

RSC: Responsive Systems Comparison, an operationalization of MATE and EEA demonstrated in application to a satellite radar system. See (Ross et al., 2008, 2009) for complete description.

Strategy: When used by itself in this thesis, refers to decision rules that attempt to optimize over an entire era (i.e., an *era-level strategy*). The phrase "*epoch-level strategies*" is occasionally used to refer to *tactics* to foster consistency with prior research (VASC). See Section 5.3.3.2 for details.

Tactics: See epoch-level strategies.

Trajectory: The sequence and durations of all of a design's variables' levels throughout an era. For example, if a design is paired with a strategy, then its trajectory will include all of the designs to which the strategy changes throughout the era, plus the frames encountered.

VASC: Valuation Approach for Strategic Changeability. See (Fitzgerald, 2012) for more detail.

X-TOS: X-Terrestial Observer Swarm, a satellite case study used in applications of MATE and EEA, including applications by VASC.

References

A* Algorithm. (2014). *Wikipedia*. Retrieved May 13, 2014, from http://en.wikipedia.org/wiki/A*_search_algorithm

Abbass, H. a., Bender, A., Dam, H. H., Baker, S., Whitacre, J., & Sarker, R. (2008). Computational scenario-based capability planning. *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation - GECCO '08*, 1437. doi:10.1145/1389095.1389378

Adams, B. M., Ebeida, M. S., Eldred, M. S., Jakeman, J. D., Swiler, L. P., Bohnhoff, W. J., … Hough, P. D. (2013). Dakota , A Multilevel Parallel Object-Oriented Framework for Design Optimization , Parameter Estimation , Uncertainty Quantification , and Sensitivity Analysis Version 5 . 4 Developers Manual, (December 2009).

Affordability Analyses: How do we do it? (2012). Retrieved from http://www.mors.org/UserFiles/file/2013 - Meeting Affordabilty Analysis/Outbriefs/06 _ INCOSE Plenary Panel Brief 2012-08-01.pdf

Arena, M., Blickstein, I., Younossi, O., & Grammich, C. (2006). Why Have Navy Ship Costs Risen? Retrieved January 13, 2013, from http://www.rand.org/pubs/research_briefs/RB9182/index1.html

Bankes, S. (1993). Exploratory Modeling for Policy Analysis. *Operations Research*. doi:10.1287/opre.41.3.435

Berlin, I. (2013). *The hedgehog and the fox: An essay on Tolstoy's view of history*. Princeton University Press.

Biere, A. (2009). *Handbook of satisfiability* (Vol. 185). IOS Press.

Blanchard, B. S., & Fabrycky, W. J. (2006). *Systems Engineering and Analysis*. *Prentice Hall International Series in Industrial and Systems Engineering* (p. 804).

Bobinis, J., Haimowitz, J., Tuttle, P., & Garrison, C. (2012). Affordability Considerations: Cost effective capability. Retrieved February 11, 2012, from http://www.mors.org/UserFiles/file/2012 - Meeting Affordabilty Analysis/Read Aheads/Affordability Considerations_Cost Effective Capability.pdf

Campbell, M., Hoane Jr., A. J., & Hsu, F. (2002). Deep Blue. *Artificial Intelligence*, *134*, 57–83. doi:10.1016/S0004-3702(01)00129-1

Carter, A. B. (2010a). Better Buying Power: Guidance for obtaining greater efficiency and productivity in defense spending [Memorandum]. *Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD [AT&L])*, *September*.

Carter, A. B. (2010b). Better Buying Power: Mandate for restoring affordability and productivity in defense spending [Memorandum]. *Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD [AT&L])*, *June*.

Davis, P., & Kahan, J. (2007). *Theory and Methods for high level decision making*. Santa Monica, CA, USA.

Deep Blue. (2011). Retrieved March 27, 2014, from http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/

Defense Drawdowns Compared. (2012). *Center for Strategic & International Studies*. Retrieved January 12, 2013, from http://intercepts.defensenews.com/wp-content/uploads/2013/02/CSIS-Drawdown111.jpg

Department of Defense. (2011). *Defense Acquisition Guidebook (DAG)*. Retrieved from https://acc.dau.mil/CommunityBrow

Diller, N. (2002). *UTILIZING MULTIPLE ATTRIBUTE TRADESPACE EXPLORATION WITH CONCURRENT DESIGN FOR CREATING AEROSPACE SYSTEMS REQUIREMENTS*. Massachusetts Institute of Technology. Retrieved from http://scripts.mit.edu/~seari/seari/documents/theses/SM_DILLER.pdf

Djikstra's Algorithm. (2014). *Wikipedia*. Retrieved April 19, 2014, from http://en.wikipedia.org/wiki/Dijkstra's_algorithm

Erenrich, J., & Selman, B. (2003). Sampling combinatorial spaces using biased random walks. *IJCAI*, 9–11. Retrieved from http://www.ijcai.org/Past Proceedings/IJCAI-2003/PDF/206.pdf

Fabling, B. (2010). *Offshore Patrol Cutter (OPC) Brief to ASNE*. Retrieved from http://www.uscg.mil/acquisition/opc/pdf/opc_ASNE_16Feb10.pdf

Fischhoff, B., Dawes, R., Slovic, P., Lichtenstein, S., Tversky, A., Kahneman, D., … Sciences, D. (1991). Value Elicitation Is There Anything in There ?, *46*(8), 835–847.

Fitzgerald, M. E. (2012). *Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability*. Massachusetts Institute of Technology. Retrieved from http://seari.mit.edu/documents/theses/SM_FITZGERALD.pdf

Fitzgerald, M. E., & Ross, A. M. (2012). Mitigating contextual uncertainties with valuable changeability analysis in the multi-epoch domain. *2012 IEEE International Systems Conference SysCon 2012*, 1–8. doi:10.1109/SysCon.2012.6189466

Freeman, L. (1977). A set of measures of centrality based on betweenness. *Sociometry*. Retrieved from http://www.jstor.org/stable/3033543

Fulcoly, D. O. B. (2012). *A NORMATIVE APPROACH TO DESIGNING FOR EVOLVABILITY: METHODS AND METRICS FOR CONSIDERING EVOLVABILITY IN SYSTEMS ENGINEERING*. Massachusetts Institute of Technology.

Gaspar, H. M., Ross, A. M., Rhodes, D. H., & Erikstad, S. O. (2012). Handling Complexity Aspects in Conceptual Ship Design. In *Int'l Maritime Design Conference*. Retrieved from http://scripts.mit.edu/~seari/seari/documents/preprints/GASPAR_IMDC12.pdf

Goode, H. H., Machol, R. E., & Teichmann, T. (1957). System engineering. *Physics Today*, *10*, 34.

Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. *Journal of the American Statistical Association*. doi:10.1198/016214508000000689

Hall, A. D. (1962). A methodology for systems engineering.

Hall, A. D. (1989). *Metasystems methodology: a new synthesis and unification*. Pergamon Press New York.

Hastings, D. E. (2010). Analytic Framework for Adaptability. *Presentation to Defense Science Board*.

Herald, T. (2011). *Affordable Architectures - "How do I recognize one?"*

Howell, D., Uebelhart, S. A., Angeli, G., & Miller, D. W. (2005). Large tradespace analysis using parametric integrated models of extremely large ground-based telescopes. In *6th World Congress on Structural and Multidisciplinary Optimization*.

INCOSE Affordability Working Group Jan 2013 Summary. (2013).

Johnson-Laird, P. N. (2005). Mental models and thought. *The Cambridge Handbook of Thinking and Reasoning*, 185–208.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, 263–291.

Kahneman, D., & Tversky, A. (1984). Choices, values, and frames. *American Psychologist*, *39*(4), 341.

Kapurch, S. J. (2010). *NASA Systems Engineering Handbook*. DIANE Publishing.

Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., & Teller, S. (2011). Anytime Motion Planning using the RRT*. *2011 IEEE International Conference on Robotics and Automation*, 1478–1483. doi:10.1109/ICRA.2011.5980479

Kavraki, L., & Svestka, P. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *... and Automation, IEEE ...*. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=508439

Keeney, R. L., & Raiffa, H. (1976). Decision with multiple objectives. Wiley, New York.

Kichkaylo, T., Hoag, L., Lennon, E., & Roesler, G. (2012). Highly efficient exploration of large design spaces: fractionated satellites as an example of adaptable systems. *Procedia Computer Science*, *8*, 428–436.

LaValle, S. M., & Kuffner Jr, J. J. (2000). Rapidly-exploring random trees: Progress and prospects.

Law, A. M., & Kelton, W. D. (1991). *Simulation modeling and analysis* (Vol. 2). McGraw-Hill New York.

Lin, D., & Iii, J. (2012). Efficient Sampling from Combinatorial Space via Bridging. In *International Conference on Artificial Intelligence and Statistics* (pp. 694–702).

Mallory, C. J. (2011). *International programs contribute to affordability*.

McManus, H. L., & Schuman, T. E. (2003). Understanding the Orbital Transfer Vehicle Trade Space. *AIAA Paper*, *6370*.

Mekdeci, B. (2013). *Managing the impact of change through survivability and pliability to achieve viable systems of systems*. Citeseer.

Mekdeci, B., Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2011). Examining Survivability of Systems of Systems. *Symposium A Quarterly Journal In Modern Foreign Literatures*, 1–12. Retrieved from http://seari.mit.edu/documents/preprints/MEKDECI_INCOSE11.pdf

Mendonça, S., e Cunha, M., Kaivo-Oja, J., & Ruff, F. (2004). Wild cards, weak signals and organisational improvisation. *Futures*, *36*(2), 201–218.

MoGo: A Grid5000-based software for the Game of Go. (2012). Retrieved March 27, 2014, from https://www.lri.fr/~teytaud/mogo.html

NDIA Status Report. (2011). Retrieved from http://www.psmsc.com/Downloads/2012WinterTelecoms/Affordability Working Group - SE Division Strategy Meeting 111207.pdf

Neches, R., & Madni, A. M. (2013). Towards affordably adaptable and effective systems. *Systems Engineering*, *16*(2), 224–234.

Nickel, J. (2010). Trading political feasibility and technical optimality: Multi-attribute tradespace exploration as communication tool in transportation planning. In *Industry Studies Conference*.

Patel, A. (n.d.). Heuristics. Retrieved May 13, 2014, from http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html

Patterson, J. D. (2012). *Basis for a Rational Defense: Acquiring the Right Capability*.

Rhodes, D. H., & Ross, A. M. (2010). Five aspects of engineering complex systems emerging constructs and methods. *Systems Conference, 2010 4th Annual IEEE*. doi:10.1109/SYSTEMS.2010.5482431

Ricci, N. (2014). *Dynamic System Perspective for Design: Ility-Driving Entities as Responses to Uncertainty*. Massachusetts Institute of Technology.

Ricci, N., Schaffner, M. A., Ross, A. M., Rhodes, D. H., & Fitzgerald, M. E. (2014). Exploring Stakeholder Value Models via Interactive Visualization. *Procedia Computer Science*, *28*, 294–303.

Richards, M. G. (2009). *Multi-attribute tradespace exploration for survivability*. Massachusetts Institute of Technology, Engineering Systems Division.

Richards, M. G., Hastings, D. E., Rhodes, D. H., & Weigel, A. L. (2007). Defining Survivability for Engineering Systems. *Conference on Systems Engineering Research*, 1–12.

Roberts, C. J., Richards, M. G., Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2009). Scenario planning in dynamic multi-attribute tradespace exploration. *2009 3rd Annual IEEE Systems Conference*. doi:10.1109/SYSTEMS.2009.4815828

Ross, A. M. (2003). *Multi-Attribute Tradespace Exploration with Concurrent Design as a Value-Centric Framework for Space System Architecture and Design*. *Science*. Massachusetts Institute of Technology. Retrieved from http://seari.mit.edu/documents/theses/SM_ROSS.pdf

Ross, A. M., & Hastings, D. E. (2004). Multi-attribute tradespace exploration as front end for effective space system design. *Journal of Spacecraft ...*, *41*(1). Retrieved from http://arc.aiaa.org/doi/pdf/10.2514/1.9204

Ross, A. M., & Hastings, D. E. (2006). Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration. *American Institute of Aeronautics and Astronautics*, *7255*.

Ross, A. M., McManus, H. L., & Long, A. (2008). Responsive systems comparison method: Case study in assessing future designs in the presence of change. *AIAA Space ...*, 1–9. Retrieved from http://seari.mit.edu/documents/preprints/ROSS_AIAA08.pdf

Ross, A. M., McManus, H. L., Rhodes, D. H., Hastings, D. E., & Long, A. (2009). Responsive Systems Comparison Method: Dynamic Insights into Designing a Satellite Radar System. *AIAA SPACE 2009 Conference & Exposition*, (c), 14–17. doi:10.2514/6.2009-6542

Ross, A. M., & Rhodes, D. H. (2008). Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis.

Russell, S., & Norvig, P. (2009). Artificial Intelligence: A Modern Approach, 3rd edition. *Prentice Hall*. doi:10.1017/S0269888900007724

Sabharwal, A. (2008). *Combinatorial Problems II: Counting and Sampling Solutions*. Retrieved from http://www.cs.cornell.edu/~sabhar/tutorials/kitpc08-combinatorial-problems-II.ppt

Sarewitz, D., & Pielke Jr, R. (1999). Prediction in science and policy. *Technology in Society*, *21*(2), 121–133.

Schofield, D. M. (2010). *A framework and methodology for enhancing operational requirements development: United States Coast Guard cutter project case study*. Massachusetts Institute of Technology.

Schön, D. A. (1993). Generative metaphor: A perspective on problem-setting in social policy.

Schrijver, A. (2005). On the History of Combinatorial Optimization (Till 1960). *Handbooks in Operations Research and Management Science*. doi:10.1016/S0927-0507(05)12001-5

Schwenn, R. E., Chitikila, R. C., Laufer, D. R., Rozzi, A. D., Smythe, W. P., Best, D. B., … Graveline, W. R. (2011). *Defense Acquisitions: Assessments of Selected Weapon Programs*.

Simon, H. A. (1996). *The sciences of the artificial*. MIT press.

Smith, L. M. B., Smith, M., & Marcus, H. (2008). Updating MIT ' s Cost Estimation Model for Shipbuilding, (2004).

Storlie, C. B., Bondell, H. D., Reich, B. J., & Zhang, H. H. (2011). Surface estimation, variable selection, and the nonparametric oracle property. *Statistica Sinica*, *21*(2), 679.

Swiler, L. P., Hough, P. D., Qian, P., Xu, X., Storlie, C., & Lee, H. (2014). Surrogate models for mixed discrete-continuous variables. In *Constraint Programming and Decision Making* (pp. 181–202). Springer.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, *38*(3), 58–68.

Tolkien, J. R. R. (1974). The Fellowship of the Ring. 1954. London: Unwin Books.

Tolstoy, L. (2010). *War and peace*. Digireads. com Publishing.

Tuttle, P., & Bobinis, J. (2012). Affordability Specification, 1–13.

Van Harmelen, F., Lifschitz, V., & Porter, B. (2008). *Handbook of Knowledge Representation*. *Elsevier Science San Diego USA* (Vol. 3, pp. 3–88). doi:10.1016/S1574-6526(07)03013-1

Viscito, L., Chattopadhyay, D., & Ross, A. M. (2009). Combining Pareto Trace with Filtered Outdegree as a Metric for Identifying Valuably Flexible Systems. *CSER 2009*, *2009*, 1–9.

Viscito, L., & Ross, A. M. (2009). Quantifying flexibility in tradespace exploration: Value weighted filtered outdegree. *AIAA Space 2009*, 1–9.

Viscusi, W. K., Harrington, J. E., & Vernon, J. M. (2005). *Economics of regulation and antitrust*. MIT press.

Von Neumann, J., & Morgenstern, O. (n.d.). Theory of Games and Economic Behavior (1944). *Nature*, *246*, 15–18.

W, B. (n.d.). Methods of Search. *CSE.UNSW.edu.au*. Retrieved March 27, 2014, from http://www.cse.unsw.edu.au/~billw/Justsearch.html

Welch, P. (2012). *And Then I Thought I Was a Fish*. 12by3 Press.

Wu, M. S. (2014). *Design for Affordability in Defense and Aerospace Systems Using Tradespace-based Methods*. Massachusetts Institute of Technology.

Wu, M. S., Ross, A. M., & Rhodes, D. H. (2014). Design for Affordability in Complex Systems and Programs Using Tradespace-based Affordability Analysis. *Procedia Computer Science*, *28*, 828–837.

Appendix

"The appendix? You can live without it."

-Rachel Schaffner, Massachusetts Surgical R.N.

Pseudocode for A* Algorithm (from Wikipedia's A* algorithm):

```
function A*(start,goal)
   closedset := the empty set    // The set of nodes already evaluated.
   openset := {start}    // The set of tentative nodes to be evaluated, initially containing the start node
   came_from := the empty map    // The map of navigated nodes.

   g_score[start] := 0    // Cost from start along best known path.
   // Estimated total cost from start to goal through y.
   f_score[start] := g_score[start] + heuristic_cost_estimate(start, goal)

   while openset is not empty
      current := the node in openset having the lowest f_score[] value
      if current = goal
         return reconstruct_path(came_from, goal)

      remove current from openset
      add current to closedset
      for each neighbor in neighbor_nodes(current)
         if neighbor in closedset
            continue
         tentative_g_score := g_score[current] + dist_between(current,neighbor)

         if neighbor not in openset or tentative_g_score < g_score[neighbor]
            came_from[neighbor] := current
            g_score[neighbor] := tentative_g_score
            f_score[neighbor] := g_score[neighbor] + heuristic_cost_estimate(neighbor, goal)
            if neighbor not in openset
               add neighbor to openset

   return failure

function reconstruct_path(came_from, current_node)
   if current_node in came_from
      p := reconstruct_path(came_from, came_from[current_node])
      return (p + current_node)
   else
      return current_node
```

Matlab code for creating transition matrix for the change rule Add Engine:

```
%CREATE Transition Matrix for change option: Number of Engines
%DESCRIPTION: calculate all reachable designs in one or more arcs from adding an engine
%INPUT: a design ID, and the number of epochs (SHOULD BE 1, since the
%      recursive portion has been removed.)
%NGCSbaselineDesigns.mat is a file containing a struct of all designs evaluated in the
%baseline epoch.
%
function [reachableDesigns,moneyVector,timeVector] = calcNGCStransMatrix_EngineRule(designs,numEpochs)
load NGCSbaselineDesigns.mat
%init
LengthIdx = 6;
AirIdx = 14;
VolErrIdx = 23;
GenErrIdx = 25;
CrewIdx = 16;
EngineIdx = 9;

moneyVector=zeros(134,1);
timeVector=zeros(134,1);
%UNUSED. Originally for recursive version
%stopping condition since this is a recursive function
%if numEpochs==0
%    reachableDesigns = 0;
%    return;
%end
allCloseDesigns = 0;
for i=1:length(designs)
   currentEngines = NPSexpandedcopy(designs(i),EngineIdx);
   %find all designs that have one more engine
   closeDesigns = find(NPSexpandedcopy(:,EngineIdx) - currentEngines == 1);
   %and similar length
   closeDesigns = closeDesigns(find(abs(NPSexpandedcopy(closeDesigns,LengthIdx) - NPSexpandedcopy(designs(i),LengthIdx)) < 20));

   %and now find those close designs with less volume error
   closeDesigns = closeDesigns(NPSexpandedcopy(closeDesigns,VolErrIdx) < NPSexpandedcopy(designs(i),VolErrIdx));

   allCloseDesigns = vertcat(allCloseDesigns,closeDesigns);
end
%keep only unique,nonzero elements
allCloseDesigns = unique(allCloseDesigns(allCloseDesigns>0));
%UNUSED. Originally for recursive version
%now find the designs that they can transition to in future epochs
%allCloseDesigns = vertcat(allCloseDesigns,calcNGCStransMatrix(allCloseDesigns,numEpochs-1));

%get list of all designs
reachableDesigns = unique(allCloseDesigns(find(allCloseDesigns)));
%and calculate the cost to transition to each. (note: only works for 1 design and 1 epoch transition at present)
if (numEpochs==1)
   [moneyVector,timeVector] = calculateCostToReach(designs,reachableDesigns,'air');
end
end
```

Matlab code for epoch-level Survive strategy:

```
%% Implementation of the epoch-level Survive strategy.
%% Returns a target design, given a starting design in a frame and given
%% change mechanism matrices. Assumes single-arc transitions.
%   OUTPUTS:
%       1) single design number (ending design of the frame)
%       2) money cost of transition (if any)
%       3) time cost of transition (if any)
%       4) rule number executed (if any)
%   INPUTS:
%      1) single design number  (starting design in the frame)
%      2) frame = (epochNumber, duration)
%      3) allData = struct of all design data in all epochs (minimally
%      required: MAU and MAE in all epochs)
%      4) changeMoneyCArray = a 1xM cell array, where M is the number of
%         change rules, and each cell is a NxN transition matrix of
%         money costs, N=number of designs
%      5) changeTimeCArray = a 1xM cell array, where M is the number of
%         change rules, and each cell is a NxN transition matrix of
%         time costs, N=number of designs

function [targetDesign,moneyCost,timeCost,ruleExecuted] =
epochStrategy_Survive(design,frame,allData,changeMoneyCArray,changeTimeCArray,epochImpactCO)

   epochNum = frame(1);
   epochDuration = frame(2);
   ruleExecuted = 0;
   %might be working with structs, instead of tables as originally written.
   if ~isfield(allData,'MAU')
      allDataMAU = transpose([allData.MAU]);
      allDataMAE = transpose([allData.MAE]);
   else
      allDataMAU = [allData.MAU];
      allDataMAE = [allData.MAE];
   end

   %only load the rows of interest (this epoch's rows, that is)
   epochNum_Rows = find(transpose([allData.Epoch])==epochNum);
   %non-table
   %epochNum_Rows = find(allData(:,epochCol)==epochNum)

   designIdx=epochNum_Rows(design);
   targetDesign=design;

   %check to see if the input design is feasible in the current epoch
   if ((allDataMAU(designIdx) > -1) && (allDataMAE(designIdx) > -1))
      moneyCost=0;
      timeCost=0;
   else   %else change to the cheapest (in money) feasible design
      %init cheapest option
      moneyCost = inf;
      %time units are assumed to be same units as the frame duration
      timeCost = epochDuration;
      for i=1:length(changeMoneyCArray)  % for each change rule with monetary cost
         %store the current rule's matrix
         moneyMatrix = changeMoneyCArray{i};
```

```
        timeMatrix = changeTimeCArray{i};

        %don't consider change options blocked by the current epoch
        if epochImpactCO(epochNum,i) == 0
           continue;
        end

        %report the indices of the money matrix that are not empty
        allDesign = find(moneyMatrix(design,:));

        %now loop through the available designs we could switch to
        for j=1:length(allDesign)
           targetDesignCandidate=allDesign(j);
           %is the target design feasible?
           if (allDataMAU(epochNum_Rows(targetDesignCandidate)) > -1)
              if (allDataMAE(epochNum_Rows(targetDesignCandidate)) > -1)
                 candidateTimeCost=timeMatrix(design,targetDesignCandidate);
                 %Is changing going to take longer than the best so far?
                 if candidateTimeCost > timeCost
                    continue;
                 %Is changing going to take the same time as the
                 %best so far?
                 elseif candidateTimeCost == timeCost
                    %is changing to this target design cheaper than
                    %the previous best option we found with this time cost?
                    if (moneyMatrix(design,targetDesignCandidate)*epochImpactCO(epochNum,i) < moneyCost)
                       %update the new "best" candidate
                       moneyCost = moneyMatrix(design,targetDesignCandidate)*epochImpactCO(epochNum,i);
                       timeCost = timeMatrix(design,targetDesignCandidate);
                       targetDesign = targetDesignCandidate;
                       ruleExecuted = i;
                    end
                 %Else: candidateTimeCost < timeCost
                 else
                    %update the new "best" candidate
                    moneyCost = moneyMatrix(design,targetDesignCandidate)*epochImpactCO(epochNum,i);
                    timeCost = timeMatrix(design,targetDesignCandidate);
                    targetDesign = targetDesignCandidate;
                    ruleExecuted = i;
                 end
              end
           end
        end
     end
     %no changes available? Then no change cost.
     if moneyCost == inf
        moneyCost=0;
        timeCost=0;
     end
  end
end
```

Matlab code for the era-level Survive strategy. It is a form of the A* algorithm with some custom heuristics and cost calculations.

```
%%A* implementation for navigating a design-tradespace meta-network with
%   the era-level Survive strategy.
%
% OUTPUT: an Nx5 matrix, with N being the number of frames in the era, and
%  each row 'n' being a 5-tuple in the format:
%  <designI,changeRule,designJ,epochNumber,duration>
%  If no change was executed, designI = design J, and changeRule = 0.
%
% INPUTS:  1) a starting design number
%          2) an n-tuple of frames, i.e., <epochNum,duration>
%          3) allData, a struct of all design data evaluated in all epochs
%          4) changeMoney, a cell array of change cost transition matrices
%          5) changeTime, a cell array of change cost transition matrices
%          6) epochImpactCO, an ExC matrix, where E = number of
%          epochs total and C = number of change rules total, and the
%          value of each cell is the cost multiplier for the cost of that
%          change rule in that epoch ('0' cost for blocked change).
function [sequence,sizeOfQueue] =
calc_trajectory_eraSurvive(startingDesign,era,allData,changeMoney,changeTime,epochImpactCO)

   %prepare the trajectory with the number of frames, with 5 elements per
   %frame (designI,change,designJ,epochNumber,duration). The change=0 if
   %there is no change, and resultingly designJ = designI.
   sequence = zeros(length(era),5);
   eraLength = length(era);
   openset=zeros(1,6);
   closedset=zeros(1,6);
   %closedset=vertcat(closedset,[32,2,0,0,0,0])
   nodeChildren=[];
   %might be working with structs, instead of tables as originally written.
   if ~isfield(allData,'MAU')
      allDataMAU = transpose([allData.MAU]);
      allDataMAE = transpose([allData.MAE]);
   else
      allDataMAU = [allData.MAU];
      allDataMAE = [allData.MAE];
   end
   %Add the startingDesign to the open list
   NodeDesign = startingDesign;
   NodeFrame = 1;
   NodeParent = [0,0];
   gCost = 0;
   fCost = testFeasibility([NodeDesign,NodeFrame]);

   %this is the structure of every node, but just set the first one for
   %now
   openset(1,:) = [NodeDesign,NodeFrame,gCost,fCost,NodeParent];
   while (any(openset(1,:)))
      %get the lowest H cost on the openList
      [~,I] = min(openset,[],1);
      currentNode = openset(I(4),:);

      %if we're in the goal region, report the path found
      if (currentNode(2)==eraLength+1)
```

```
    sequence = createTrajectory(reconstruct_path(currentNode));
    sizeOfClosedSet = size(closedset,1);
    sizeOfQueue = size(openset,1)+sizeOfClosedSet;
    return
end

%add current to closedset
closedset = vertcat(closedset,currentNode);

%remove current from openset
openset(I(4),:) = [];

%SINGLE-ARC TRANSITIONS
%if the currentNode is in the same frame as its parent, then don't
%consider any other children (i.e., single-arc transitions, i.e.
%only one change per epoch).  (This assumption is elsewhere -- look
%for comments that mention SINGLE-ARC.)
nodeChildren = [];
if (currentNode(2)~=currentNode(6))
    %get currentNode's children, i.e. designs it can change to
    for j= 1:length(changeMoney)
        %don't consider change options blocked by the current epoch
        if epochImpactCO(era(currentNode(2),1),j)==0
            continue;
        end
        nodeChildren = [nodeChildren find(changeMoney{j}(currentNode(1),:))];
    end
    nodeChildren = unique(nodeChildren);
end
%make sure [this design in the next frame] is available as a child
nextFrameNode = currentNode;
nextFrameNode(2) = currentNode(2)+1;
nextFrameNode(5:6) = currentNode(1:2);

%now create the list of all children
allNodeChildren = zeros(length(nodeChildren)+1,6);
for j=1:length(nodeChildren)
    allNodeChildren(j+1,:) = [nodeChildren(j),currentNode(2),0,0,currentNode(1:2)];
end
%and add [this design in the next frame] to the list of all children
allNodeChildren(1,:) = nextFrameNode;

%for each child
for c = 1:size(allNodeChildren,1)
    currentChild = allNodeChildren(c,:);
    %if it is on the closed list already, move on (should this
    %check the parent (columns 5:6 of the closedSet node) as well?
    %I don't believe so, so it doesn't currently.)
    [~,indexInClosedSet]=ismember(currentChild(1:2),closedset(:,1:2),'rows');
    if indexInClosedSet > 0
        continue;
    end

    %get child's feasibility
    childInfeasible = testFeasibility(currentChild);
    %check if its in same frame (all except 1 child should be)
    if currentChild(2) == currentNode(2)
```

```
    %count up the cost of taking it -- time and money of change
    [timeToChange,ruleUsed] = dist_between(currentNode,currentChild);
    %debug
    if ruleUsed<1
        currentNode
        currentChild
        allNodeChildren
        ruleUsed
        asdf;
    end
    %any time spent changing is time spent in infeasible state
    moneyChange = money_between(currentNode,currentChild,ruleUsed);
    %debug
    if isnan(moneyChange)
        asdf
    end
    tentative_g_score = currentNode(3) + timeToChange*1000 + moneyChange;

    %if it's infeasible itself, then an optimistic heuristic
    %cost will be that infeasibility after changing (since SINGLE-ARC assumed)
    %otherwise, no heuristic cost since that's most optimistic
    if childInfeasible > 0
        heuristicCost = childInfeasible-timeToChange*1000;
    else
        %otherwise an optimistic heuristic is just the time
        %required to change
        heuristicCost = timeToChange*1000;
    end
else
    %if it is in the next frame, (i.e. same design in next
    %frame) then count the current design's infeasibility
    %against it (if applicable)
    tentative_g_score = currentNode(3) + testFeasibility(currentNode);

    %if this child is the same design, but in the next frame, and
    %it's infeasible there, then we need to check what the lowest
    %cost to get out of it is. (only check one change, even if
    %not assuming SINGLE-ARC Transitions, since we want optimism)
    if childInfeasible > 1
        %if it's not feasible in the next frame, cost is set
        %accordingly
        hCost=childInfeasible;
        moneyChange=0;
        ruleUsed = -1;
        %this might be a "child" of the current design in the
        %last frame of the era, whose frame index actually
        %exceeds the eraLength. But otherwise, let it change out of infeasibility if need be.
        if (currentChild(2) <= eraLength)
            currentEpoch = era(currentChild(2),1);
            %and now get the cheapest change possible, if any (only
            %needs to be optimistic, not accurate, for an
            %admissible heuristic)
            for b= 1:length(changeTime)
                %don't consider change options blocked by the epoch
                if epochImpactCO(currentEpoch,b)==0
                    continue;
                end
```

```
                thisDesignsChangeRow = changeTime{b}(currentChild(1),:);
                indicesNotZero = find(thisDesignsChangeRow);
                [lowdist idx] = min(thisDesignsChangeRow(indicesNotZero));
                idx = indicesNotZero(idx);
                if ~isempty(lowdist) && lowdist*1000 < hCost
                   hCost = lowdist*1000;
                   ruleUsed = b;
                   futureChild=[idx,currentChild(2)];
                end
             end
             % if a change was found, then get the time required to change
             if ruleUsed > 0
                moneyChange = money_between(currentChild,futureChild,ruleUsed);
                %debug
                if isnan(moneyChange)
                   asdf
                end
             end
          end

          %now set the heuristic cost
          heuristicCost = (hCost + moneyChange);%*(0.98^currentChild(2))

       else
          %otherwise, this design in the next frame is feasible,
          %so estimated no cost
          heuristicCost = 0;
       end
    end

    %update the g_score of the child node
    currentChild(3) = tentative_g_score;
    %and update the heuristicCost
    currentChild(4) = currentChild(3) + heuristicCost;
    %make currentNode the parentNode of this node
    currentChild(5:6) = currentNode(1:2);

    [~,indexInOpenSet]=ismember(currentChild(1:2),openset(:,1:2),'rows');
    %if it's not on the open list yet
    if indexInOpenSet < 1
       %add to the open list
       openset = vertcat(openset,currentChild);
    else  %if it's already on the open list
       %if G cost is better
       %use G cost as the measure of this path over the existing
       if tentative_g_score < openset(indexInOpenSet,3)
          %replace the existing node on the openset with this
          %node
          openset(indexInOpenSet,:) = currentChild(:);
       end
    end
 end
 %prevent program from crashing (shouldn't really ever happen)
 if length(openset)<1
    break;
 end
end
```

```
%no path found. :(
%(should never happen). :)
sequence=-1;

%return 1000*timeUnit for every timeUnit this node is infeasible
function [costOfInfeasibility] = testFeasibility(node)
   %check if this is our "extra" node at the end (i.e., the goal
   %region)
   if node(2)>eraLength
      %if so, grab its parent's frame
      frameNum = node(6);
   else
      frameNum = node(2);
   end
   %get the index into allNGCSdata table
   designIdx = node(1)+134*(era(frameNum,1)-1);

   %and return either 0 (if feasible) or the penalty multiplied by
   %the time (if infeasible)
   if (allDataMAU(designIdx)>-1 && allDataMAE(designIdx)>-1)
      costOfInfeasibility = 0;
   else
      %
      costOfInfeasibility = 1000*era(frameNum,2);
   end
end

%designed to be used after the dist_between function gives us
%the cheapest time-cost rule used
function [moneyToChange] = money_between(nodeI,nodeJ,rule)
   moneyToChange = NaN;
   if nodeI(1) == nodeJ(1)
      moneyToChange=0;
      return
   end
   %modify the cost by the current epoch's impact on changes (should
   %not be 0, ever, since we call dist_between first)
   moneyDist = changeMoney{rule}(nodeI(1),nodeJ(1))*epochImpactCO(era(nodeI(2),1),rule);
   %this function should only be called when there is a known
   %transition rule connecting nodeI and nodeJ. But just in case...
   if moneyDist > 0
      moneyToChange = moneyDist;
   end
end

%lookup the time cost in change between two nodes,
%return the cheapest, along with which rule we used
function [timeCost,rule] = dist_between(nodeI,nodeJ)
   timeCost=inf;
   rule=0;
   if nodeI(1) == nodeJ(1)
      timeCost=0;
      rule=0;
      return
   end
   for a= 1:length(changeTime)
```

```
        %don't consider blocked options
        if epochImpactCO(era(nodeI(2),1),a)==0
            continue
        end
        dist = changeTime{a}(nodeI(1),nodeJ(1));
        if dist>0 && dist < timeCost
            timeCost = dist;
            rule = a;
        end
    end
  end

  %using each node's parent, reconstruct the path from the goal node
  function [out] = reconstruct_path(current_node)
    if current_node(5)==0
      out = current_node;
    else
        [~,idxInClosedSet]=ismember(current_node(5:6),closedset(:,1:2),'rows');
        out = vertcat(reconstruct_path(closedset(idxInClosedSet,:)),current_node);
    end
  end

  %convert the path found in reconstruct_path() to the 5-tuple trajectory form:
  %design-change-design-epoch-duration
  %design-change-design-epoch-duration
  function [seq] = createTrajectory(finalPath)
    seq = zeros(eraLength,5);
    for i=1:eraLength
        seq(i,end-1) = era(i,1);
        seq(i,end) = era(i,2);
        indices = find(finalPath(:,2) == i);
        for k=1:length(indices)
            seq(i,k*2-1)=finalPath(indices(k),1);
            if (length(indices)<2)
                seq(i,end-2)=finalPath(indices(k),1);
            elseif k==2
                [~,transitionRule] = dist_between(finalPath(indices(k-1),:),finalPath(indices(k),:));
                seq(i,2)=transitionRule;
            end
        end
    end
    %if there was a change in the last frame, record the change and
    %final design
    if seq(end,3) ~= finalPath(end,1)
        [~, seq(end,2)] = dist_between([seq(end,3),0],[finalPath(end,1),0]);
        seq(end,3) = finalPath(end,1);
    end
  end
end
```

Matlab code for solving the all-pairs shortest path problem using Djikstra's, creating a Full Accessibility Matrix for all combinations of change options.

```
%%Djikstra's algorithm for generating a change matrix that records the
%%shortest path to each designJ from any designI, over as many arcs as are
%%available to designI. Adapted from Wikipedia, Spring 2014.
%
%OUTPUT:
%       1) one MxM matrix recording the least cost to get to each designJ
%          from any designI, where M is the number of designs.
%       2) one MxMxN matrix recording the number of times each change
%          mechanism (little n) was used to get from Mi to Mj.
%
%INPUT:  1) a 1xN cell array of change matrices, each matrix of size MxM,
%          where M is the number of designs, and the content of each cell
%          is the cost associated with transitioning from Mi to Mj.
%
function [leastCostMatrix,leastCostRulesUsed] = preprocessAllArcs(changeCostCellArray)
   tic
   numDesigns = size(changeCostCellArray{1},1);
   numChanges = length(changeCostCellArray);
   leastCostMatrix = zeros(numDesigns);
   leastCostRulesUsed = zeros(numDesigns,numDesigns,numChanges);
   %run Djikstra's for every design
   for i=1:numDesigns
      if mod(i,100)==0
         toc
         i/numDesigns
      end
      %initialize our lists
      allDesignNodes = zeros(numDesigns,2);
      allChangeRuleSlots = zeros(numDesigns,numChanges);
      allNodes = [allDesignNodes allChangeRuleSlots];
      openList = zeros(numDesigns,2+numChanges);
      currentDesignNum = i;
      %1. Assign to every node a tentative distance value: set it to zero for
      %our initial node and to infinity for all other nodes
      %create all nodes in the network from the number of designs present
         for j=1:numDesigns
            %design number will be the node ID, the second value will be
            %the "distance" to this node from our currentDesignNum
            %node, and the last value is the parent nodeID
            allNodes(j,1:2) = [j,inf];
         end
         allNodes(currentDesignNum,1:2) = [currentDesignNum,0];
      %2. Set initial node as current. Create a set
      %of unvisited nodes called the unvisited set consisting of all the
      %nodes
         currentNode = allNodes(currentDesignNum,:);
         for j=1:numDesigns
            openList(j,:) = allNodes(j,:);
         end
     %3. For current node, consider all unvisited neighbors and calculate
      %their "tentative" distances to the current node. Compare with their
      %currently assigned value and assign the smaller one.
      [~,minDistIdx] = min(openList,[],1);
      minTentativeDist = openList(minDistIdx(2),2);
```

```
    while ~isinf(minTentativeDist)
      neighbors = [];
      for c=1:numChanges
        neighbors = [neighbors find(changeCostCellArray{c}(currentNode(1),:)>0)];
      end
      neighbors = unique(neighbors);
      %loop through neighbors
      for k=1:length(neighbors)
        j = neighbors(k);
        %get distance to each neighbor, used to be dist_between
        %function
          edgeCostToNeighbor=inf;
          ruleUsed=0;
          for a= 1:numChanges
            dist = changeCostCellArray{a}(currentNode(1),j);
            if dist>0 && dist < edgeCostToNeighbor
              edgeCostToNeighbor = dist;
              ruleUsed = a;
            end
          end
        %[edgeCostToNeighbor,ruleUsed] = dist_between(currentNode(1),j);
        costToCurrentNode = currentNode(2);
        if edgeCostToNeighbor+costToCurrentNode < allNodes(j,2)
          %update cost to this neighbor
          allNodes(j,2) = edgeCostToNeighbor+costToCurrentNode;
          %update which/number of rules used to get to this neighbor
          allNodes(j,3:end)=currentNode(3:end);
          allNodes(j,2+ruleUsed)=currentNode(2+ruleUsed) + 1;
          %update the rule tracking matrix
          leastCostRulesUsed(currentDesignNum,j,1:end) = ...
            allNodes(j,3:end);
          %and update the open set, if this neighbor's on it
          if (openList(j,1) ~= inf)
            openList(j,2) = edgeCostToNeighbor+costToCurrentNode;
            openList(j,3:end)=allNodes(j,3:end);
          end
        end
      end
    %4. If done with all neighbors of the current node, mark it as visisted
    %and remove it from the "unvisited set". Never check it again.
      openList(currentNode(1),:) = inf;
    %5. If the smallest tentative distance in the unvisisted set is infinity,
    %then stop. The algorithm has finisehd.
      %(Taken care of by the "while" loop above
    %6. Select the unvisisted node that is marked with the smallest
    %tentative distance; set as the new "current node". Go back to step 3.
      [~,minDistIdx] = min(openList,[],1);
      minTentativeDist = openList(minDistIdx(2),2);
      currentNode = openList(minDistIdx(2),:);
    end %return to "while" in step 3
    %if done with this design's shortest path, record it in the
    %appropriate place
    allNodes(allNodes==inf) = 0;
    leastCostMatrix(i,:) = allNodes(:,2);
  end
  toc
end
```

Matlab code for generating the input to the Force-Directed Visualization web code.

```
%INPUTS:  changeMatrix = a single change matrix with costs of transitioning
%between designs  (and 0 cost means no transition).
%        forceDirectedFilename = directory+filename for output TXT, for
%        use with the forceDirected HTML file.
%
%OUTPUTS: 1) a .txt file with a list of nodes and directed links between nodes.
%    IMPORTANT NOTE: may need to delete an odd character in the output file,
%as well as two extraneous commas (after the collections of nodes and edges).

function out = makeTradespaceNetworkForceDirected(changeMatrix,forceDirectedFilename)
   N = length(changeMatrix);

   allNodes = {N,1};
   allEdges = {};
   k=0;
   for i=1:size(changeMatrix,1)

      allNodes{i} = strcat('{"name":"Design',num2str(i),'","group":1},');
      for j=1:size(changeMatrix,2)
         if changeMatrix(i,j) > 0
            k=k+1;
            allEdges{k} = strcat('{"source":',num2str(i-1),',"target":',num2str(j-1),',"value":',num2str(changeMatrix(i,j)),'},');
          end
      end

   end
   allLines = strcat('{"nodes":[',allNodes{:},'],"links":[',allEdges{:},']}');
   dlmwrite(forceDirectedFilename,allLines,'');
end
```

Web code for generating a D3 interactive force-directed graph of a network.

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>
.node {
  stroke: #fff;
  stroke-width: 1.5px;
}
.link {
  stroke: #999;
  stroke-opacity: .6;
}
</style>
<body>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script>
var width = 960,
    height = 500;
var color = d3.scale.category20();
var force = d3.layout.force()
    .charge(-20)
    .linkDistance(30)
    .size([width, height]);
var svg = d3.select("body").append("svg")
    .attr("width", width)
    .attr("height", height);
d3.json("forceDirected.txt", function(error, graph) {
  force
      .nodes(graph.nodes)
      .links(graph.links)
      .start();
  var link = svg.selectAll(".link")
      .data(graph.links)
    .enter().append("line")
      .attr("class", "link")
      .style("stroke-width", function(d) { return Math.sqrt(Math.sqrt(d.value)); });
  var node = svg.selectAll(".node")
      .data(graph.nodes)
    .enter().append("circle")
      .attr("class", "node")
      .attr("r", 5)
      .style("fill", function(d) { return color(d.group); })
      .call(force.drag);
  node.append("title")
      .text(function(d) { return d.name; });
  force.on("tick", function() {
    link.attr("x1", function(d) { return d.source.x; })
        .attr("y1", function(d) { return d.source.y; })
        .attr("x2", function(d) { return d.target.x; })
        .attr("y2", function(d) { return d.target.y; });
    node.attr("cx", function(d) { return d.x; })
        .attr("cy", function(d) { return d.y; });
  });
});
</script>
</body>
```