MANAGING UNCERTAINTY IN SYSTEMS WITH A VALUATION APPROACH FOR STRATEGIC CHANGEABILITY

by

Matthew Edward Fitzgerald

S.B. Aeronautics and Astronautics
Massachusetts Institute of Technology, 2010

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 2012

Signature of Author...
Department of Aeronautics and Astronautics
May 23, 2012

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

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

Accepted by...
Daniel E. Hastings
Professor of Aeronautics and Astronautics and Engineering Systems
Thesis Reader

Accepted by...
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

MANAGING UNCERTAINTY IN SYSTEMS WITH A VALUATION APPROACH FOR STRATEGIC CHANGEABILITY

by

Matthew Edward Fitzgerald

Submitted to the Department of Aeronautics and Astronautics on May 23, 2012 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aeronautics and Astronautics

Abstract

Complex engineering systems are frequently exposed to large amounts of uncertainty, as many exogenous, uncontrollable conditions change over time and can affect the performance and value delivery of a system. Engineering practice has begun to address that need, with recent methods frequently targeting such techniques as uncertainty quantification or robust engineering as key goals. While passive robustness is beneficial for these long-lived systems, designing for passive robustness frequently results in sub-optimal point designs, as optimization is forgone in favor of safety. Changeability offers an alternative means for supporting value throughout a system's lifecycle by allowing the system to change in response to, or in anticipation of, the resolution of uncertainty, potentially enabling the system to perform at- or near-optimally in a wide range of contexts.

The state of the practice for valuing changeability in engineering systems relies mostly on options theory, which is associated with a number of assumptions that are frequently inappropriate when applied to change options embedded in systems. This has played a part in limiting the inclusion of changeability in fielded systems, as the standard techniques for calculating the benefits of change are often inapplicable and thus are less trusted than valuations of passive robustness. Without the ability to properly and believably value changeability, system designers will continue to look elsewhere for protection against uncertainty. A more generally applicable method for valuing changeability would greatly enhance the understanding and appeal of changeability early in the design process, and allow for the justification of its associated costs.

This research has resulted in a new five-step approach, called the Valuation Approach for Strategic Changeability (VASC). VASC was designed to capture the multi-dimensional value of changeability while limiting the number of necessary assumptions by building off of previous research on Epoch-Era Analysis. A suite of new metrics (including Effective Fuzzy Pareto Trace, Fuzzy Pareto Number, and Fuzzy Pareto Shift), capturing different types of valuable changeability information, are included in the approach, which is capable of delivering insight both with and without the computational burden of simulation. The application of VASC to three space system case studies demonstrates the large range of insight about the usage and value of changeability able to be extracted with the approach. Discussion about the strengths and weaknesses of VASC is included, particularly in comparison with Real Options Analysis, and a number of promising avenues for future improvements and extensions to VASC are identified.

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

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

Dedicated to my family, for their constant support, and my friends, whose support only wavered when I complained for too long.

Acknowledgements

I would like to thank many people who have contributed to this research over the past two years. Certainly, the leadership of the Systems Engineering Advancement Research Initiative (SEAri) is to thank for the opportunity to work on a research project with a goal as open-ended as "improve our ability to value change", and for the hours of prodding necessary to get a project like that on track with a first-time research assistant on point. So thank you **Dr. Adam Ross** and **Dr. Donna Rhodes**, for two years of research with significantly more freedom (if not free time) than I ever expected to receive. Thanks should also be given to the other great minds that see fit to grace SEAri students with their knowledge, such as **Professor Daniel Hastings**, who assuredly has less time to give than anyone appreciates, and **Dr. Hugh McManus**, who delivered some significant bursts of creativity and excitement into my research whenever he dropped by. SEAri students (past, present, future, and affiliated) were also a major source of input and feedback, without whom this research would have progressed significantly slower, and to a less satisfying result. Particular thanks are due to **Dan Fulcoly** (officemate/commiserator), **Clark Beesemyer** (gamemaster), **Nirav Shah** (idea man), **Brian Mekdeci** (perspective), and **Greg O'Neill** (suffered the most inane questions and misguided brainstorming when I was just starting out). Additional thanks go out to **Paul Grogan**, **Nicola Ricci**, **Matt Frye**, **Henrique Gaspar**, and **Hank Roark**, for their input over the past year or more.

An informal thank you is due as well to anyone who, over many years and iterations, has created or worked with the **X-TOS**, **Space Tug**, or **Satellite Radar** case studies. There are a lot of you, many who I have never met, but your hard work gave me a big leg up on all of this research.

And finally, to my **family** and **friends**: if you would call me either one of those, thank you! Being a graduate student would be a pretty big drag without all of you, and your support makes the work enjoyable. I promise to have some spare time eventually!

The author gratefully acknowledges the financial support for this research from the Systems Engineering Advancement Research Initiative (SEAri), a part of the MIT Sociotechnical Systems Research Center (SSRC). This research was also sponsored in part by the US Defense Advanced Research Projects Agency (DARPA). The views expressed in this thesis are those of the authors and are not endorsed by the sponsor.

Biographical Note

Matt Fitzgerald is Massachusetts born, raised, and educated. Originally from Dover, MA, he moved all the way to far-off Cambridge in order to attend the Massachusetts Institute of Technology upon finishing high school in 2006. After completing his Bachelor's degree in Aeronautics and Astronautics in 2010, he was still tired from his move four years before and decided to stay for a Master's degree in the same field, which hopefully he earns for this thesis. Most of his work has absolutely no relationship to anything he has done before, because he has a terminal case of "try everything, stop at nothing". Recently, he has worked with passive flow control for air-breathing propulsion, advanced compression technology, and now changeability valuation. The only thing consistent about his work is that he consistently makes it harder than it needs to be.

Matt Fitzgerald has only two goals in life, and they are exceedingly mundane: first, to create something pervasive enough that at some point he will meet a person who uses it, so he can say "Yeah, I'm the guy who invented that," and second, to become general manager of the Boston Red Sox. Completely reasonable, right?

In his free time, not that he has very much of it, Matt Fitzgerald is a man of simple pleasures. He likes laughing and seeks out opportunities to do so, usually by hanging out with his friends and riffing on whatever the topic of conversation happens to be (unfortunately, he likes laughing at his own jokes). He likes games of skill, as long as there is a winner and a loser (and he feels like he can win). He has impeccable taste in music, which he listens to constantly, and is that guy who sings to himself when he doesn't think anyone is within earshot. He also likes reading, but tries to avoid good books because they tend to creep into his studying time until he finishes them.

All things considered, Matt Fitzgerald doesn't dislike many things at all. Temperatures above 75 degrees make him a little cranky. People who take him too seriously can be frustrating. Most of all, though, Matt Fitzgerald hates writing about himself, especially in the third person, but does it anyway for the sake of posterity.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Changeability, like many of the so-called "ilities," is a system property that serves to improve lifetime value delivery. The "ilities" can be understood as characteristics of systems that enable value to be produced, but produce no value on their own. In particular, changeability corresponds with the ability to alter either the physical design parameters or operations of the system and can be leveraged in any of the lifecycle phases of common engineering systems: design, build, integration and test, and operate. For example, the ability to quickly redesign a particular subsystem of a rocket in the event of a requirements update would represent design-phase changeability, whereas the ability to burn fuel and adjust orbit altitude would correspond to operations-phase changeability.

Changeability is a potentially valuable addition to many engineering systems. Conceptually, the benefits of changeability are clear: opportunity can be seized and risk avoided by designing a system such that it has the capability to alter its form or behavior. This can enable the system to both avoid failure scenarios and also to maximize value under a variety of contexts. Common risks for engineering systems include schedule delays, variability in user needs or preferences, development of superior technology at a later date, and the much discussed and feared "unknown unknowns": unforeseen and unpredictable circumstances of any type, against which changeability has often been identified as a promising source of insurance (Baldwin, 2011). However, many of these risks can also become opportunities. For example, if a superior propulsion technology is created after fielding a space vehicle, there is *risk* associated with becoming obsolete in the face of new competition, but also *opportunity* for significant performance improvement if the vehicle was designed such that it can change to readily incorporate the new technology. As such, large, complex systems with long lifecycles are in a key position to benefit from changeability, as they are regularly exposed to significant amounts of uncertainty as they age over time and the world changes around them.

The inclusion of changeability, however, typically comes with associated costs, including ideation/development costs, physical build and inclusion costs, and potentially additional costs required to exercise a change. These costs are frequently well known, but the benefits of changeability are significantly harder to capture, especially when the system's performance or benefits are not readily monetized. This difference in ease of measurement has limited the consideration of changeability in the design process of many systems: when the costs are explicit and the benefits implicit, it becomes difficult to justify the inclusion of changeability-enabling features. Existing techniques for valuing the benefits of changeability are useful for some applications but suffer under the burden of a large number of assumptions that limits their general applicability. An improved means of valuation for changeability has the potential to allow changeability to be considered more effectively in the early design phase of systems that could greatly benefit from increased responsiveness to shifts in their operational context.

1.2 Scope

There are many potentially rich research avenues on the topic of changeability. The definition of changeability is far from concrete; many papers have been published, each attempting to chronicle the usage of the word and related terms, such as flexibility and adaptability, with the hope of clarifying and at least partially unifying the field of research. Also, the process of including changeability in a system features multiple tasks that could all be improved with additional research. These tasks form a conceptual "changeability lifecycle" that includes the steps of brainstorming and developing potential locations and enablers for changeability in the system, determining the value of the identified change options (and using this to justify their inclusion), and finally the management and usage of any available changeability. This process is shown in Figure 1-1.

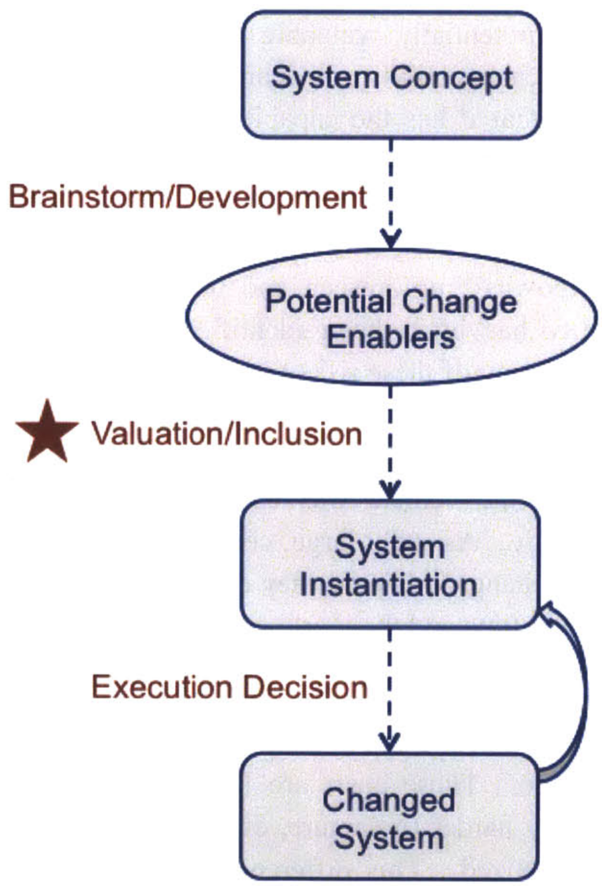


Figure 1-1: Conceptual Changeability "Lifecycle" – Research Scope Starred

The scope of this research is firmly within the area of changeability *valuation*, as indicated by the red star. The valuation process has received the bulk of the attention of changeability research, likely because it is the critical step towards justifying the inclusion of changeability in a system; without proper valuation, changeability will be fighting an uphill battle for consideration in the design process. While previous research on this topic has resulted in significant improvements, there are extant concerns and applicability challenges for the existing methods. Potential insights or improvements in the other areas of the "changeability lifecycle" will be secondary to the main goal of developing a more general means of valuation, relying on fewer assumptions.

1.3 Methodology

To accomplish the goal of this research, the following steps were performed: 1) literature review, 2) identification of specific research questions, 3) development of metrics and methods, and 4) the application of those metrics and methods to case studies. These steps are now described.

1.3.1 Literature Review

A literature review of the existing methods for valuing changeability is of paramount importance. As previously noted, there is a large body of research dedicated to the valuation of changeability. It is critical that an understanding of the different methods available to engineers at this time is achieved, paying attention to the strengths and weaknesses of each. Of particular interest are the weaknesses, because ideally any new techniques or recommendations resulting from this research will address gaps identified in the current body of knowledge and practice. The results of this research should be framed in such a way as to provide the most value to the practice of valuing changeability, by minimizing overlap with previous research work. The literature review also encompasses publications on other topics related to changeability, including the definition of the term and the creative development process, as these may inform possible improvements to changeability valuation activities.

1.3.2 Identification of Specific Research Questions

As stated, the goal of improving the process of valuing changeability is very broad. Using previous research to identify specific research questions is key for focusing this research on a beneficial and achievable purpose. These questions were identified by finding weaknesses in existing methods through the literature review, and prioritized by the amount of benefit resolving these weaknesses will provide to practitioners. By concentrating on addressing a small set of extant issues with changeability analysis, the likelihood of contributing materially to the field increases.

1.3.3 Development of Metrics and Methods

The main contribution of this research is intended to be the development of new metrics and methods for valuing changeability. These contributions are aimed at addressing the research questions from the previous step and being as generally applicable as possible. As the hope is that eventually the results of this research could be employed effectively by system designers on real projects, the metrics and methods should require as little overhead as possible, both in prior education and time or effort to apply.

1.3.4 Application to Case Studies

Metrics and methods developed over the course of this research were applied to a range of example case studies, in order to demonstrate their potential for uncovering insight about valuable changeability. These insights will hopefully allow a system designer to effectively distinguish value between potential change options and to allow a changeable system's performance value to be compared directly to passively robust systems, putting the two different design paradigms (i.e. changeability versus robustness) on equal footing in the design process. The demonstration of these insights using case studies also serves as a first pass at validation, which is often extremely challenging for human-in-the-loop processes, but should provide an excellent foundation for future research on the topic.

1.4 Thesis Overview

The following sections of this thesis cover the performance and completion of the above plan for this research. Section 2 details the literature surrounding changeability and comments on the variable applicability, strengths, and weaknesses of existing valuation techniques. Section 3 discusses the particular research questions identified as guiding concepts for the analysis created by this research, why they are important, and why previous techniques have failed to adequately address them. Section 4 explains the formulation of the set of metrics developed over the course of this research. Each metric is discussed both in mathematical terms and in "plain English" terms, to demonstrate how each captures a particular aspect of changeability that is of potential interest to system designers. Section 5 covers the details of the Valuation Approach for Strategic Changeability (VASC), which is the method created to utilize the changeability metrics and synthesize their insights with those of existing techniques. Each step of VASC has identified inputs, activities, and outputs for system designers to follow. Section 6 details the application of VASC to three different case studies, demonstrating the types of insight about changeability within a design space obtained with this method, beyond the capabilities of previous valuation techniques. Section 7 revisits the research with an extended discussion of the applicability, strengths, and weaknesses of VASC, along with potential avenues for future research and extensions to the VASC method. Finally, Section 8 concludes the thesis with final thoughts and lessons learned from this research endeavor.

2 Literature Overview

This section will briefly cover the current state of the field of changeability within engineering disciplines. Attention will be paid both to the meaning of the term, acknowledging the frequently disparate uses of "changeability" and similar words as abstract concepts, as well as common methods and formulations for quantifying and valuing the changeability of engineering systems. The review includes an introduction to Epoch-Era Analysis and the changeability metrics developed in association with it, and concludes with a summary of the potential room for improvement in the field of valuing changeability.

2.1 Changeability Terminology

2.1.1 Changeability and Related Terms

Changeability is an abstract design concept that is experiencing an increase in interest as engineering systems grow, both in budget size and system lifetime, demanding more emphasis on value delivery over time and within different contexts. The apparent potential performance advantages of changeable systems, the difficulty of designing a system fully robust to changes introduced across decades, and the increased cost of failure are driving the popularity of changeability and related terms as means for improving lifetime value delivery. The basic concept is simple and universal: changeability is nothing more than *the ability of a system to change*. However, the meaning of that phrase is subject to a wide range of interpretations, largely based on the application it is being used to describe. For example, one word with many similarities to changeability is *flexibility*; indeed, the words are used interchangeably quite often. Saleh et al. (2009) performed a survey of the use of the word "flexibility" in the literature for different fields, mainly managerial, manufacturing, and engineering design, while cataloguing the different meanings of the usage. Focusing here on the meaning for engineering systems, Saleh finds two distinct uses: one for flexibility *in the design process* and another for flexibility *in the design*. Even those subtypes of flexibility have been used differently, with design process flexibility being applied to both customers (flexibility in requirements specified) and designers (flexibility in constraints imposed). In-design flexibility is similarly split amongst various definitions, although most relate quite directly to the ability of the system to perform different functions. Also recently, in a progress report for research into the value of flexibility, Deshmukh et al. (2010) compiled an excellent summary of the definitions of flexibility used not in the colloquial sense, but by researchers in the field, comparing their similarities and differences taxonomically. Again, this revealed a significant amount of variation even between the scientists and academics performing research directly on the topic, though in this case the differences were largely related to the reasons for the change or whether or not the "speed" of the change (otherwise often referred to as *agility*) was a critical aspect. Any further interest in the breadth of potential understandings of changeability/flexibility is referred to this source.

2.1.2 Changeability in Engineering

Fricke and Schulz (2005) published a paper summarizing unifying principles behind designing for changeability (DfC) in engineering. In addition to covering why changeability is important in the modern technological environment, they identified four key features of changeability: robustness, flexibility, agility, and adaptability. All four of these features would fall under Saleh's *in design* heading, as they are used to understand the changeability of designs and not of the designing process. Using their definitions, robustness is the ability of a system to withstand environmental changes while flexibility is the ability to alter the system in response to those changes. The key difference between the two is the passive nature of robustness versus the active nature of flexibility, two different approaches to delivering value over time. Adaptability fits in between the two, as it is defined as the ability of a system to change itself in response to the environment; depending on the frame of reference this can be viewed as passive (no external control needed) or active (the system is changing). Agility is defined as the ability to change in a short time frame, and can be applied to both adaptable and external changes.

Ross et al. (2008) attempted to clarify the definitions of many of these "ilities" by creating a framework for a system change and then assigning particular features to correspond to the differences between otherwise-similar "ilities". His changeability framework models every potential system change as the result of three elements, an *agent*, a *mechanism*, and an *effect*, and is pictured in Figure 2-1. The change agent is what instigates the system change. The change mechanism is the means by which the system is able to change, be it an operational change or a part replacement or any other method defined by what is referred to as a *transition rule*, the algorithmic representation of a system modification. Then the change effect is the actual difference between the starting and ending states of the system.

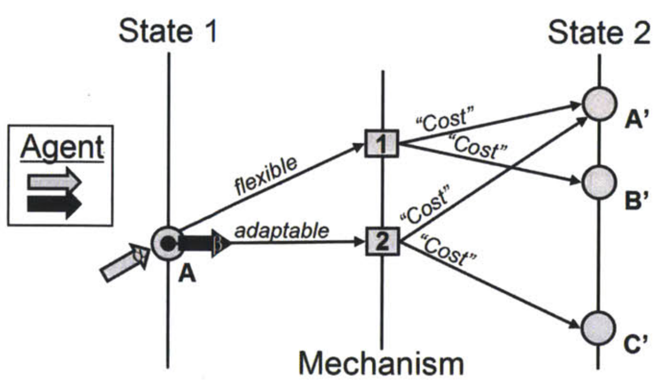


Figure 2-1: Agent-Mechanism-Effect Change Framework[1]

This framework is then used to provide clearer definitions of many of the abstract "ility" concepts that are so frequently used with different meanings in different papers. The change

[1] Ross et al., 2008

agent determines whether a change is *flexible* or *adaptable*; flexible changes correspond to agents external to the system and adaptable changes to internal agents. The change effect determines what type of changeability the transition rule offers and is separated into three categories, *modifiability*, *scalability*, and *robustness*. By their effect on the system parameters, robustness corresponds to constant parameters, scalability to changing parameter levels, and modifiability to changing the parameter set. This detailed breakdown of the change process is useful to this research and as such these "ility" definitions and the agent/mechanism/effect framework will be used in the remainder of this thesis.

2.2 Quantification and Valuation of Changeability

For changeability to be successfully included in the design process, a formal valuation method is necessary. The development of a robust, generalizable valuation method has proven to be a difficult process, often forgone in favor of more simple quantification procedures, and this challenge continues to limit the utilization of changeability in engineering today. Without the ability to ascribe proper value to it, changeability remains undervalued thanks to the better understood and more easily quantified nature of passive value robustness. This section will cover a sample of relevant changeability metrics and valuation methods in the literature of finance and different engineering disciplines, illustrating their basic conceptions of changeability and identifying their strengths and weaknesses.

2.2.1 Options and Real Options Analysis

An option, as traditionally defined in finance, is the right but not the obligation to purchase something at a later date. Classic options theory, in its current form, was pioneered by Black and Scholes (1973), with their discovery of a differential equation that is obeyed by all European options: options that must be exercised (or not) at a time specified when purchased. A number of different techniques for valuing options developed around the Black-Scholes formula, from analytic solutions to binomial lattice methods to Monte Carlo implementations. Real Options Analysis (ROA) was developed in an attempt to apply options theory to business and management situations. The goal of ROA, as opposed to the more traditional financial options, is to ascribe a monetary value to the flexibility provided by an option in a physical system, for example to invest in R&D for a new product, rather than simply projecting the most likely future outcome. Many of the standard finance and options-related techniques, such as Net Present Value (NPV) and Discounted Cash Flow (DCF) analysis, are applied to real options problems. However, the assumptions inherent in the Black-Scholes formula make its direct application to any sort of option beyond a standard financial option questionable. Myers (1984) gives an excellent summary of the challenges in applying financial theory to system decision-making strategic analysis, but makes a compelling argument for the value of approaching this category of problems with a financial viewpoint. Table 2-1 shows a few of the commonly known incongruities between Black-Scholes assumptions and non-financial options, with specific explanations for the discrepancies when considering options on engineering systems.

Table 2-1: Issues with Applying Black-Scholes Assumptions to Engineering Systems

Black-Scholes Assumption	Issue with Engineering System
European option (fixed exercise date)	Options are typically embedded and can be exercised at any time
Zero arbitrage[2]	Large scale systems are not being traded on a perfect market, and inefficiencies may exist
Geometric Brownian Motion of asset, with constant drift and volatility[3]	Long system lifetimes make it impossible to guarantee that these assumptions are true, or will remain true
Infinite divisibility of asset	Options are frequently binary, go/no-go operations
Existence of a risk-free interest rate	Engineering system value is frequently non-monetary, making this hard to define

2.2.2 "On" Options versus "In" Options

Recently, de Neufville (2004) made the distinction between options "on" and "in" systems. Options "on" systems are those of the more traditional managerial analysis, options to undertake or abandon projects. The ability to properly value options "on" systems allows business decisions to be made that defer costs into the future for potential later benefit. Most real options valuation techniques were created to value this type of option and are generally variations of financial options techniques. The alterations from the standard financial techniques are designed to address one or more of the concerns mentioned in the previous section using approximations. The choice of technique is largely driven by the application, as a given assumption will be more problematic for some applications than others.

Mathews and Datar (2007) developed one of the more intuitive methods for evaluating options "on" systems with their eponymous Datar-Mathews (DM) method. The DM method is analogous to the standard financial analysis practice of NPV, a technique that discounts future earnings at a fixed rate, collapsing all future value into a corresponding present value. NPV is typically performed with market projections that are deemed "most likely"; the DM method takes the underlying variables and, rather than setting them to be a constant "most likely" value,

---

[2] Arbitrage opportunities, a type of "market inefficiency", represent the chance to make a guaranteed profit by trading an asset between two separate markets for which it is priced differently. "Perfect markets" feature no arbitrage opportunities, because the natural order of financial markets is such that any arbitrage opportunities are quickly eliminated as both markets adjust their prices to prevent arbitrage.

[3] Drift (μ) and volatility (σ) are fixed parameters of Geometric Brownian Motion (GBM), a stochastic process based on the Wiener Process, W_t, describing the gradual change in mean and the variability, respectively. The Black-Scholes formulation requires that the asset being valued obeys GBM in order for the math to resolve properly. The GBM function is $f(t) = f(0) * \exp\{\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t\}$

samples them from a triangular distribution ranging from the most pessimistic projection to the most optimistic projection, with the maximum probability at the most likely point. Then a Monte Carlo procedure is employed, where each trial resamples from the triangular distributions. This results in a distribution of potential values for the option. If the average value of the option is greater than the purchase, it is likely to be worth investing.

When considering changeability of engineering systems, the type of option being considered is almost exclusively "in" the system. While "on" options do have relevance to engineering, particularly with regards to financing research and development for multiple potential systems, it is the "in" options that enable in-operation changeability. For that reason, methods for valuing "on" options will not be the focus of this research. For the remainder of this thesis, "option" stated alone can be assumed to mean an "in" option unless specified otherwise.

2.2.3 Valuing "In" Options

Methods to value options "in" systems have not been as thoroughly investigated as those "on" systems, but are still the predominant choices for valuing changeability of this type. The "in" options are distinguished by being internal to the design of the system being fielded. A typical example of this type of option would be the option to include a package on a satellite which would allow it to interface with other satellites as relay points instead of with the ground only; this will cost money up front but could enable future value. Wilds et al. (2007) performed real options analysis for an unmanned vehicle, demonstrating the applicability of the standard ROA decision-tree and binomial lattice techniques to an option allowing the fuselage design to accommodate two sizes of wings for different mission types. However, each of those methods suffers from some of the previously listed problems with applying Black-Scholes techniques to engineering system options, which can bring the results of the analysis into question. There have been many other demonstrations of valuing "in" options using classic options techniques in literature, including de Neufville, Scholtes and Wang (2006) with spreadsheet analysis and simulation, and Mun and Housel (2010) with Monte Carlo analysis.

Recently, Pierce (2010) developed a technique explicitly for valuing "in" options, which he named Variable Expiration. The technique uses a similar conceptual basis as the DM method, taking constants that underlie the Black-Scholes equation and changing them into random variables. The "variable expiration" refers to the expiration date of the European option inherent in Black-Scholes. This was previously mentioned as a serious conceptual break between the assumptions of Black-Scholes and the application of many "in" options, which do not have a set expiration date. The Variable Expiration technique sets the expiration date of the option to be a random variable, allowing for Monte Carlo samples to be drawn with the future value stream of the option beginning at different times in the lifecycle. This can be performed concurrently with the Monte Carlo sampling of any other value-determining variables, similar to the DM method. It also can utilize multiple discount rates, which allows value streams to be discounted differently depending on their nature and more accurately represents the range of risks that are not encompassed by the single Black-Scholes risk-free interest rate.

The Variable Expiration technique, however, is also not without flaws. Most noticeable is that the technique can only value options when viewed as add-ons to a baseline system architecture. This means that, while the analysis can be used to compare different options on the same architecture, it cannot be used to directly compare options on different architectures. Thus the practical application of Variable Expiration would likely have to occur late in the design cycle, after a system design has been selected, and the technique has limited ability to provide insight into overall system design. Another issue is that, while viewing the point in time at which an optional package begins to provide value (and the option is exercised) as a random variable is useful, there is no similar mathematical construct in the method for the potential cessation of usefulness at a later time. This limits the types of scenarios for which the method can appropriately value the option. For example, when valuing an optional package on a satellite that begins delivering value only when a different satellite is launched in the future (at an unknown time), the Variable Expiration technique assumes that the second satellite, once launched, will never break or be deactivated. That particular assumption may not seem unreasonable, but the restriction presents a much more serious problem when attempting to value an option which may deliver value for limited circumstances: for example, only when its country is at war, a condition that may occur and then end multiple times over a satellite's lifetime.

2.2.4 DSMs and the Placement of "In" Options

Silver and de Weck (2007) created the Time-expanded Decision Network (TDN) as a means to model system lifecycles with the end goal of identifying and valuing opportunities to insert real options into the system. The TDN is initially created as a network of system configuration nodes connected by change paths with associated switching costs. The nodes are then represented temporally, as pictured in Figure 2-2, and the maximum-value path between the nodes over time can be calculated for a given demand scenario. If different nodes or subsets of nodes are dominant design decisions under different demand profiles, the analysis can be iterated with the addition of potential real options to lower the switching costs between those domains in order to improve robustness against demand changes.

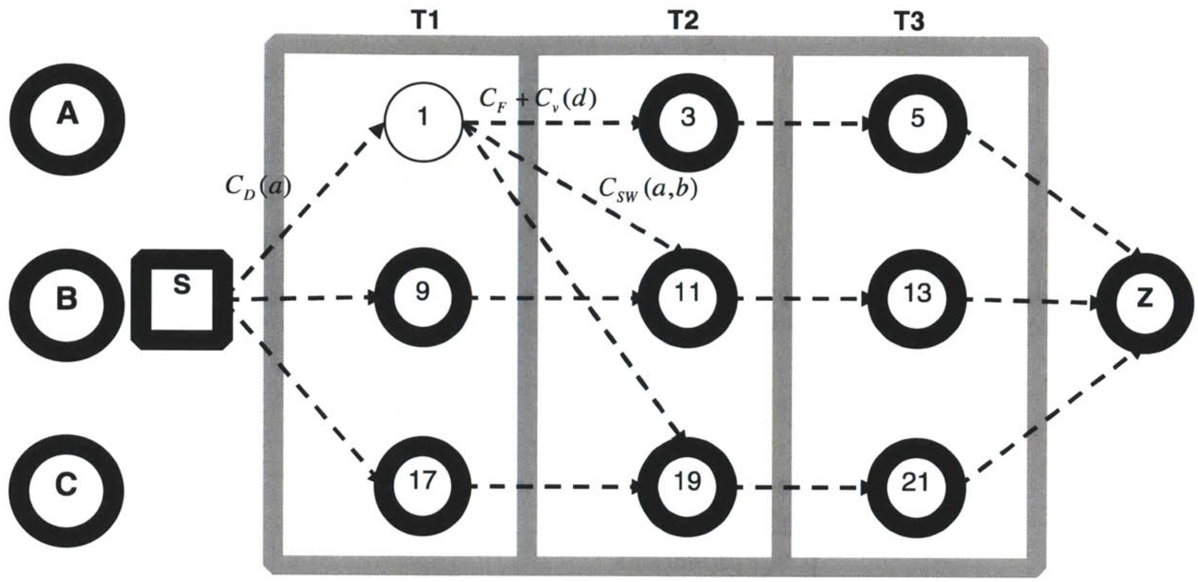


Figure 2-2: An Example TDN, With 3 Designs and 3 Time Periods[4]

The TDN is an appealing method mostly because of its ability to not only value potential options but also to identify where the insertion of an option would be most useful, assisting the human-centered process of ideation. However, the entire method is predicated on the same revenue and cost modeling necessary for the traditional financial calculations such as NPV, which is extremely difficult to model and justify for certain applications. It also works bottom-up from a complete demand profile over time, which could make it very challenging to draw effective conclusions for systems with high demand variability as many different profiles would need to be compared.

The Design Structure Matrix (Steward 1981), or DSM, is a technique for modeling the interdependencies of system parameters as they relate to each other. DSMs have long been used to quantify interconnectedness and, by extension, system modularity via clustering of dependent parameters. This form of modularity is heuristically associated with higher amounts of changeability in the system: if a system has more decoupled subsets of parameters, it is theoretically more easily changed later in the design process or over the course of the system's lifecycle. Kalligeros et al. (2006) demonstrate the application of an algorithm for the identification of the largest potential design "platform" for a project by analyzing the sensitivity-DSM, which includes the functional requirements of the system in addition to the design parameters. The "platform" is defined as the set of parameters insensitive to changing requirements, and is presumably to be used as a basis for members of a design family intended to work on a variety of problems. Danilovic and Browning (2007) noted that DSMs consider only inter-component relationships within a domain, and created the Domain Mapping Matrix (DMM)

---

[4] Silver and de Weck, 2007

to allow for identification and clustering of components between domains, revealing insight with a higher scope, at the architecture level. A similar identification-focused branch of DSM research was the creation of the Engineering Systems Matrix (ESM), a composite of DSMs of varying levels of scope, intending to model complex systems from the highest to lowest levels simultaneously (Bartolomei et al. 2006, Bartolomei 2007). The ESM can be used to identify "hot" and "cold spots" in the system architecture, where hot spots are those places where the insertion of options is potentially the most valuable. The ESM is shown in Figure 2-3, along with its relation to other well-known DSMs that are included in its composite structure.

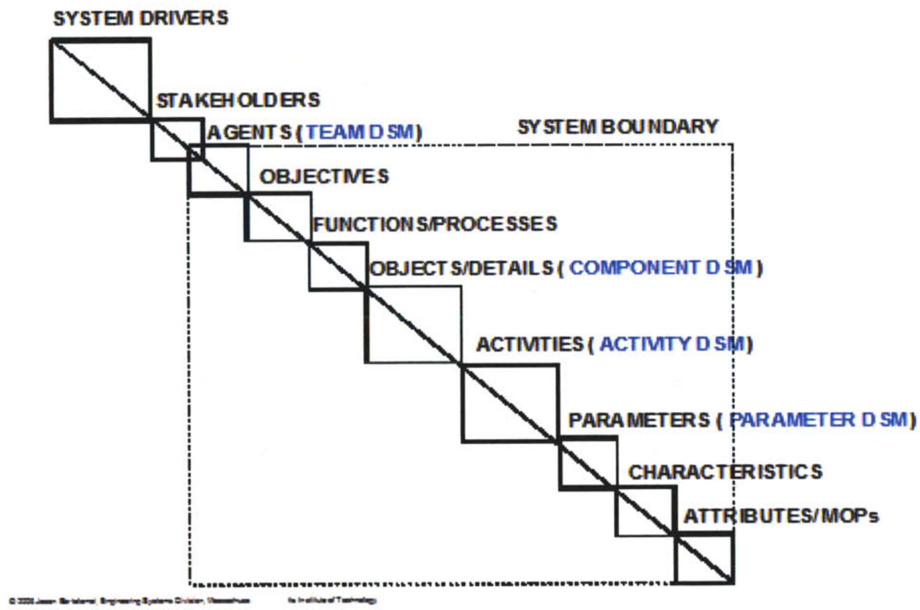


Figure 2-3: The Engineering Systems Matrix[5]

Mikaelian developed the Integrated Real Options Framework (IRF) with the goal of assisting the process of holistic enterprise identification and valuation of real options (Mikaelian, 2009; Mikaelian et al., 2009). To do this, the enterprise is modeled with a Coupled Dependency Structure Matrix (C-DSM), of which the ESM is one type. Furthermore, options are split into two parts: the option mechanism and option type. Three changeability metrics are identified to work on this framework, attempting to capture flexibility, optionability, and realizability. In this framework, optionability represents the number of different option types enabled by a mechanism, realizability is the number of mechanisms that enable an option type, and flexibility is the number of options that enable some goal or objective, as demonstrated in Figure 2-4. Then, these metrics are used to score a C-DSM's changeability, assisting the design process with heuristics such as 'objectives with low flexibility are good targets for insertion of additional options.' This framework is remarkably descriptive, but may suffer from over-complication, as

[5] Bartolomei et al., 2006

its definitions of the "ilities" are somewhat removed from the general discourse. Also, the metrics are quantifying the degree of changeability rather than valuing the presence of the options, which is useful for the stated goal of identification of mechanism/option inclusion opportunities but is often insufficient for the justification of the associated costs.

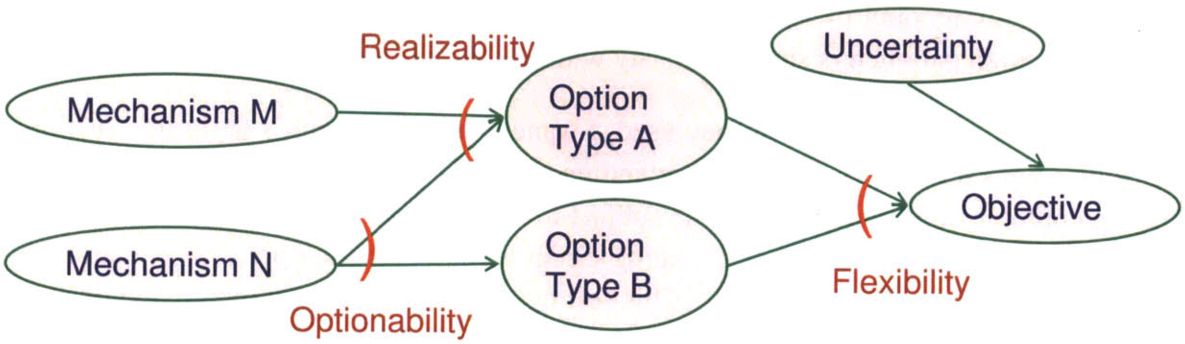


Figure 2-4: Conceptual Diagram for Flexibility, Optionability, and Realizability[6]

Sullivan et al. (2001) created a process that treats modules or clusters in DSMs as real options, allowing for the testing of a variable number of "experiments" on these modules, which may find an alternative set of parameters that increases system value. This technique does not rely on the classic financial options valuation methods, but instead assumes an associated cost and a probabilistic distribution of differential value (over the existing configuration) for any conducted "experiment" on a module. While this novel way of valuing options is potentially a good way to escape from the burden of monetization of value, it introduces a number of additional assumptions in the modeling of the distributions of these potential benefits from "experiments," which have to be set by the system designers and justified prior to their execution. Sullivan's research is also in the field of software engineering, for which this concept of "experimentation" has a relatively low cost, and it is not clear that this sort of option would present itself in other applications; for example, it is unlikely that a modular spacecraft project will have the budget to allow for the prototyping of multiple experimental modules.

2.2.5 Valuing Changeability without Options Theory

The complications inherent in employing options theory to value changeability have driven research into alternative methods as well. Swaney and Grossmann (1985) developed an index for flexibility, again used as a subset of changeability. In their context, flexibility refers to the operational flexibility in a chemical plant's processes. The underlying parameters of the process (temperatures, pressures, etc.) are only acceptable over limited ranges; this allows a multidimensional space of feasible plant operating conditions to be created. Any design point

[6] Mikaelian, 2009

must be located somewhere in the space. Since the parameters are only partially controllable and will fluctuate during regular operation, this space can be scaled in each dimension by the expected deviations in the corresponding parameters. Taking this scaled space and setting the design point to be at the origin, the flexibility index is then defined as the length of a half-side of the largest hypercube that can be inscribed in the space centered at the origin. In simpler terms, it is the maximum amount of variation, scaled by the expected variations, which can be applied to all of the design parameters simultaneously without the process failing.

Apparent in the application they used, Swaney and Grossmann's definition of flexibility might fall more under the general understanding of robustness, which accurately describes the ability to continue delivering value while subject to changing conditions. However, this concept of framing the ability to cope with changing circumstances, whether actively or passively, has easily drawn parallels to changeability. This sort of quantification technique, based on measures of the physical parameter space, offers a potential formula for changeability metrics.

Tackling the problem from the other end of the system, Olewnik et al. (2004, 2006) propose a flexibility metric that is measured in the "performance space" rather than the parameter space. This involves specifying a large number of potential designs and identifying the performance features that provide value. A Pareto set of all the designs can be generated for each pairwise combination of performance attributes. The flexibility metric is calibrated by the sum of the distances between the extreme points in each of these Pareto sets. An ideal, fully flexible design would be able to change into any of these potential designs, and thus the maximum possible flexibility is found by considering all of the designs (even if no design could practically switch between all of them). Any actual flexible designs are likely only able to switch between a subset of the entire design space, and their flexibility is quantified using only the spread across each two-attribute Pareto set achievable with that subset of designs. The design flexibility, normalized by the maximum potential flexibility, provides a means to compare different designs quickly and effectively by showing what fraction of the performance space they can span.

A key advantage for this method over parameter space methods is that the performance attributes are much more directly linked with value delivery than the environmental variables. However, there remains a large difference between a *quantification* metric and a *value* metric that is not addressed by simply quantifying in the performance space rather than the design space. One concern might arise when considering that the value derived from the different performance attributes is not equally weighted. Although not discussed by the authors, the one-dimensional distances used in their metric can likely be scaled by the appropriate weighting factor if that were the case. A larger problem arises when considering that the value delivered by the attributes is potentially nonlinear; this situation would negate the comparability of distances even within a single dimension, since the distances are measured in the performance space and not on a scaled value space. Olewnik et al. (2006) suggest using their metric in tandem with

NPV analysis and making tradeoffs to analyze the effect of reducing/increasing flexibility on cost effectiveness, but that involves bringing in all of the weaknesses of NPV analysis as well.

Mills (2009) constructed a method for evaluating enterprise architecture specifically for Air Force applications, by combining Decision Analysis and Value Focused Thinking (VFT) with the goal of identifying decision points in advance of their occurrence and utilizing that increased leverage to design better systems at the architecture level. Under his framework, entitled Value-Driven Enterprise Architecture (VDEA), the "ilities" take the role of the *value hierarchy* used in VFT, which is created immediately after the initial problem identification step in order to clarify value at the highest level of abstraction. The "ilities" are then measured for the value they contribute to a system architecture using the Department of Defense Architecture Framework (DoDAF).

Mills assembles a set of "ilities" into two hierarchies, labeled as the "System Effectiveness" and "Architecture Quality" hierarchies. Under this scheme, changeability (labeled as flexibility) is a Tier 2 "ility" in the "System Effectiveness" hierarchy, under the heading of the Tier 1 "ility" of capability. Mills notes that flexibility is particularly difficult to measure, stating that the value is abstract and thus must be quantified using a proxy of potential enablers of system flexibility. The DoDAF does not specify any architecture views specifically for flexibility, so Mills suggests the use of SV-8 (Systems Evolution Description) to identify any planned improvements to the system. This data is qualitatively used to judge the potential for adaptation in the system, and the ease with which it can occur.

VDEA as a tool offers a very top-down approach to designing large-scale system architectures, and as such is less focused on the method of valuation of each individual "ility" in its value hierarchy. Certainly, the qualitative approach to valuing changeability is appealing given the abstract nature of the concept, but a more rigorous and quantitative method would allow for greater repeatability and require less expertise for high-level system decision-makers to utilize. Also, the use of the DoDAF and the designated "ility" hierarchy may be sufficient to encompass all potential Air Force system architecting needs but it is not necessarily general enough to apply to other potential fields, for which a different architecture framework or "ility" hierarchy may be more appropriate.

2.3 Epoch-Era Analysis

The following sections discuss Epoch-Era Analysis (EEA), a piecewise-constant framework for modeling uncertainty over time. Although not directly linked with changeability, it has been extended with a significant amount of research into quantifying and valuing changeability.

2.3.1 An Introduction to Epoch-Era Analysis

Epoch-Era Analysis is a system design approach, developed by Ross and Rhodes (2008), designed to clarify the effects of time and context on the value of a system in a structured way. The base unit of time in the method is the *epoch*, which is defined by a set of variables specifying the context in which the system operates. These variables can define any exogenous factors that have an effect on the usage and value of the system: weather patterns, political scenarios, financial situations, operational plans, and the availability of other technologies are all potential *epoch variables*. The complete set of epochs, differentiated using these variables, can then be assembled into *eras*, ordered sequences of epochs creating a description of a potential progression of contexts over time, as shown in Figure 2-5. This framework provides an intuitive basis upon which to perform analysis of value delivery over time for systems under the effects of changing circumstances and operating conditions, an important step to take when evaluating large-scale engineering systems with long lifespans.

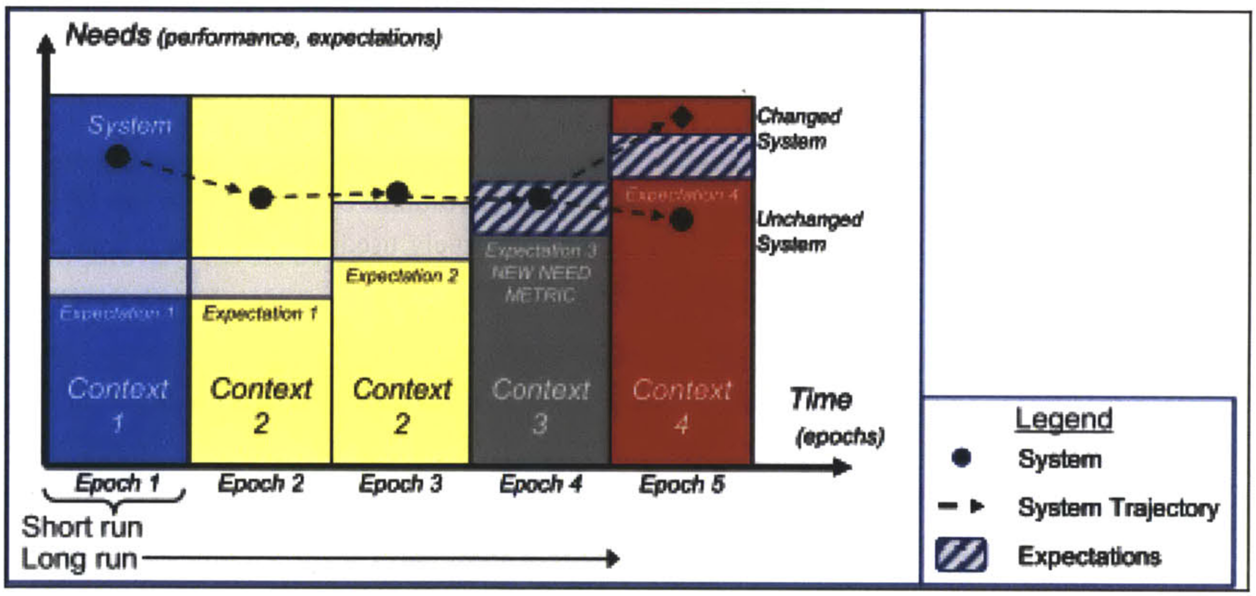


Figure 2-5: An Example Era, with Changing Needs and Contexts[7]

Epoch-Era Analysis was developed with the intent for it to be used in conjunction with Multi-Attribute Tradespace Exploration (MATE), which models large numbers of designs and compares their utilities, typically represented as combinations of nonlinear functions of performance attributes (Ross et al., 2004). MATE is a powerful method for conceptual system design, allowing for the evaluation and comparison of many different potential designs that could be chosen for building and fielding. A broad design vector enumeration is used to define many potential systems that are then modeled using a computer simulation, allowing for a more

---

[7] Ross and Rhodes, 2008

complete exploration of the entire design space, rather than the traditional engineering practice of focusing on a handful of potential designs, frequently locking-in some decisions prematurely, and selecting from amongst them.

In addition to its function as a temporal extension of the typically static-context field of tradespace exploration, Epoch-Era Analysis can be used as a framework for considering value-over-time regardless of the underlying methodology. Treating the passage of time as a stochastic sequence of static conditions can be used to extend other common engineering practices, including the investigation of a single point design for which time-dependent performance variables are present. This allows for a broader application of EEA to different types of engineering studies.

2.3.2 Changeability Metrics Developed for Epoch-Era Analysis

Using the agent-mechanism-effect framework of change referenced earlier, Ross (2006) developed the *outdegree* and *filtered outdegree* metrics as quantification of design changeability. For any given tradespace with defined transition rules, a *tradespace network* can be generated, which connects design points via directed paths signifying a change from one design to another according to a rule. Connected designs differ only by the effect of the change forming the path, and multiple paths can connect two points if there are different change mechanisms that cause the same effect. Outdegree (OD) is defined as the number of outgoing paths from a design; thus, a design's outdegree functions as a counting of the number of changes defined by the transition rules that apply to the design. Filtered outdegree (FOD) takes outdegree a step further by acknowledging that all changes come with associated costs, whether in time or money or some other commodity, and that certain changes will not be implemented if the cost is too great. Filtered outdegree is then the same as outdegree but with all change mechanisms with a cost above a set filter not counted. Adjusting the filter provides insight into the cost-to-quantity relationship of changeability for a design, as demonstrated in Figure 2-6: as the location of the acceptable cost threshold (gray vertical line) moves, the three designs may reorder in their relative changeability.

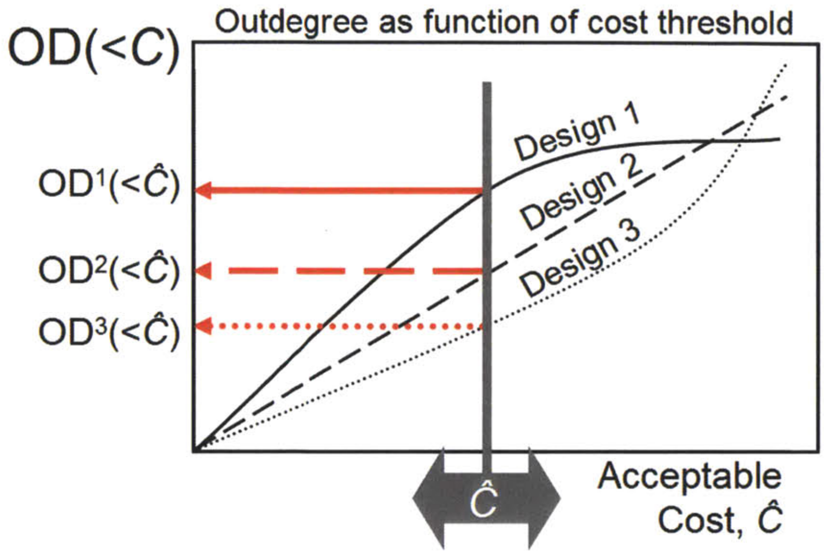


Figure 2-6: Filtered Outdegree, as a Function of Acceptable Cost Filter[8]

Filtered outdegree was later used as the foundation for an attempt to create a value metric for changeability, referred to as *value weighted filtered outdegree* (VWFO). This metric went through a number of different iterations (Viscito, Chattopadhyay, Ross 2009; Viscito and Ross 2009; Hastings, 2010), which are presented below:

$$(1)\quad VWFO_i^k = \frac{1}{N}\sum_{j=1}^{N}[(u_j^{k+1} - u_i^{k+1}) * Arc_{i,j}^k]$$

$$(2)\quad VWFO_i^k = \frac{1}{N-1}\sum_{j=1}^{N-1}[sign(u_j^{k+1} - u_i^{k+1}) * Arc_{i,j}^k]$$

$$(3)\quad VWFO_i^k = \frac{1}{N-1}\sum_{j=1}^{N-1}[H(u_j^{k+1} - u_i^{k+1}) * Arc_{i,j}^k]$$

N is the number of designs considered
k is the current epoch
k+1 is the next epoch in the era
i is the design under consideration
j is the destination design
u_i^{k+1} is the utility of design i in the k+1 epoch
u_j^{k+1} is the utility of design j in the k+1 epoch
$Arc_{i,j}^k$ is a logical value (1 if a path exists between i and j in epoch k, 0 otherwise)
Sign() is the sign function (-1,0, or 1 depending on the sign of the contents)
H() is the Heaviside step function

---

[8] Ross et al., 2008

Despite similar forms, these three equations measure distinctly different types of valuable changeability. Equation (1) equates the valuable changeability to the weighted average of the utility differences across all of the transition paths from a single design moving into a given epoch. Of major concern here is that the multi-attribute utility function is not linear, nor is it necessarily the same in different epochs; therefore those utility differences do not always measure the same difference in value and are not well suited for comparing designs between epochs.

Equation (2) shows an effort to return to the standard "path-counting" style of filtered outdegree. By introducing the sign function before the utility difference, the utilities only serve to determine if the path in question results in a positive or negative effect on utility, with a +1 or -1 added to the sum respectively instead of simply counting all of the paths with +1. A positive and a negative transition will cancel each other out in the formula; this does an admirable job reflecting the uncertainty of the effect of a change on value when transitioning into an unknown future epoch. However, if the changes will be implemented in a reactionary fashion, responding to epoch changes rather than anticipating them, then no negative-value change would ever be initiated. For this reason, Equation (3) uses the Heaviside step function instead of the sign function, counting only the positive-value changes. This difference between (2) and (3) shows that the agent-mechanism-effect description of change is applicable to different types of decision making, however an ideal metric would apply to both situations without requiring a change in the formula. The path-counting approach of (2) and (3) also does not have any weighting like Equation (1) does, which limits the effectiveness of the metric by assigning value only to the number of available change paths rather than including the effect of the changes on system value.

2.4 Conclusion from the Literature

Research into the nature of changeability and means to quantify and value it has been going on for approximately 30 years, and yet little has been universally accepted, even within the field of engineering. Many definitions of changeability continue to be employed, typically tailored to the field of whoever is writing the definition. Real Options Analysis is the most well-developed family of techniques designed to value changeability, but suffers from the necessary application of assumptions used in the financial options literature, which are not always appropriate for engineering systems. The degree to which these assumptions are inappropriate is only magnified when considering the "in" options that have great potential to increase the value of large, complex engineering systems with long expected lifetimes. Additionally, the process of monetizing value, which nearly all ROA techniques depend on in order to determine value, is frequently not justifiable for many engineering applications, which deliver value in non-revenue-generating ways. Existing techniques for measuring changeability without monetization or options theory have a variety of drawbacks associated with each method. Some of these techniques feature an overdependence on qualitative judgement; others are a more effective means of quantification than valuation. Since valuation is ultimately what is required in order to justify the inclusion of changeability in engineering systems against its associated explicit costs to build and use, a strictly qualitative or quantitative approach is not sufficient.

Thus, it is a reasonable conclusion that there is a need for improvement in the area of valuing changeability for real systems, particularly with regards to the development of a method that avoids the drawbacks of ROA but still has an intuitive connection to the value provided by the system. Such a method would dramatically improve system designers' ability to properly value the changeability offered by potential options "in" systems that do not produce value in easily monetized terms.

3 Research Questions and Key Concepts

This section will introduce the characteristics and capabilities identified, via previous literature and during the early stages of this research effort, as of particular interest for advancing the field of valuing changeability. It will then discuss two key concepts utilized as foundations for this research: Epoch-Era Analysis (EEA) and changeability execution strategies.

3.1 Research Questions

As previously mentioned in Section 1.3.2, the identification of useful properties for valuable changeability analysis techniques was determined to be of importance for focusing this research on specific research questions that would offer material improvement over existing techniques. The following three questions were proposed:

1. Can a method or set of metrics be created that value changeability with fewer inherent assumptions than existing methods?
2. Is it possible to analyze the multi-dimensional value of changeability using a single method, and without losing any fidelity?
3. Can metrics that accomplish the above goals also display qualities beneficial for generality, including dataset independence and context universality?

These research questions are discussed in more detail in the following subsections, and then previous metrics and methods covered in the literature review are compared in terms of their degree of success in accomplishing the corresponding goals.

3.1.1 Reduced Assumptions

One of the most common observations of existing changeability valuation methods as noted by the literature review was that the methods were frequently dependent on assumptions about the nature of the design problem in question. In particular, Real Options Analysis techniques based on the Black-Scholes formula depend on numerous assumptions, including Geometric Brownian Motion of underlying assets, infinite divisibility, and others covered in the literature review, all of which are often inappropriate for engineering applications. Other options-related methods rely on other simplifying assumptions, including things like known probability distributions for uncertainties, monetization of value, and fixed one-time change execution moments. Indeed, changeability analysis using ROA is potentially disregarded by many decision-makers *because* of its unrealistic assumptions (Shah et al., 2008).

Excessive and unrealistic assumptions therefore were identified as an outstanding problem for the field of valuing changeability. The ability to properly justify the inclusion of changeability in large engineering systems has been hampered by the presence of very specific and often inappropriate assumptions in existing valuation techniques. Reducing the number of assumptions necessary for assessing the value provided by changeability enablers has the potential to improve the likelihood of convincing decision-makers that such an investment is a prudent choice. Thus, any methods or metrics created for this research should rely on as few assumptions as possible. Particular effort should be given to avoid the assumptions made by ROA, in order to offer an alternative method to the large number of projects that are incompatible with those assumptions.

3.1.2 Multi-Dimensionality of Changeability Value

It is important to note that changeability offers value in two distinct ways: (1) the increase in system value resulting from a change, as well as (2) the number of options available, generating robustness to perturbations via breadth of choice and redundancy (a system with more options is more likely to have a high-value option available when an uncertainty resolves). These two aspects can be thought of more simply as the *magnitude* and *counting* value of changeability, coming from the use of changes and the presence of many change options, respectively. A complete accounting of the value of changeability would consider both of these sources; unfortunately, these two aspects do not provide value in ways that can be measured on the same scale, resulting in metrics that target only one or the other. For example, Filtered Outdegree is explicitly designed to "count" option paths, and its extension Value-Weighted Filtered Outdegree does not adequately resolve the magnitude value, as previously referenced in the literature review. Similarly, the process of monetizing value and scoring designs based on NPV or variants thereof, as in many options-based methods, has the side-effect of aggregating out the counting value of changeability, so that it cannot be examined.

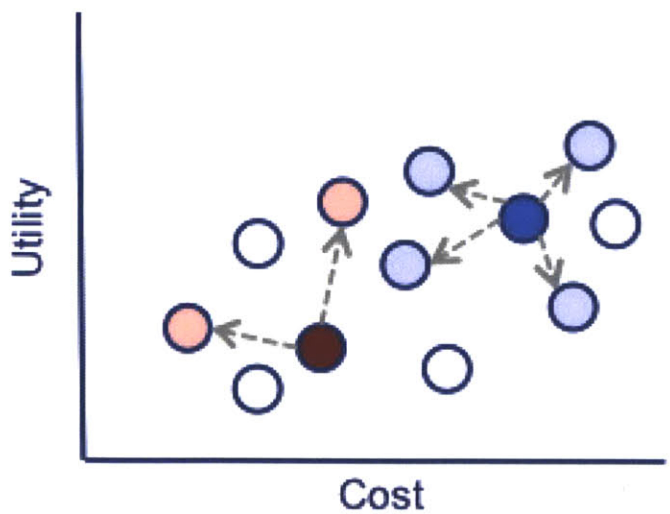


Figure 3-1: A Notional Tradespace Illustrating High Magnitude Value (Red) vs. High Counting Value (Blue)

To illustrate these concepts, consider the above diagram. Figure 3-1 shows a notional tradespace, with two designs highlighted. The red design has two potential change paths, one of which results in a substantial increase in utility. As a result of this large-benefit change path, the red design would be considered to have high *magnitude* value of changeability, as it can use a change path to significantly improve itself. On the other hand, the blue design has four potential change paths, none of which produce nearly as much utility benefit as the red design's change paths. However, the blue design's changeability is not without value. The presence of four change paths suggests that, were this tradespace to rearrange itself due to a shift in context or needs, a change path to a high-benefit end state is more likely to belong to the blue design than the red design. This resilience to uncertainty is a result of the *counting* value of having multiple changes available to the system.

For these reasons, investigating the possibility of calculating magnitude and counting value in a single analysis was determined to be of considerable importance to this research. A full understanding of the potential benefits of including changeability in a system cannot be achieved without looking at both of these sources of value. The ability to compare designs using both of these aspects is also potentially an excellent way to discriminate between otherwise similar designs.

3.1.3 Dataset Independence

A useful quality for metrics to possess is for their scores to be *independent* of the set of designs being considered. If a design's score is a function of the other designs under consideration, the use of that metric adds a large burden of proof on the design team to show that any "good" design is good regardless of the other options under consideration. This burden is particularly important for metrics valuing changeability, as many stakeholders require concrete evidence of the added-value gains of changeability options in order to be convinced to fund them. Independence is also a useful property even for applications where value relative to alternatives is acceptable, as it allows the set of designs of interest to be modified during analysis while maintaining metric stability. As an example of a project where relative value is acceptable, think of a launch vehicle selection for a fully completed payload cleared for deployment: there is a fixed set of existing alternatives for launch vehicle and one must be selected (the project is not optional as the payload must be put into orbit). There is no need to prove a baseline "goodness" for the selected vehicle as long as it is superior to the others, because all alternatives are included in the decision and the decision to choose none is not an option. Under conditions like these, independent metrics are less critical but still superior to dependent metrics.

The desire for independent metrics is not a new one, as establishing a concrete value over a relative value is always preferable. The monetized values resulting from Real Options Analysis are independent, as each option is valued based on only its own performance under uncertainty. However, many of the tradespace changeability metrics are *not* independent, but instead are explicitly calculated via the relationships between design points. For example, Outdegree is dependent on the number of other designs in the tradespace, as it will increase or decrease as other potential end states are added or removed from consideration. Normalization helps alleviate this problem, as the metric then measures the fraction of the tradespace accessible via changeability; the score of a design will still change as other designs are introduced, but the implications of being able to reach 20% of the design space are more intuitive than the ability to change via 40 paths, which carries different value in a tradespace of 50 designs than one of 5000.

Thus, dataset independence was chosen as a key feature for any metrics created from this research. The presence of independent measures of value will allow for the establishment of value relative to a fixed baseline rather than relative to other designs. Independence also allows for the freedom to change the set of designs under consideration in the middle of the analysis process without mandating the recalculation of the metrics, saving time and effort. A method capable of independent value judgements in both magnitude and counting value has, however, yet to be created; the feasibility of such a method remains an open research question.

3.1.4 Context Universality

Another useful property for changeability metrics is the *universality* criterion, which stipulates that a score of X for the metric is equally as good or bad as a score of X in a different context. Without this distinction, it is extremely difficult to effectively use the metric to address problems of uncertainty. For example, if a study were utilizing Epoch-Era Analysis as its framework for uncertainty, scores in a non-universal metric become incomparable across different epochs: once the context changes, the numerical result of a non-universal metric can remain the same but imply a different actual value.

Universality is a key problem for any metrics that use multi-attribute utility (MAU) as a quantification of value, as MAU is neither on a ratio scale nor ordinal between epochs under different preferences, making any utility score not equivalent in value to the same score in a different epoch. Previous valuable changeability metrics such as Value-Weighted Filtered Outdegree suffer from this weakness. The monetization of performance utilized in ROA also fails to meet this criterion, as the "value of a dollar" varies in the eye of the beholder, and is also sometimes subject to nonlinear preferences in a similar way to MAU.

In general, when any modeled uncertainty changes, only scores of universal metrics can be compared directly to their previous values. This is a potentially critical distinction to make when valuing changeability, as differences in reported value before and after a change are likely of great interest. Therefore, attempting to include universality in any created metrics was chosen as a specific objective out of the third research question.

3.1.5 Comparison of Existing Techniques

Directly quantifying the number of assumptions for different methods is ultimately not productive, as assumptions will vary in the severity with which they depart from reality. Furthermore, this severity is heavily dependent on the application, and mostly a qualitative judgement. However, the existing changeability valuation methods can be examined with an eye for their ability to distinguish magnitude and counting value, and whether or not they are utilizing independent and universal metrics. This comparison is shown in Table 3-1.

Table 3-1: Comparison of Existing Changeability Valuation Methods

	Magnitude	Counting	Independent	Universal
Filtered Outdegree[9]	No	Yes	No	Yes
Normalized Filtered Outdegree	No	Yes	In the limit of large tradespaces	Yes
Value-Weighted Filtered Outdegree[10]	Binary good/bad	Yes	No	No
Performance Space[11]	Yes	No	No	No
Parameter Space[12]	No	Maximum range only	Yes	Yes
Real Options Analysis (NPV)	Yes	Implicitly in aggregation – cannot interpret	Yes	No

None of these methods possess all four of these properties. The ability to distinguish both magnitude and counting value is also not achieved adequately in both dimensions for any method, and the only method to use a metric both independent and universal is the parameter space hypercube method from Swaney and Grossman (1985), which does not deliver an accurate report of either dimension of value. This data reaffirms the observation that it is extremely difficult to fully value and understand system changeability using a single metric, perhaps suggesting that a suite of metrics designed to work together is a more feasible plan for a complete analysis. Regardless, if this research is capable of producing a method that can deliver all four of these properties, then it will have made a material contribution to system designers' tools for valuing changeability.

3.2 Key Concepts

Valuing changeability requires two key preliminary steps. First, there must be some model for time and uncertainty, as changeability only has value in their presence (otherwise, a simple performance optimization would suffice). Second, there should be a model for the usage of the changeability, in order to understand how changeability will be utilized when exposed to uncertainty. To accomplish these tasks, this research will leverage previous work in Epoch-Era Analysis, and employ a new concept of a *changeability execution strategy*. The following sections will clarify why these two solutions were chosen and the benefits of their application.

---

[9] Ross, 2006
[10] Hastings, 2010
[11] Olewnik et al., 2004, 2006
[12] Swaney and Grossman, 1985

3.2.1 Tradespace Exploration

This section will provide a brief introduction to the paradigm and terminology of *tradespace exploration*. The remainder of this thesis is couched in tradespace language, as it is the method used for describing the concepts and performing the case studies, and thus a basic understanding of it is essential. However, note that tradespace exploration is not a required aspect of either Epoch-Era Analysis or changeability execution strategies, but is rather used here in order to make them easier to understand. For more information on non-tradespace variants of this work, refer to Section 7.3.1.

Tradespace exploration is a technique for assisting conceptual system design. The "traditional" design approach is to select a single or small set of potential designs, create detailed models for their performance, and then conduct local "trades" on the design variables to search for potential nearby improvements. This practice risks the premature lock-in of particular system traits, resulting in a reduction of the design space. Tradespace exploration is an alternative practice that focuses on a more thorough consideration of the design space, one that will even consider sub-optimal designs (however "optimality" may be defined), with the understanding that certain valuable behaviors may be not captured by the chosen value metric (Ross and Hastings, 2005). In Multi-Attribute Tradespace Exploration (MATE), the often chosen value metric is a utility function, created as a combination of different performance attributes which are rated from zero, defined as minimally acceptable, to one, at which point no benefit is gained from performing better (Ross et al., 2004).

The large set of designs evaluated during this process is called the *tradespace*. Tradespaces are typically viewed with a graph showing each evaluated design as a point on a plot with utility-cost axes (hence, the common term *design point*), as this view is able to display critical information about the relationship between the design space and two important decision metrics succinctly and effectively. Viewing the tradespace on alternative axes is sometimes useful for uncovering different relationships amongst designs. When using the agent-mechanism-effect change framework, a tradespace can be transformed into a *tradespace network* upon the definition and application of change mechanisms (Ross and Hastings, 2006). A change mechanism represents a way in which the system is capable of changing one or more of its design variables; for example, an adjustable seat in a car can change positions to accommodate differently sized drivers, or a surveillance aircraft can change its patrol route to cover different territory. Change mechanisms can be used to algorithmically generate change paths between different design points using logical statements referred to as *transition rules*, as illustrated in Figure 3-2. The connected designs differ only by the design variable(s) able to be altered by the change mechanism, and this difference is the effect of the change. For the purposes of valuing changeability, these paths in the tradespace network must be considered, as they represent the choices available to each design.

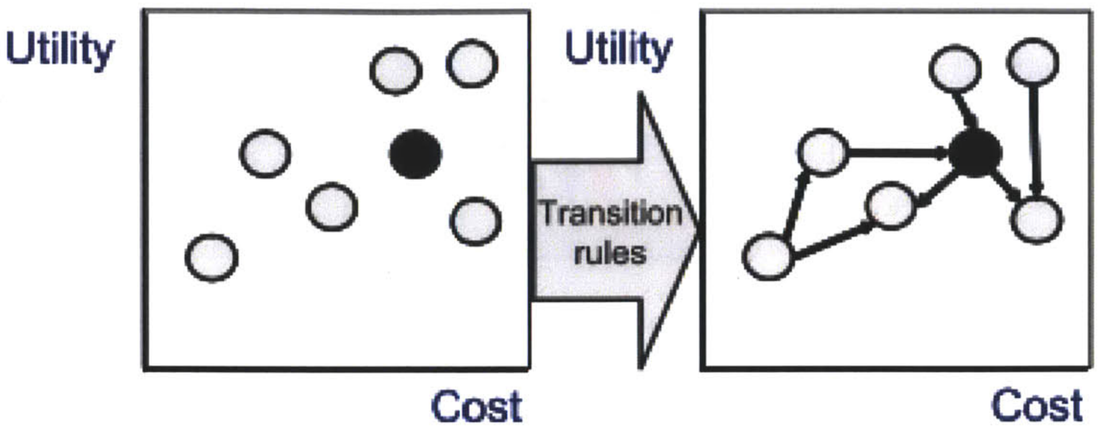


Figure 3-2: Creation of a Tradespace Network[13]

3.2.2 Epoch-Era Analysis

As discussed in Section 2.3, Epoch-Era Analysis (EEA) is a framework for modeling uncertainty over time (Ross and Rhodes, 2008). Using it, the short-term contexts that a system can be operated in are referred to as *epochs*, defined by vectors of *epoch variables* encompassing exogenous variables that affect the system and stakeholder preferences. Epochs are then assembled into sequences called *eras*, which model potential long-term futures that the system may face.

EEA provides a means of considering uncertainty that aligns itself with the tradespace exploration paradigm, establishing a large number of alternative contexts: potentially up to the point of a full-factorial combination of the entire epoch variable vector. This is different than the majority of previous methods, which either model a single exogenous uncertainty in detail (typically simulating over some kind of probability distribution) or use a small set of uncertainty "snapshots" to perform scenario planning. This tradespace-esque view of uncertainty, shown in Figure 3-3, is part of the reason that EEA pairs so well with Multi-Attribute Tradespace Exploration (MATE), as they combine to consider an extremely wide range of design and uncertainty, casting a wide net to find any potential valuable design solutions.

[13] Ross and Hastings, 2006

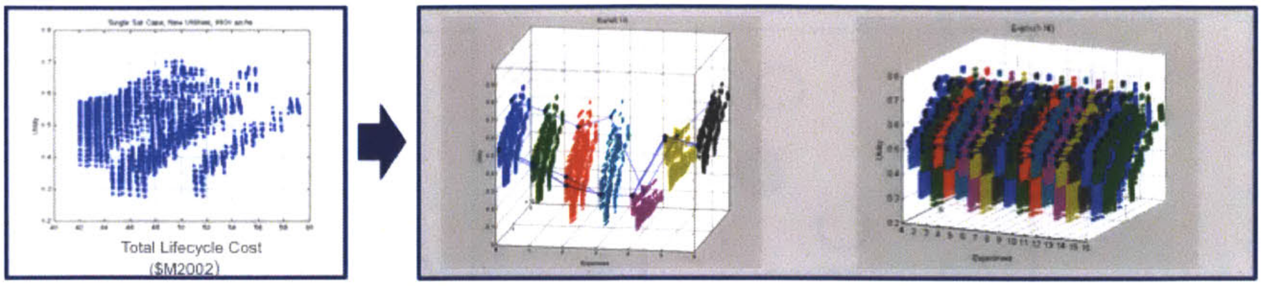


Figure 3-3: EEA "Tradespace" of Tradespaces to Model Uncertainty[14]

For these purposes, EEA is an excellent choice to model uncertainty. The ability to consider a wide range of both designs and uncertainties will help us compare passively robust and actively changeable designs at the same time. Additionally, the use of epochs as small static-context chunks allows us to investigate the value of changeability in a system under a large number of different circumstances without requiring the generation of probability distributions for the variables or any continuous-time mathematics.

3.2.3 Changeability Execution Strategies

The value of changeability is difficult to calculate largely because it is *latent*. Changeability does not deliver value when not in use; in fact, the presence of change enablers may even *decrease* system performance by moving the system away from optimality due to drawbacks such as added weight or size. Thus, much of the research on valuing changeability has focused on accounting for the uncertainty associated with its use. ROA and the parameter space method are good examples of valuation techniques with a heavy emphasis placed on the uncertainty surrounding the system.

However, changeability is not simply a product of uncertainty, but also of *usage*. Consider again the conceptual changeability "lifecycle" shown in Figure 3-4. This problem is not one of observe-then-value, because value must be determined before it can be observed. Benefit is only received at the execution point; that is, changeability provides value only when it is used. Thus, a method for finding the true value of changeability should not be considering all *potential* change paths, but rather only the ones that are *used*. The difficulty, then, is that to perform a proper valuation, *how* changeability will be used in the future must be determined.

[14] Ross, 2006

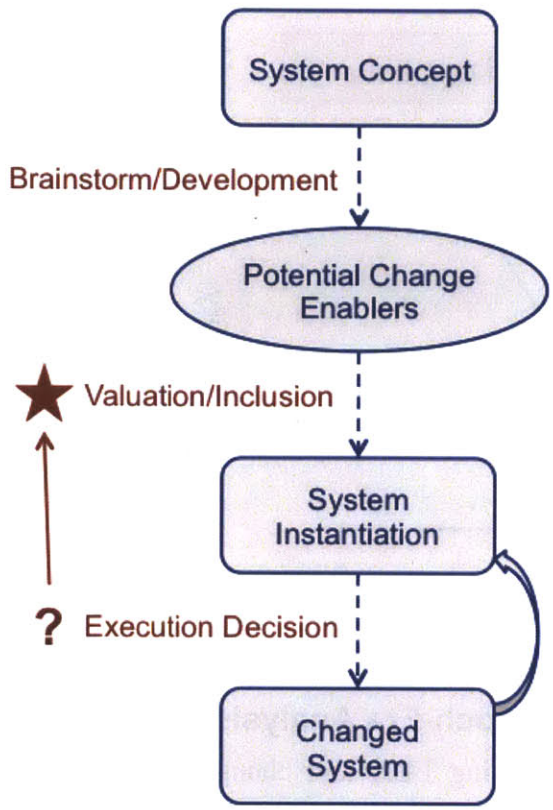


Figure 3-4: Conceptual Changeability "Lifecycle" – Execution Drives Value

To this end, a *changeability execution strategy* (or simply a *strategy* from hereon) can be defined, that specifies how the system stakeholder intends to utilize any available changeability. This strategy can range from the simple (maximize utility at any cost) to the complex (execute change targeting the available design with highest predicted lifetime value, but only if utility falls below a certain threshold and design increases in cost efficiency, and under certain conditions changes are not allowed). Employing a given strategy thins out the multitude of *possible* options down to one *selected* option for a given design, and it is this selected transition that should be used to value the system's changeability.

Figure 3-5 shows the application of two example strategies to a single design with multiple change paths in a simple notional tradespace network. The strategy is represented as a logic statement that is used to select from amongst the available paths the one that the stakeholder would choose to use. Frequently, it will be of interest to consider multiple strategies, as the different strategies can be compared for their relative effectiveness at increasing system value and system stakeholders may not be sure what logic they *would* or *should* use.

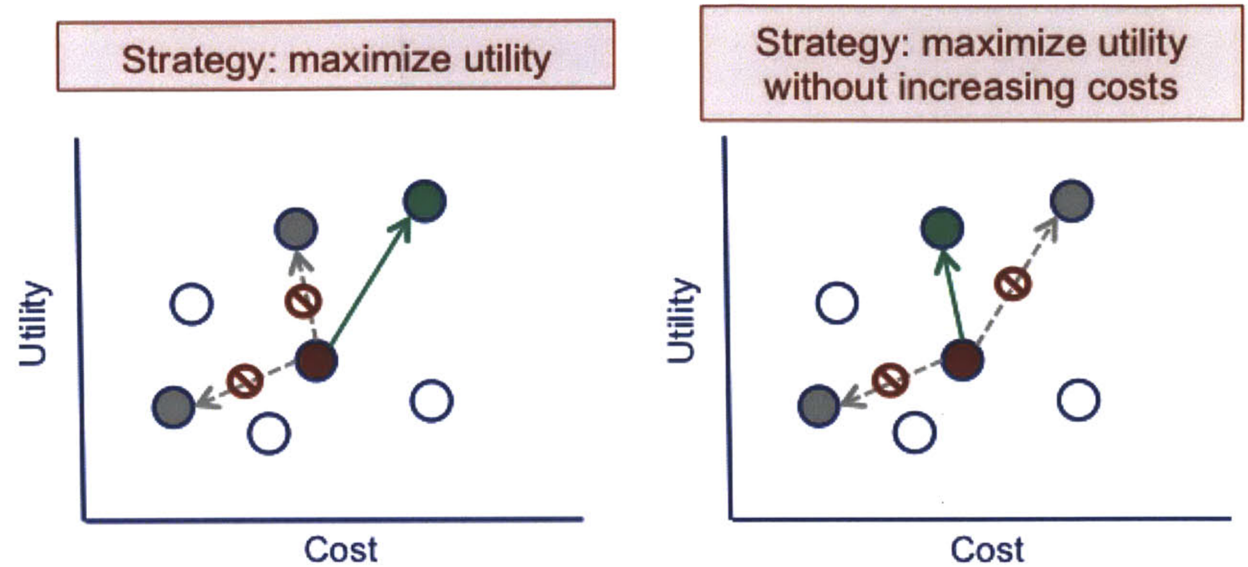


Figure 3-5: Two Example Strategies Applied to a Design

3.2.4 Strategies in Epoch-Era Analysis

The true benefit of using EEA and changeability strategies comes from using them together. When applied to each design in a tradespace network, a strategy simplifies the network by restricting the outgoing number of paths from each design to one (or zero, if no change path is present or deemed acceptable by the strategy), and this occurs for each epoch. This is demonstrated for a single epoch tradespace in Figure 3-6.

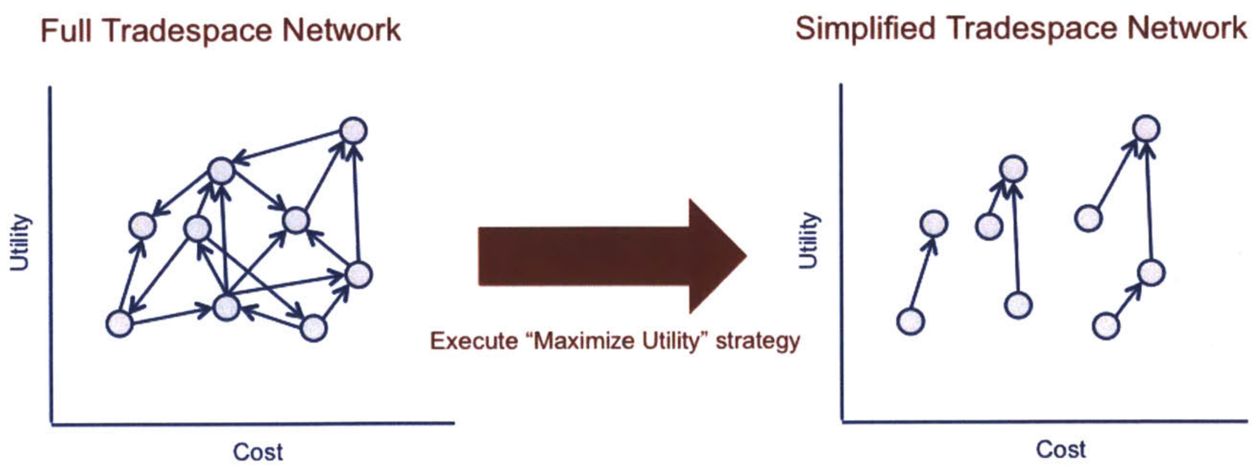


Figure 3-6: Tradespace Network Simplification via Strategy

For representation in equations, this thesis will refer to the selected *destination*, or *end state*, of design d as d^*, where d^* is defined separately for each epoch and can be equal to d if the design does not change. When multiple strategies are under consideration, different indicators should replace *, such as d^{*i}, to clarify which strategic end state is being referenced.

The use of strategies with EEA allows the reconciliation of the magnitude and counting values of changeability. The single selected path from each design in each epoch will be valued for its magnitude: the benefit gained from executing that design change. The counting value of the changeability options for a design is revealed when looking across the epoch space: when confronted with a full variety of changing contexts, designs with more options will tend to have a higher magnitude value in more of those contexts. If a design with many change paths does *not* result in a higher magnitude value across a range of epochs, then it should have a low counting *value* in spite of its large quantity. By intelligently probing these two sources of information, system designers can extract insight about the total valuable changeability inherent in a design. This ability is enabled through the use of changeability strategies within the EEA framework, and represents a large step forward over the models used in the various methods of Table 3-1, none of which were capable of separating and demonstrating both magnitude and counting value.

What remains then, is the creation of metrics with which to value the strategy-selected changes across the epoch space created by EEA. As expressed by the research questions, ideally the developed metrics will be both independent and universal. The next section details the formation of a set of metrics that can be applied across the epoch space in order to calculate the value of changeability in different ways, which may be of interest to system designers.

4 Multi-Epoch Analysis Metrics

Multi-Epoch Analysis is a subset of Epoch-Era Analysis dealing only with the epoch space and not considering eras; again, the "uncertainty tradespace" paradigm is an apt one. The benefits of performing Multi-Epoch Analysis are both intuitive and computational. First, it is of interest to system designers to understand the performance of a system (and its changeability) across the entire range of potential uncertainty, without necessarily delving into time-ordering effects. Multi-Epoch Analysis can provide insight into the types of changeability executed in response to uncertainty used by a given system under a range of possible strategies. In addition to being informative, Multi-Epoch Analysis can also come at a lower computational cost than simulation or Monte Carlo techniques, which require a large number of samples and function calls in order to achieve statistical significance. Thus, Multi-Epoch Analysis is a strong foundation upon which to create new valuable changeability metrics, improving on existing techniques (Fitzgerald et al., 2011). The following subsections describe metrics for use in Multi-Epoch Analysis, beginning with some metrics of interest from previous EEA research and then moving on to new metrics designed to target valuable changeability.

4.1 Previously Created Metrics

The most common metric for measuring passive robustness in EEA is *Normalized Pareto Trace* (NPT; Ross, Rhodes, Hastings, 2009). NPT calculates the fraction of epochs in the epoch space for which a given design is on the Pareto front. A design in a tradespace is on the Pareto front (with respect to at least two objectives) if it is not dominated by any other designs: that is, there is no design that performs better in each objective. This is a weaker statement than one of true Pareto efficiency in the multivariable optimization sense, as a tradespace offers only discrete design points and thus points on the Pareto front may not be truly Pareto efficient if the design enumeration is not at a fine enough granularity. However, for well-constructed tradespaces (with a sufficient fineness of samples and appropriate design variable ranges), the concept is effectively the same. If presented without descriptors of the objectives, the "Pareto front" is nearly always the cost-utility Pareto front, and this is the case for NPT. NPT captures the passive robustness of a system by calculating how likely it is to be an optimal tradeoff of cost and utility for a randomly selected context.

Smaling (2005) developed the idea of a "fuzzy" Pareto front for tradespace problems, which allows for a buffer from the true Pareto front defined by a percentage of the range of the objectives. Smaling's goal was to expand the set of designs under consideration in a tradespace study by considering near-efficient designs that fall within the bounds of model uncertainty. This fuzziness concept was extended to NPT, creating fuzzy Normalized Pareto Trace (fNPT), which counts epochs for which a design is within a certain percentage fuzziness of efficiency. This can be invaluable for locating designs that may not be on the Pareto front very often or at all, but yet are nearly cost-utility efficient in a larger number of epochs, rather than only finding strictly efficient designs.

The previously created multi-epoch metrics for changeability (outdegree, filtered outdegree, and value-weighted filtered outdegree) were covered in detail in the literature review; refer back to Section 2 for more information on how to calculate these metrics and what they are measuring. Note that, as the concept of a changeability strategy is a new one, these metrics do not utilize strategies or strategic end states and instead look at all potential change paths from a design.

4.2 Fuzzy Pareto Number

Consider again the concept of fuzzy Pareto efficiency. The distance from the Pareto front is calculated as a percentage of the range of the cost/utility data; for example, all 1% fuzzy Pareto efficient designs will be less than or equal to 1% of the range of costs more and 1% of the range of utilities less than a truly (0%) Pareto efficient design. Although this was originally intended to broaden the range of designs considered efficient, this concept can be adapted for a different purpose. The Fuzzy Pareto Number (FPN) of design d is the smallest percentage K for which the design is in the fuzzy Pareto set P_K, as in Figure 4-1.

$$FPN(d) = \min\{ K \mid d \subset P_K \}$$

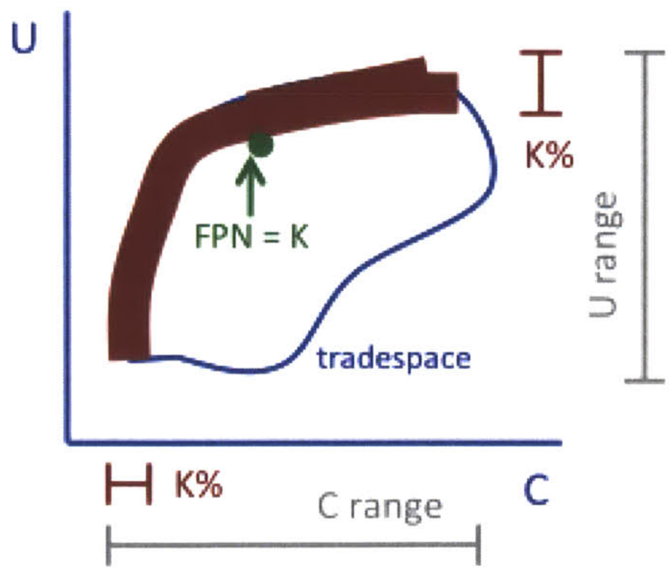


Figure 4-1: K% Fuzzy Pareto Set

FPN is a measure of cost/utility efficiency calculated for each design in each epoch and for which smaller is better, with a minimum of 0 and a maximum of 100. In this thesis FPN will be presented only as integers (whole percentages of the data range), but it can be calculated to any desired fineness, limited only by the computation time necessary for additional detail. FPN will be used as an indicator of design value and efficiency in other metrics. Note that, unlike many other measures of value such as multi-attribute utility, FPN is universal in scale; that is, a design with an FPN of 3 in any epoch is within 3% of cost-efficiency, regardless of the shape of the tradespace, the functions used to define it, or the design's exact position. FPN is also independent of the other designs in the tradespace with the assumption that the tradespace has been sufficiently sampled to approach the true Pareto front[15]. Under this assumption, the FPN of a design will not change with the addition or removal of any points, as they will not affect the Pareto front that defines P_K. Although at first glance this seems like a large restriction, it is essentially in line with the assumptions made with drawing any conclusions from tradespace exploration: that a given tradespace is accurately representing the design space well enough to understand the tradeoffs between cost and utility. Also, small changes in the Pareto front will result only in small changes in FPN for some of the designs in the tradespace, so FPN is not overly sensitive to small fidelity improvements. Only the omission of a radical design option that vastly changes the Pareto front would completely invalidate FPN data and, in the event one of these designs *was* discovered, FPN could simply be calculated again.

Alternatively, a multivariable optimization could be run to numerically determine the true Pareto front of the performance space and that Pareto front could be used to judge FPN, making all calculations independent of the enumerated designs, but expending the effort to do this may defeat the purpose of performing tradespace exploration. Further comments on non-tradespace extensions of this research are located in Section 7.3.1.

4.3 Effective Normalized Pareto Trace

With the definition of a strategy, it is now clear that some designs will choose to change when confronted with particular epochs. Why then should designs be judged based on their *own* FPN, given that they will change in response to an epoch shift? Thus, an *effective* version of the Pareto trace metrics (eNPT, efNPT) is defined, which considers not the FPN of the design d itself in each epoch, but the FPN of that design's strategically selected end state d^* for each epoch. This allows designs that frequently change in response to epoch shifts to be graded not on their baseline performance, but on their changeability-enhanced performance.

$$eNPT(d) = [\ \Sigma_{epochs}\ 1\{FPN(d^*)=0\}\ ] \div N_{epochs}$$

$$efNPT(d,K) = [\ \Sigma_{epochs}\ 1\{FPN(d^*)\leq K\}\ ] \div N_{epochs}$$

---

[15] FPN is also scaled by the utility and cost ranges of the tradespace, so outlier designs (particularly those with significantly high cost that are not Pareto efficient, because the other dimensions of these axes are bounded) can change the reported FPN values, and thus ideally are avoided and removed from the tradespace before calculating FPN in order to promote range stability. However, even with outlier designs, the conclusions drawn from FPN should remain relevant. Some consideration been given to replacing FPN with a similar metric that uses the cost and utility ranges of Pareto efficient designs only (rather than the entire tradespace), the benefit being that it becomes completely independent of outlier high-cost designs. Drawbacks to this metric would include the lost connection to the existing formulation of fuzzy Pareto sets, an unbounded ceiling (instead of the upper bound at 100), and that it would be technically undefined for tradespaces with only a single dominant design. Future research could attempt to apply this metric and determine if the benefits outweigh the costs.

A design that scores high in eNPT or efNPT could be said to be "frequently cost efficient across the space of potential future scenarios *when considering its planned usage of changeability*". Note that these metrics consider passive robustness and changeability-enabled robustness simultaneously, as passively robust designs (which will not change very often due to naturally high cost efficiency) will be graded on their own FPN for most epochs. This is a powerful way to aggregate two methods of value delivery, active and passive, into a single value metric.

4.4 Fuzzy Pareto Shift

While the previous metrics attempted to quantify a measure of robustness across the uncertainty space that acknowledges changeability, clarifying the magnitude of the value of a design's selected changes is likely also of interest. For example, two designs may score identically in eNPT but derive vastly different value from their respective change options because of differing amounts of passive robustness versus changeability. To gain these insights, the Fuzzy Pareto Shift (FPS) distribution of each design can be analyzed.

$$FPS(d) = FPN(d) - FPN(d^*)$$

FPS is defined as simply the difference in FPN of the pre- and post-change states (*d* and *d**) for a given design in a given epoch. Thus, a design with an FPN of 25 that transitions to a design with an FPN of 4 would have an FPS of 25-4 = +21, as in Figure 4-2. The "shift" in Fuzzy Pareto Shift represents an increase or decrease in cost efficiency as the result of executed changeability. An increase in FPN would result in a negative FPS; this is meant to signal a loss of efficiency, but does not necessarily signify that a mistake has been made, as the implementation of many strategies (for example, utility maximization) will sacrifice efficiency for gains in other objectives. An epoch in which no transition is made will have an FPS of zero, as the initial and final states are the same.

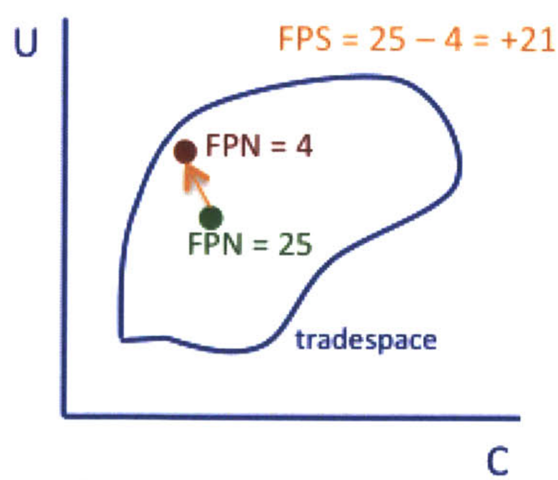


Figure 4-2: Example FPS of a Designated Change

Because it is defined separately for each epoch, a design's FPS is best viewed as a distribution across all epochs. This distribution curve can be compared against other designs for an intuitive understanding of the relative frequencies of different magnitudes of changeability value occurring in each design across the epoch space: the counting value of the design's changes. When breaking down this distribution into representative statistics, preference should be given to order statistics (minimum, maximum, median, percentiles) over averages; the distributions are often heavily skewed by positive and/or negative outliers, making the distribution mean ill-suited to summarizing the central tendency of the design's performance.

To illustrate the benefits of analyzing FPS, consider the following updated version of Table 3-1:

Table 4-1: Updated Valuable Changeability Method Comparison Table

	Magnitude	Counting	Independent	Universal
Filtered Outdegree[16]	No	Yes	No	Yes
Normalized Filtered Outdegree	No	Yes	In the limit of large tradespaces	Yes
Value-Weighted Filtered Outdegree[17]	Binary good/bad	Yes	No	No
Performance Space[18]	Yes	No	No	No
Parameter Space[19]	No	Maximum range only	Yes	Yes
Real Options Analysis (NPV)	Yes	Implicitly in aggregation – cannot interpret	Yes	No
FPN	Not a changeability metric	Not a changeability metric	Yes w/ converged Pareto front	Yes
FPS (FPN + strategy)	Yes	No	Yes w/ converged Pareto front	Yes
FPS distribution (FPS + EEA)	Yes	Yes	Yes w/ converged Pareto front	Yes

As Table 4-1 shows, FPN satisfies the Universal and Independent (with one assumption) criteria. Combining FPN with the definition of a changeability strategy allows for the calculation of FPS, which quantifies the magnitude value of the selected change. Finally, viewing the FPS distribution across the entire epoch space can demonstrate the counting value of

[16] Ross, 2006
[17] Hastings, 2010
[18] Olewnik et al., 2004, 2006
[19] Swaney and Grossman, 1985

changeability for the design. The combination of FPN, EEA, and changeability strategy has allowed us to check off all four desired aspects of this changeability analysis, breaking new ground in the field.

FPS is explicitly calculating the efficiency effect of executed changeability on the system. Thus, interpreting it directly as the *value* of changeability is not always appropriate. This is a critical observation related to the nature of valuing changeability as determined by a strategy: how can you use any one metric to calculate value regardless of strategy, when the success of a strategy is determined with regards to its own decision logic? If, for example, the strategy of choice is to maximize utility at all times, shouldn't some utility-based metric be used to determine the value of changeability? This presents two key issues: using a different metric for each strategy would make it extremely difficult to compare different strategies, and customizing the metric to the strategy's objective is ultimately uninformative, as each strategy is "best" at achieving its own goal (by its very definition). Thus, FPS is used as a stand-in for overall value, as cost-utility efficiency is of interest to system stakeholders in all but the most unusual situations, and it can be compared on an independent and universal basis between designs, epochs, and strategies.

4.5 Available Rank Improvement

Unlike the previous metrics, Available Rank Improvement (ARI) does not depend on a strategy, but rather presupposes an attempt to maximize utility. Recall the definition of a change mechanism in the agent-mechanism-effect framework: a change mechanism is defined as a single means for a design to change; for example, a modular payload bay to swap payloads and thrusters to alter orbit characteristics are two potential change mechanisms for a satellite system (Ross et al., 2008). ARI is calculated for each change mechanism (r) separately, as the maximum possible improvement in utility rank-ordering achievable using only that change mechanism. The term d' represents all design points able to be reached, starting from design d using only mechanism r.

$$ARI(r,d) = Rank(d) - \min\{Rank(d')\}$$

ARI is an imperfect metric, as it requires a tradespace and depends heavily on the chosen enumeration of designs, but serves adequately as an indicator of potential achievable value enabled by the inclusion of a particular change mechanism. A "strategic" version of ARI can be calculated by swapping out d' with the strategic end state d^* used in the previous metrics and removing the min{} function, but this form is not recommended. The reason for restricting ARI to a "maximize utility" improvement ranking is similar to those discussed about the relationship between FPS and strategy: since a strategy promotes a goal that is not necessarily represented by utility, it is not particularly relevant to compare the strategic end states of a general strategy with the associated gains in utility (FPS avoided this distinction with the reasoning that design cost-utility efficiency is always of interest, regardless of strategy). Thus, ARI is best employed to represent what utility gains are made "available" by a change mechanism, and the other metrics are more appropriate for evaluating executed strategic transitions.

This approach to a change mechanism-centric valuation of changeability is a new one, fundamentally different than that of previous techniques, which focus on designs. The ability to value the changeability provided by single change mechanisms or path enablers can allow system designers to make more educated decisions about their inclusion or removal in the system.

4.6 Differential Value with Removal Weakness

Sticking with the idea of change mechanism-centric analysis, a system designer may be interested in dissecting the valuable changeability of a design, attributing it appropriately to different change mechanisms. In particular, if a design with a large number of change mechanisms (and significant value derived from changeability) is only utilizing a single mechanism, this is of twofold importance. First, unused change mechanisms may represent an opportunity to reduce the complexity of the system, saving development and build costs, or potentially redirecting them to other tasks. Second, the critical change mechanism is a candidate for increased attention as it drives a large portion of system value. It is potentially a good decision to either establish redundancy or otherwise increased assurance of operation, as a failure of that particular change mechanism could render the system valueless and unable to change to improve in certain epochs.

Thus, a means of quantifying the importance of a change mechanism for delivering system value is of great interest. This task can be accomplished by *removing* any single change mechanism (or set of mechanisms), recalculating the strategy-selected change paths *without* considering the paths of the removed mechanism, and then comparing the *differential value* of the design before and after the removal using the other multi-epoch metrics. If the differential value is high, the removed mechanism can be considered critical and a candidate for improvement; if it is low to non-existent, the removed mechanism can be considered for removal from the actual system.

In particular, the differential value in FPS is referred to as the *removal weakness* of the design to a particular mechanism. Removal weakness can be calculated for a design across each epoch and plotted as a distribution, just like FPS. Removal weakness distributions with a large amount of weight near zero indicate a relatively unimportant mechanism, and as the distribution shifts into negative values (indicating a decrease in efficiency effects of changeability) the mechanism is indicated as more critical. An example removal weakness calculation (for the same design and epoch as in Figure 4-2) is shown in Figure 4-3.

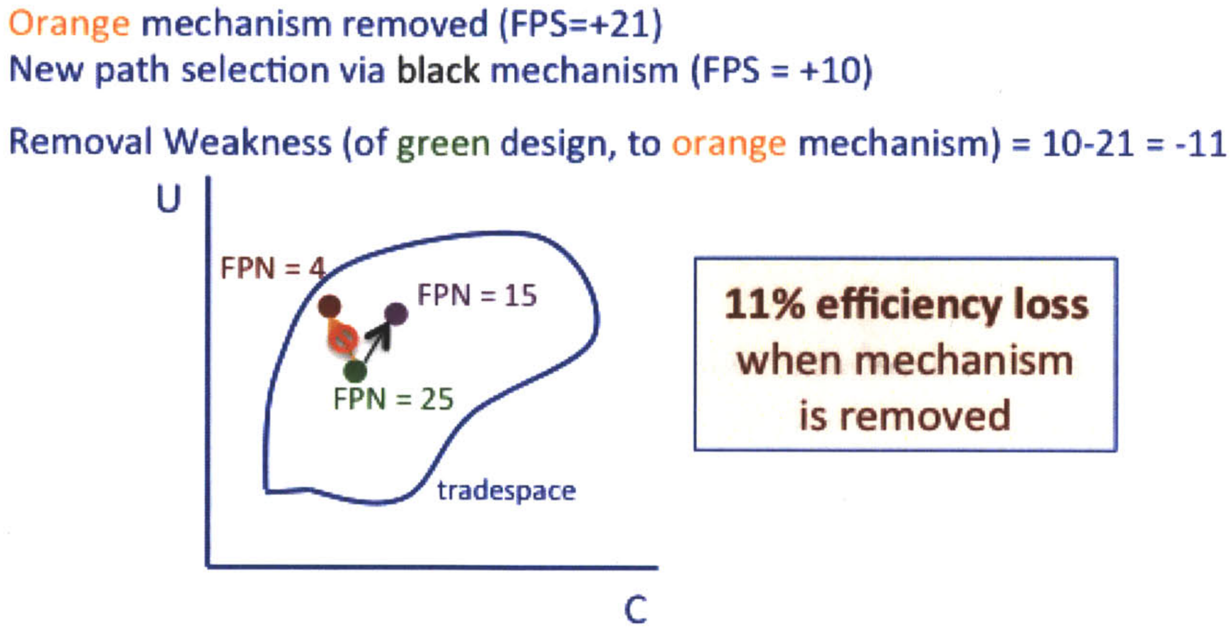


Figure 4-3: Example Removal Weakness Calculation

4.7 Metrics Summary

This section has introduced four metrics, designed to calculate different aspects of valuable changeability in the multi-epoch space of EEA, in conjunction with the definition of a changeability execution strategy. These metrics are summarized in Table 4-2.

Table 4-2: Summary of Metrics Developed in this Research

Metric Acronym	Stands For	Targeted Insight	Definition
eNPT, efNPT	Effective (Fuzzy) Normalized Pareto Trace	Robustness via changeability	Fraction of epochs in which design's changed end state is on the (fuzzy) utility-cost Pareto front
FPN	Fuzzy Pareto Number	Design efficiency	% margin needed to include design in the fuzzy Pareto front
FPS	Fuzzy Pareto Shift	Efficiency value of a change	Difference in FPN before and after executed change
ARI	Available Rank Improvement	Potential utility value of a mechanism	# of designs able to be passed in utility using a change mechanism

Because these metrics all measure different aspects of valuable changeability, it is not immediately apparent how they should be used together. For this reason, a structured approach for utilizing these metrics to generate insights about the changeability of a system was determined to be a critical addition to the research. The following section will describe the resulting approach and how these metrics fit into a complete valuation method.

5 Valuation Approach for Strategic Changeability (VASC)

So far, this thesis has discussed the creation of a modeling basis for the valuation of changeability and a number of metrics which can be applied to that formulation in order to quantify and value the changeability of designs. However, it was decided that a formalized "approach" for performing this analysis would be useful for synthesizing the concepts and metrics discussed in the previous section, and ideally promote guidance and repeatability when applying these metrics to different problems. Goals for this approach included:

- Assist in uncovering difficult-to-extract information about the system from design and performance data and the resulting changeability metrics
- Identify valuable designs created with different design paradigms (e.g., robustness, changeability)
- Identify the changeability strategies that enable system value
- Assess which change mechanisms deliver the most value, and their criticality to overall system performance under a given strategy
- Establish cost/benefit tradeoffs for the inclusion/exclusion of changeability in a design

The resulting approach is entitled the Valuation Approach for Strategic Changeability (VASC). Described below as a five-step process, VASC is designed to structure the analysis of valuable changeability for system designers (Fitzgerald et al., 2012). The steps are as follows:

1. Set up data for Epoch-Era Analysis
2. Identify designs of interest
3. Define changeability execution strategies
4. Perform Multi-Epoch Analysis
5. Perform Era Simulation and Analysis

VASC is also able to be applied iteratively, successively refining the analysis until the system designers are comfortable with their understanding of the potential for valuable changeability in the system. The flow of data in VASC is illustrated in Figure 5-1.

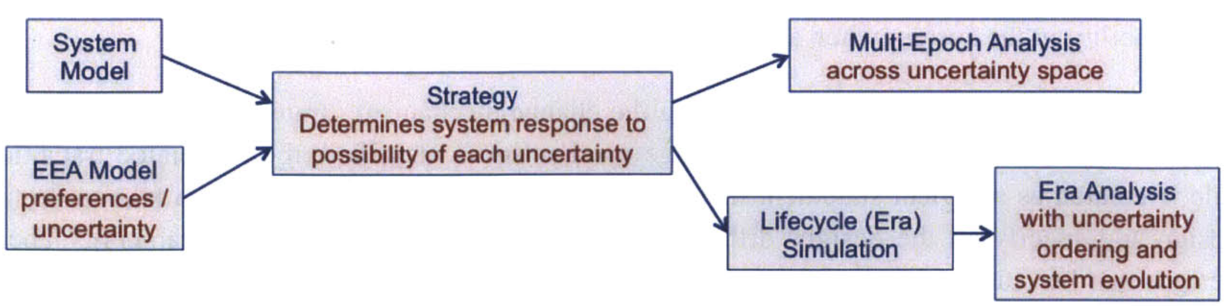


Figure 5-1: VASC Data Flow

The remainder of this section details each step of the VASC process, noting the inputs, activities, and outputs associated with each step. The application of VASC to example case studies will be demonstrated in Section 6.

5.1 Setting up Epoch-Era Analysis

Step 1 puts the case in question into the Epoch-Era Analysis framework, allowing for piecewise consideration of time in sequences of segments with constant context and needs. The main task for this step is identifying all of the data needed to perform the analysis and putting it in the proper form. Design variables, change mechanisms, desired performance attributes, stakeholder preferences, and context variables are all created at this point, using the expertise of both system stakeholders and experienced engineers familiar with the system. If a model for evaluating the static performance of any given design in any given context does not already exist, it should also be created in this step. Outputs include the design vectors and epochs intended to be analyzed, transition matrices indicating all the paths for each considered change mechanism by their starting and ending design point and associated cost, and the FPN for each design/epoch pair. For more information on setting up EEA, refer to the AIAA publication of Ross et al. (2009), which details the initial steps of their Responsive Systems Comparison (RSC) method, including the framing of a problem in EEA form. A summary of Step 1 is shown at the end of this section in Table 5-1.

When working with previously created data sets, typically the design vector and performance model will already exist. In this case, the remaining tasks are identifying the uncertainties and the change mechanisms. Again, the opinions of the stakeholders or domain experts are invaluable for these tasks, but sometimes these opinions will not be available (e.g., due to schedule constraints or the exploration of a new field with no known experts). If attempting to perform these tasks without this expertise, consider establishing parameters of the performance model as epoch variables (varying them over a reasonable range to model potential uncertainty) and identifying the design variables that should be the most simple to change for a completed system (establishing change mechanisms that allow this variable to be altered). This level of detail may seem minor, but can be sufficient for generating insight about valuable changeability in the system when a more detailed model is unable to be created.

Transition matrices are created using the change mechanism / transition rule relationship mentioned in Section 3.2.1. When a change mechanism is identified, the associated transition rule is created as a logical statement that can be implemented as an algorithm to take an initial design and specify all the designs differing only by the appropriate design variable(s). These designs can be reached by the initial design via an execution of that mechanism; i.e., a change path exists between them. The transition matrix compiles these potential change paths for the entire design space, indicating the costs of changing *from* every design *to* every design reachable via the given change mechanism, thus taking the form of a square matrix of size equal to the number of designs in the tradespace.

It is worth discussing here the benefits of considering multi-arc change paths. Under the agent-mechanism-effect framework, each change path is associated with a single mechanism and connects two designs. It is not inconceivable that a stakeholder may desire to utilize multiple change mechanisms at the same time, changing the system in multiple ways at once in order to best satisfy his preferences. This is because two change mechanisms when considered together may be able to deliver more value than when considered apart. This idea of *coupling* effects of change mechanisms is demonstrated in Figure 5-2.

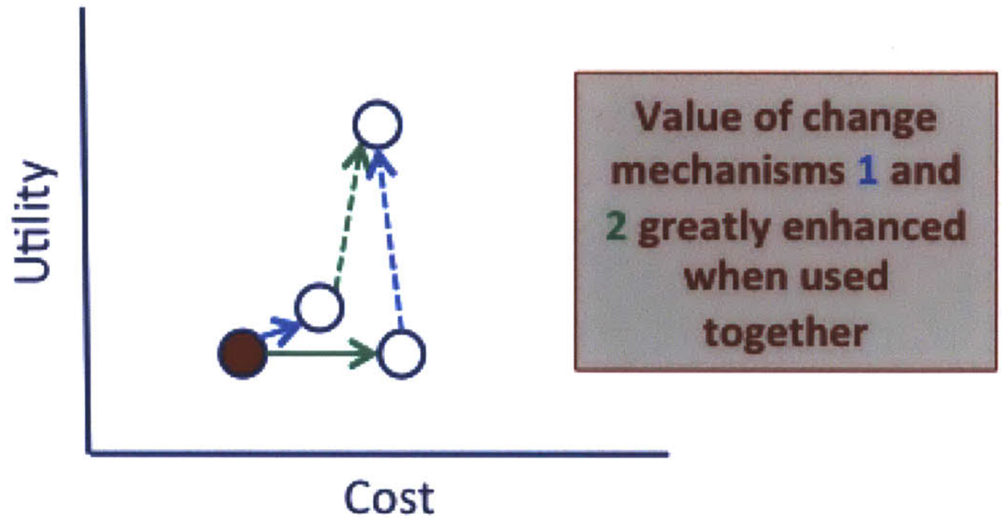


Figure 5-2: Example Change Mechanisms Creating Additional Value When Used Together

To consider multi-arc change paths, the paths enabled by different mechanisms must be manipulated together. A simple computer algorithm was created which identifies all potential multi-arc paths (limited to some specified maximum number of arcs) starting from a given design, and then records the non-dominated paths (along the different change cost dimensions, typically dollar cost and time delay) to each potential end state. This results in a collapsed transition matrix, created from all of the individual mechanism transition matrices, which can be referred to as a *full accessibility matrix*, indicating all potential design connections via multiple change mechanisms and paths. This process is illustrated in Figure 5-3.

Performing subsequent analysis using the full accessibility matrix instead of the individual transition matrices will allow for the consideration of change mechanism coupling effects. Note however, that the creation of the full accessibility matrix is computationally expensive, as the number of multi-arc paths that must be considered scales with the number of potential end states raised to the power of the number of arcs allowed. In the past, computational limits restricted the feasibility of using a full accessibility matrix and the case studies performed for this research represent the first application of one in analysis.

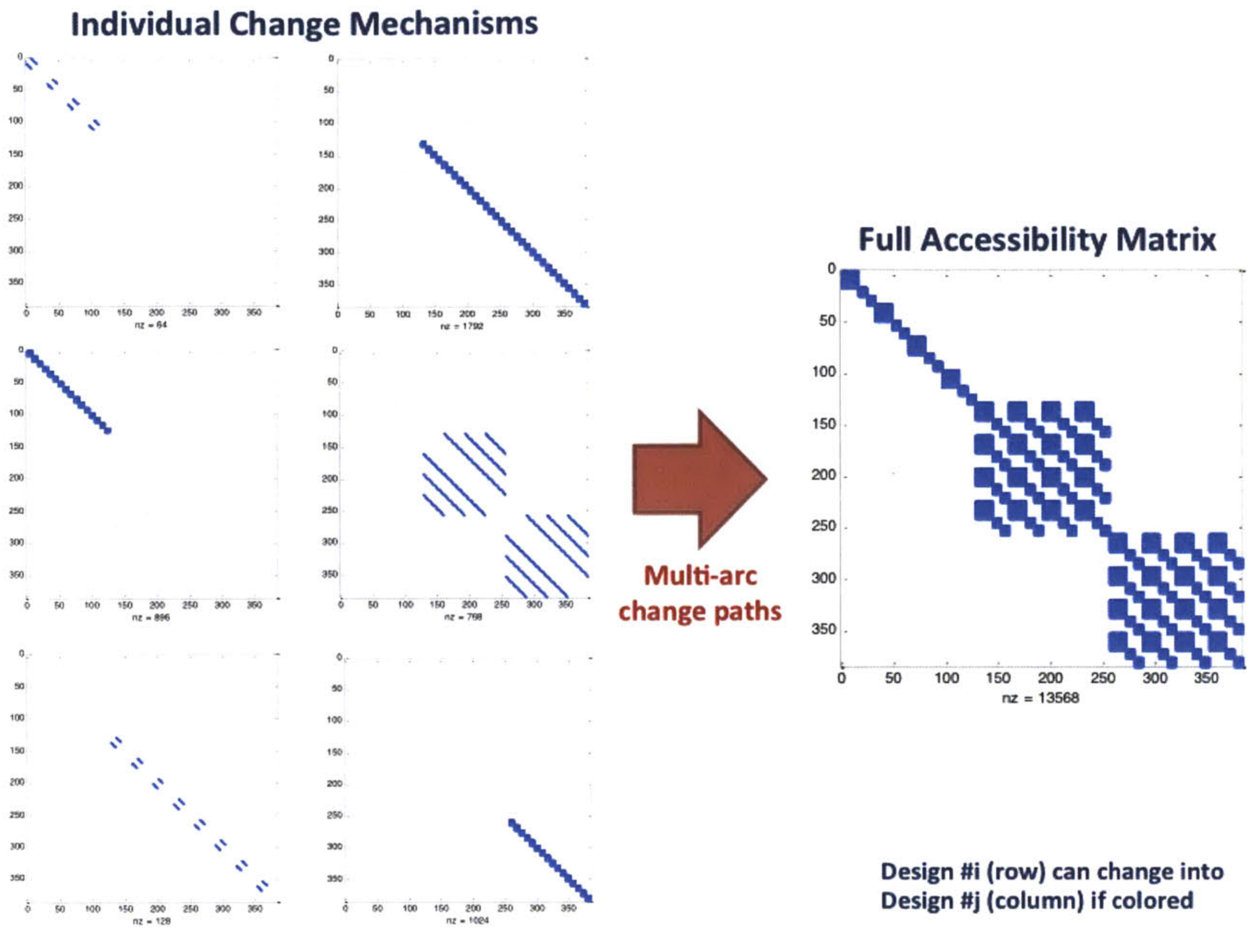


Figure 5-3: Creation of the Full Accessibility Matrix

Table 5-1: VASC Step 1 Summary - Set up Epoch-Era Analysis

Inputs	Expert opinion
	Stakeholder opinion
Activities	Identify/define: Design variables Performance attributes Stakeholder preferences (utility functions) Context variables Change mechanisms
	Create system model that evaluates performance of any design in any context
	Collapse transition matrices into full accessibility matrix (if feasible)
Outputs	List of design vectors to consider (combinations of design variable levels)
	List of epochs to consider (combinations of context variables / stakeholder preferences)
	Performance of each design in each epoch (utility, FPN)
	Transition matrices / full accessibility matrix
Goals	Prepare data for further analysis

5.2 Identifying Designs of Interest

Step 2 of VASC involves selecting a subset of the designs in the tradespace for further analysis in the remaining steps. This is necessary to reduce both the computation time and the difficulty of synthesizing and grasping the results of the approach by limiting the scope of the system designer's full attention. It is also possible that many designs in the tradespace will be obviously sub-optimal selections to build, regardless of how valuably changeable they are, and this step prevents the wasting of computational effort on these designs. A summary of Step 2 is shown in Table 5-2.

Multi-epoch screening metrics (e.g. Normalized Pareto Trace and Fuzzy Normalized Pareto Trace for value robust designs, and Filtered Outdegree for highly changeable designs) are the main technique by which system designers can quickly select potentially valuable designs. Any other desired design identification techniques (such as picking favorite designs through reuse or high performance in other metrics) are also acceptable; the only requirement here is to select a small fraction of the full tradespace. If concurrent visualization for comparison is desired, then the number of designs in this set should be on the order of 5-7 for clarity purposes.

Note that this step is particularly important when intending to utilize VASC iteratively. For example, in a tradespace that considers designs of very different types, such as land, sea, and air designs all intended to perform the same task, it is probably beneficial to perform a separate iteration of VASC for each of the types. This allows more similar designs to be compared directly within each iteration (a simpler cognitive process), and delays the comparison of fundamentally different designs until after the preferred options are identified for each type. For this example, the selection of designs of interest could be limited to only land designs for the first iteration, sea for the second iteration, and air for the third iteration, before performing a final iteration that selects the most promising designs for each type from the previous iterations for comparison.

Table 5-2: VASC Step 2 Summary - Select Designs of Interest

Inputs	EEA setup
Activities	Calculate screening metrics: NPT / fNPT FOD Any other stakeholder desired metrics
	Use screening metrics to select promising designs
Outputs	Subset of designs for further analysis
Goals	Limit scope of analysis to reduce computation time and improve clarity

5.3 Defining Changeability Execution Strategies

Defined in step 3, the concept of "changeability strategy" is the unifying factor of the approach, specifying the logic that interprets the system condition and identifies the change path that should be executed for any given epoch. The set of possible change execution strategies is chosen in this step, defining strategies in terms of logic used to select the desired change mechanism and end state for each design in each epoch. Then, for each design/epoch pair the most desirable change path according to each strategy is calculated and recorded for later analysis. Outputs of this step include the realized end states and transition costs for each combination of design/epoch/strategy. A summary of Step 3 is shown in Table 5-3.

When defining strategies, remember that any logic can be used as long as it is capable of selecting a single change path (or none) for each design in each epoch. If possible, interviewing the system stakeholders to determine how they intend to utilize changeability over the system's lifecycle and using this statement to create a matching strategy should be performed. If the stakeholder is unsure or would like analysis to be performed on a range of strategies in an attempt to find the most beneficial way to manage changeability, then a range of potential strategies can be created for comparison, usually from expert opinions. These strategies can vary from the most simple (maximize utility in all epochs) to extremely complex (maximize efficiency, but only if the cost of executing the change is less than a given threshold that varies from epoch to epoch, and in certain epochs some change mechanisms are not available). However, the more complex the strategy, the more complex the design or epoch spaces need to be in order to generate results that activate all the conditions, so simpler problems should mostly utilize simple strategies.

Special attention should be paid to a particular caveat: there is no "right" or "best" strategy to test on every project. Each strategy statement implies a different goal, and each is best at achieving its own goal. It is up to the system designers and stakeholders to interpret the results of VASC for the value of changeability or lifetime system value if a prescriptive statement about the "correct" choice of strategy is desired.

Table 5-3: VASC Step 3 Summary - Define Changeability Execution Strategies

Inputs	Designs of interest
	EEA setup
	Stakeholder/expert opinion
Activities	Determine strategies
	Define the decision logic amongst change paths for each strategy
	Generate selected change path for each design/epoch/strategy combinations
Outputs	Set of strategies to use / compare
	Executed change paths under each strategy for further analysis
Goals	Determine *how* changeability will be executed for this system across the range of uncertainty under consideration

5.4 Performing Multi-Epoch Analysis

Multi-Epoch Analysis includes the interpretation of design performance across the epoch space but without the time-ordering effects of eras. Performing multi-epoch changeability analysis, as in VASC, involves looking not only at a design's performance in each epoch but also considering the strategic transitions determined in the previous step and the performance of the targeted end state design in each epoch. The main method for interpreting this data involves calculating multi-epoch metrics, such as Effective NPT and Effective Fuzzy NPT, Fuzzy Pareto Shift, Removal Weakness, and Available Rank Increase. This generates information on when, why, and how designs of interest are changing within epochs and the value of those changes, as well as identification of particularly valuable change mechanisms and/or designs which rely on a single mechanism for a large portion of their value (Fitzgerald and Ross, 2012a). A summary of Step 4 is shown in Table 5-4.

It is important to keep in mind that the goal of performing Multi-Epoch Analysis is to understand the *space* of potential uncertainties with regards to how each design of interest will respond with its changeability under a given strategy. The strategy determines what the design *will* do under what *may* happen, and multi-epoch metrics operate on this information. The setup of EEA allows system designers to calculate this information without the need for simulation, giving powerful insights at low costs.

Table 5-4: VASC Step 4 Summary - Multi-Epoch Analysis

Inputs	Designs of interest
	Strategic change executions (paths and end states for each design in each epoch)
Activities	Calculation of multi-epoch metrics for each design/strategy eNPT/efNPT FPS ARI Removal Weakness
	Visualization of data with appropriate graphs/tables
Outputs	When/why/how designs of interest are changing
	Comparison of design changeability values (without time-ordering uncertainty)
	Comparison of strategies for enabling value in different designs
	Valuable/critical change mechanisms
Goals	Understand how each design of interest will respond when faced with the complete range of considered contexts and stakeholder preferences
	Determine the relationships between different strategies
	Computationally inexpensive insights into valuable changeability

5.5 Performing Era Simulation and Analysis

For Step 5, additional assumptions are applied in order to extract more information out of the strategic changes. An era constructor is created, which generates eras by stringing together sequences of epochs. The assumptions are related to the likelihood of switching between given epochs, and the durations of each epoch: these can vary from deterministic (lifecycle scenario planning) to fully probabilistic (associated probability distributions for each epoch's duration and the likelihood of transitioning into each other epoch when one ends). Applying these assumptions allows for the creation and analysis of sample eras. Sample eras give important lifecycle information on the designs of interest as they perform, change, and age over time, as well as help identify valuable change mechanisms. Of course, unless only deterministic eras are considered, many samples must be run for each design in order to properly cover the range of potential long-term futures and achieve statistical significance, making this step of VASC significantly more computationally expensive than Multi-Epoch Analysis. Era Analysis allows for the collection of many types of information on lifecycle performance, including change mechanism usage frequency and likelihood, statistics on average/aggregate utility provided and design efficiency, comparison of strategies and change mechanism usage for each design, and the "going rates" for tradeoffs between adding and removing changeability (Fitzgerald and Ross, 2012b). A summary of Step 5 is shown in Table 5-5.

Conceptually, the process of simulation in Era Analysis is very similar to that of simulation in Real Options Analysis studies: a set of assumptions is created in order to model the system's lifecycle under uncertainty, which can then be sampled repeatedly to generate information. However, EEA is specifically designed to allow for the consideration of multiple sources of uncertainty (epoch variables), while ROA typically follows the evolution of a single continuous uncertain variable. Additionally, the EEA setup of piecewise-static tradespaces allows for analysis without depending on monetization of value, instead using other metrics. However, revenue functions and discounted cash flow analysis can still be used if deemed appropriate. This is a good example of how EEA and VASC offer a more general alternative to ROA, focusing more on design and change mechanism exploration and allowing more assumptions to be layered on as desired rather than including them up front.

Table 5-5: VASC Step 5 Summary - Era Simulation and Analysis

Inputs	Designs of interest
	Strategic change executions (paths and end states for each design in each epoch)
Activities	Creation of era constructor
	Simulation of many sample eras for each design of interest under each strategy
Outputs	Change mechanism usage frequency / likelihood
	Statistics on lifecycle value delivery (distributions for total utility, efficiency, etc.)
	Comparison of design lifecycle performance under different change strategies
	"Going rates" for changeability tradeoffs
Goals	Understand how designs perform as they change over time
	Put value in aggregate lifecycle terms to assist decision makers

Step 5 is the final step of VASC, so this completes the summary of the approach. In order to assist in the comprehension of VASC, the following section of this thesis will demonstrate the application of VASC to a few case studies, walking through the key steps and showing the types of insights able to be extracted using the approach.

6 Application of VASC

This section will show the application of VASC to two case studies. These cases demonstrate the insight able to be drawn from the use of VASC and the ways by which data can be manipulated and visualized in order to find these insights. As a reminder, the five steps of VASC are:

1. Set up data for Epoch-Era Analysis
2. Identify designs of interest
3. Define changeability execution strategies
4. Perform Multi-Epoch Analysis
5. Perform Era Simulation and Analysis

A third case study is introduced, but not completed due to computational limitations, which are discussed. All of the computational aspects of these case studies were implemented using the MATLAB® programming environment, which is extremely useful for tradespace exploration because it is optimized for operating with vectors and matrices, which are the preferred means of holding the data for the large number of designs evaluated in a tradespace.

6.1 X-TOS

X-TOS is a proposed particle-collecting satellite designed to sample atmospheric density in low Earth orbit. In 2002, a full MATE study was performed on X-TOS in order to characterize the design space and find potential promising designs for the system (16.89 Space System Engineering, 2002). The study was comprise of 7840 designs, created from 8 design variables, with a performance model measuring 5 utility-generating attributes; the variables and related attributes are shown in Table 6-1. Evaluated but not included in the final tradespace were multi-satellite configurations, due to vastly increased cost for only marginal increased utility. Also, it should be noted that infeasible design vector combinations were *pre-eliminated* from the tradespace, reducing the number of designs under consideration to 3384. Most importantly for this work, changeability was noted to be highly desirable in the X-TOS final report, because an unknown parameter (atmospheric density, which the system was designed to measure) had a large impact on the performance of the satellite.

Table 6-1: X-TOS Design Variables and Associated Performance Attributes

Design Variable	Directly Associated Attributes
Apogee	Lifetime, Altitude
Perigee	Lifetime, Altitude
Inclination	Lifetime, Altitude, Max Latitude, Time at Equator
Antenna Gain	Latency
Comm. Architecture	Latency
Propulsion Type	Lifetime
Power Type	Lifetime
ΔV Capability	Lifetime

2006 saw X-TOS revived as a case study for a research effort to quantify changeability (Ross and Hastings 2006). To expand the case, eight potential change mechanisms were identified, and their associate transition rules are included in Table 6-2. Two binary design variables, "tugable" and "refuelable", were added as enablers for the appropriate change mechanisms, included at a fixed cost to the system. Note that there is a significant range in change execution costs, as the fuel-burning change mechanisms (#1-3) are modeled to cost orders of magnitude less to employ, in both time and money, than the others.

Table 6-2: X-TOS Change Mechanisms

#	Change Mechanism	Design Requirement	Effect
1	Plane Burn	Sufficient ΔV	Change inclination, decrease ΔV
2	Apogee Burn	Sufficient ΔV	Change apogee, decrease ΔV
3	Perigee Burn	Sufficient ΔV	Change perigee, decrease ΔV
4	Plane Tug	Tugable	Change inclination
5	Apogee Tug	Tugable	Change apogee
6	Perigee Tug	Tugable	Change perigee
7	Space Refuel	Refuelable	Increase ΔV
8	Add Sat.	(none)	Change all orbit parameters and ΔV

The application of the X-TOS case for this research had two main goals: (1) to serve as an experimentation case to develop VASC and the changeability metrics, and (2) to test the metrics, particularly Fuzzy Pareto Shift (FPS), through empirical investigation for their capability to generate useful insights on valuable changeability.

6.1.1 Set Up Epoch-Era Analysis

Recall: in this step, the necessary constructs for VASC's uncertainty model are created, particularly the epochs (using stakeholder preferences and context variables) and the change mechanisms / transition matrices.

The 2006 version of X-TOS was also created with a goal to test the sensitivity of the MATE insights to changes in the user preferences that defined the utility function (Ross, 2006). To this end, 58 different preference sets were created, by varying the base preferences in one of four ways: (1) Changing the value-delivering attribute set; (2) Changing the attribute weightings in the multi-attribute utility function; (3) Linearizing the attribute utility curves; and (4) Using different utility aggregating functions. Only one of these four perturbations was applied at any one time. A sensitivity study such as this is designed to help answer "what if" questions such as "What if we didn't elicit the 'right' requirements/preferences?" and "What if we didn't use the 'right' priorities for the attributes?" User preferences are designated as one of the most important epoch variables. These 58 preference sets are able to be used to create 58 different epochs, for a simple application of X-TOS to EEA.

6.1.2 Select Designs of Interest

Recall: in this step, screening metrics are utilized in order to reduce the number of designs considered in full detail to a manageable amount, ideally less than 10 for the comparisons to be most comprehensible.

In order to screen the designs, Normalized Pareto Trace (NPT) is considered, with the goal of identifying designs that are passively value robust: cost-utility efficient in a large number of epochs. NPT is plotted for the entire design space in Figure 6-1, with Design 31 highlighted as the design with the best NPT. Design 31 is thus a candidate for further investigation, as it might be expected to perform well based on its robustness. By considering Fuzzy NPT, designs that are nearly cost-utility efficient will also be identified: Figure 6-2 shows this plot and highlights Designs 1, 345, 689, and 2759, which are all within 1% of the Pareto front for all 58 epochs. Note that these are not the only designs with an fNPT of 1; they were selected to provide as broad a range of the design variables as possible, in order achieve as much diversity and cover as much of the design space as possible. If VASC were to be iterated a second time on other designs of interest, there are many more choices here: 42 designs have an fNPT of 1 in this case.

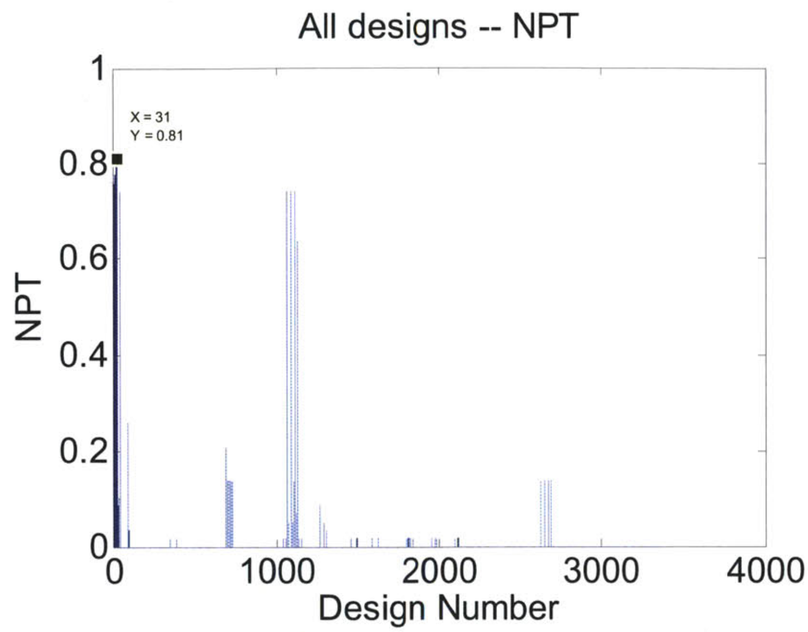


Figure 6-1: X-TOS Complete Design Space NPT

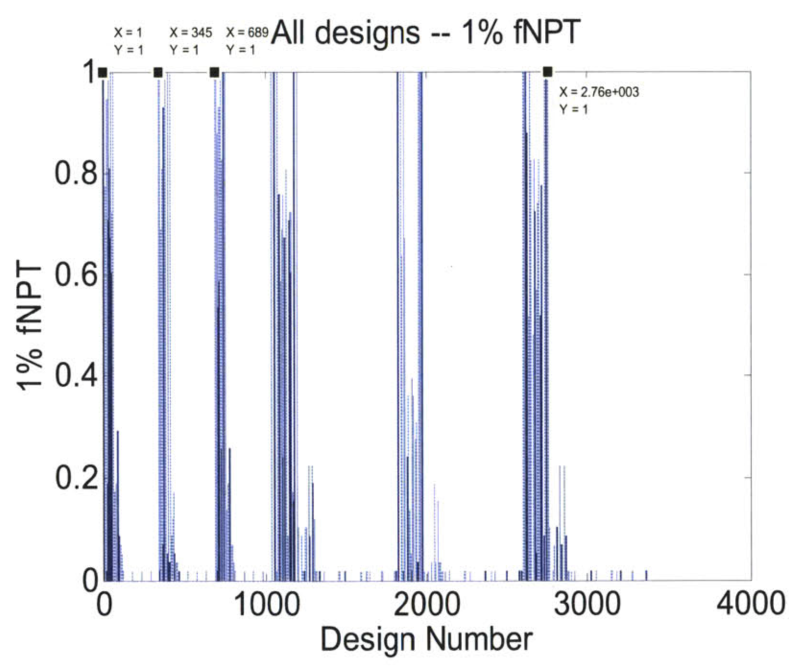


Figure 6-2: X-TOS Complete Design Space 1% fNPT

Another useful screening metric is Filtered Outdegree (FOD), which can be used to identify designs with a large number of outgoing change arcs. Heuristically, these designs are expected to derive more value from changeability because of their increased number of options; this will be tested with the rest of the VASC process. Varying the filter in the FOD equation allows designs to be found with large numbers of options available for different levels of acceptable transition cost; this can be useful because a design with a large number of cheap transition options but not the most options overall may provide a less expensive highly-changeable alternative. Figure 6-3 shows the FOD for the design space at two different thresholds: one essentially unlimited (10^{10} dollars and seconds) and the other quite limited (10^3 dollars and 10^5 seconds), identifying four more designs of interest. Again, a small number of candidate designs were selected from these plots, deciding from amongst ties by attempting to diversify the selection of design variables as much as possible.

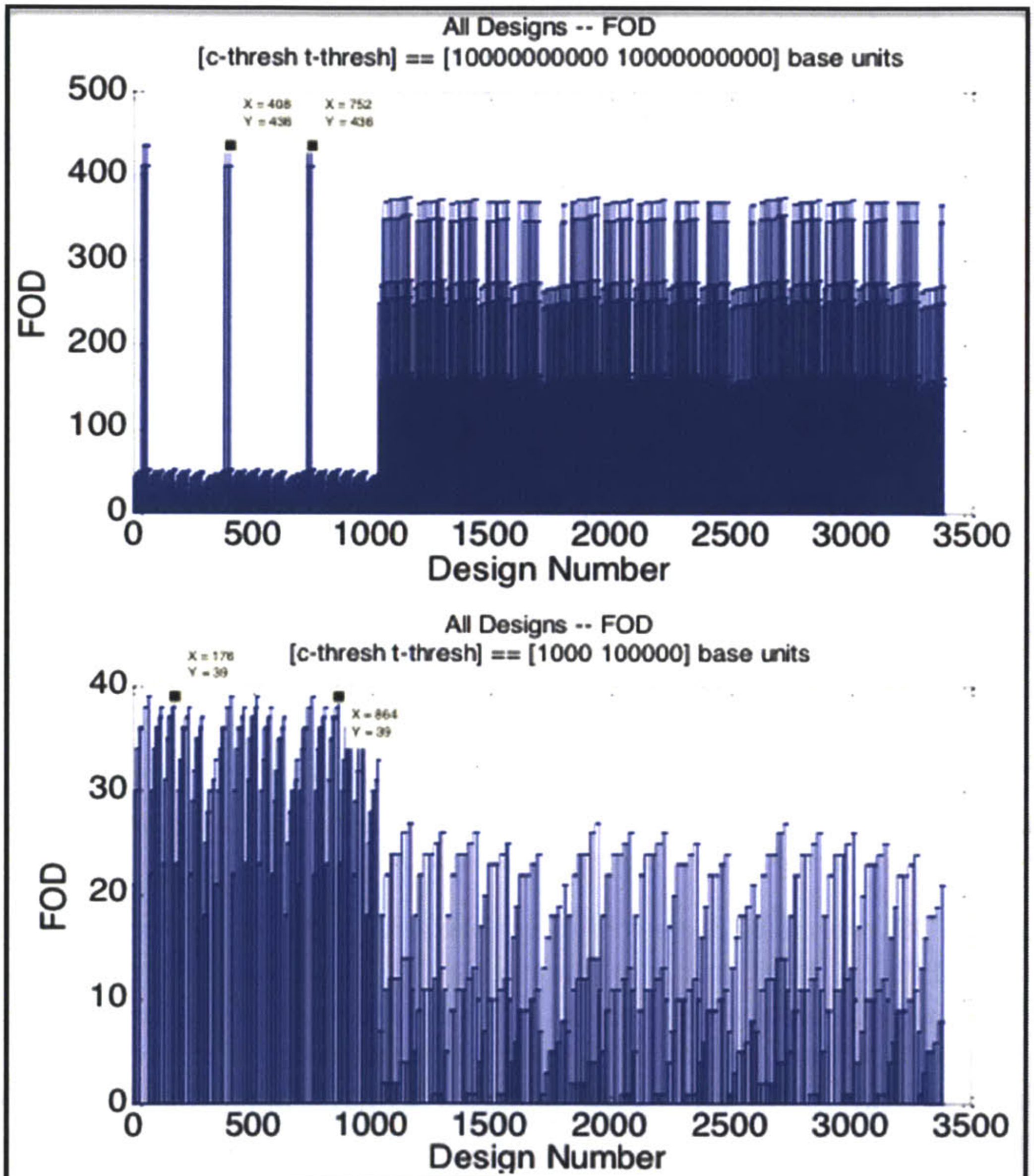


Figure 6-3: X-TOS Complete Design Space FOD for Two Filters

The nine selected designs of interest are listed in Table 6-3 with the respective values of a subset of their design variables (orbit characteristics, ΔV, and propulsion type), a reference letter to be used in future plots, and a note of which screening metric was used to identify it. Note that the screening metrics selected only designs with chemical propulsion: this is a first-order insight suggesting that chemical propulsion is dominant over the other options (in this case, electric). If the decision maker feels that electric propulsion carries some positive characteristic which is not being captured by the utility function or the screening metrics, VASC could be repeated with a different set of designs of interest, focusing on electric satellites. Also of interest is the fact that three of the designs identified here match the set of designs of interest used in the 2006 X-TOS changeability study, suggesting that the screening metrics are in line with previous expert insights.

Table 6-3: X-TOS Designs of Interest

Design #	Reference	Inclination	Apogee	Perigee	Delta V	Prop Type	Screening Metric
1	A	30	458.33	150	1200	Chem	fNPT
31	B	30	458.33	283.33	1200	Chem	NPT
176	C	30	1075	350	1200	Chem	FOD
345	D	70	458.33	150	1200	Chem	fNPT
408	E	70	458.33	350	100	Chem	FOD
689	F	90	458.33	150	1200	Chem	fNPT
752	G	90	458.33	350	100	Chem	FOD
864	H	90	1075	350	1200	Chem	FOD
2759	I	90	766.67	216.67	400	Chem	fNPT

6.1.3 Define Changeability Execution Strategies

Recall: in this step, the main task is creating the strategies used to determine which available change path will be selected by each design in each epoch.

For X-TOS, two strategies are considered: Maximize Utility and Maximize Efficiency (as measured by FPN). These are basic strategies that target simple measures of system performance at any cost, attempting to predict how a stakeholder might choose to use the system, since this is an academic project with no true stakeholder to interview for his intended changeability strategy. After picking the strategies, a MATLAB® script finds the executed transition (if any) for each design in each epoch with each strategy, which feeds into the following steps.

6.1.4 Perform Multi-Epoch Analysis

Recall: in this step, the new valuable changeability metrics are applied to the designs of interest in order to extract information about their performance across the space of uncertainty, without the computational burden of simulation.

To begin Multi-Epoch Analysis, each design's Effective Normalized Pareto Trace (eNPT), which is a changeability-acknowledging version of one of the screening metrics (NPT), can be considered in order to see how changeability is affecting their overall robustness. Table 6-4 shows the NPT of the designs of interest side by side with the eNPT resulting from each strategy. Maximizing efficiency, clearly results in an increase from NPT to eNPT because the strategy does not allow for changes that move designs away from the Pareto front. Design I, selected for its FOD and not its passive robustness, is also noticeable for having an eNPT of one under this strategy, implying that it is always capable of transitioning to a design on the Pareto front. The Maximize Utility strategy results in a decrease from NPT to eNPT for most designs; an effect like this is frequently likely, as greater utility can often be achieved only with diminishing returns on costs, thus reducing efficiency and moving designs away from the Pareto front.

Table 6-4: X-TOS - NPT and eNPT

Design	Do Nothing (NPT)	Max U	Max Eff
A	0.776	0.103	1
B	0.810	0.103	1
C	0	0	0.052
D	0.017	0.086	1
E	0	0	0.052
F	0.207	0.086	1
G	0	0	0.052
H	0	0	0.052
I	0	0	1

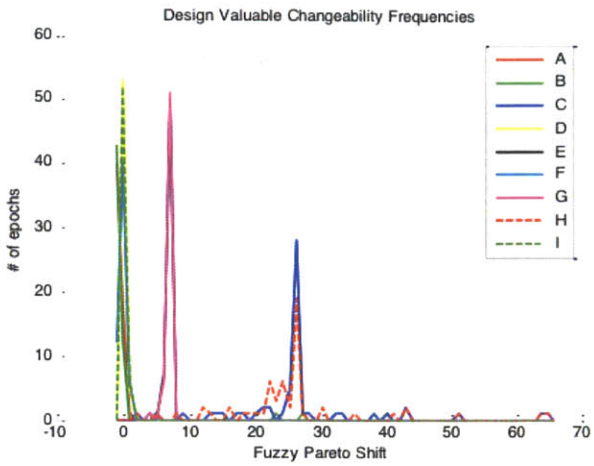
Next, allowing for a fuzzy margin on these metrics will give a more inclusive view on Pareto efficient designs. Table 6-5 shows the corresponding fNPT and Effective Fuzzy Normalized Pareto Trace (efNPT) of the designs of interest with a 1% fuzziness, and improvements over the not-fuzzy values highlighted in green. All of the designs of interest selected in Step 2 perform near-optimally (within 1%) for the majority of the epoch space when their changeability is taken into consideration, with the worst efNPT score in the table at a very high 0.879. This also should alleviate any concern over the use of the Maximize Utility strategy that was created when noting the significant drop from NPT to eNPT for the passively robust designs, as the 1% fuzziness level does not note any decrease between fNPT and efNPT, so the efficiency losses in pursuit of higher utility are minimal.

Table 6-5: X-TOS - 1% fNPT and efNPT

Design	Do Nothing (fNPT)	Max U	Max Eff
A	1	1	1
B	0.948	1	1
C	0	0.879	0.914
D	1	1	1
E	0	0.879	0.914
F	1	1	1
G	0	0.879	0.914
H	0	0.879	0.914
I	1	1	1

Moving on, the FPS distributions can be viewed to derive insight on the effect of the selected changes on design efficiency. Figure 6-4 and Table 6-6 show the FPS distributions and order statistics (minimum, median, maximum, and the 1st and 3rd quartiles) for the designs of interest under the Maximize Utility strategy. Comparing the FPS distributions of the designs of interest is often the most insightful aspect of this step of VASC, as it allows for an understanding of the similarities and differences between the valuable changeability of the designs. Designs A and B, which were high NPT selections, do not undergo many significantly value-affecting changes, as visible in the large spike near zero FPS, although B does have a +40 FPS for one epoch in which it performs poorly but is able to recover a large amount of value. Designs C and H appear to derive the most value from their changeability under this strategy; each has a spike in the low twenties range that comprises about half of the epochs in the epoch space, with no epochs causing a reduction in efficiency and a few high outliers of over sixty percent efficiency improvement. Designs E and G also perform well, with consistent improvement of around 7% efficiency in most epochs. The table provides a slightly less cluttered view of the same information, with the results highlighted by a heat map from red (bad) to green (good). Note that order statistics are presented and not mean or standard deviation: since FPS distributions are frequently skewed (as they are here), the mean is a misleading measure of central tendency.

Table 6-6: X-TOS Maximize Utility FPS Order Statistics

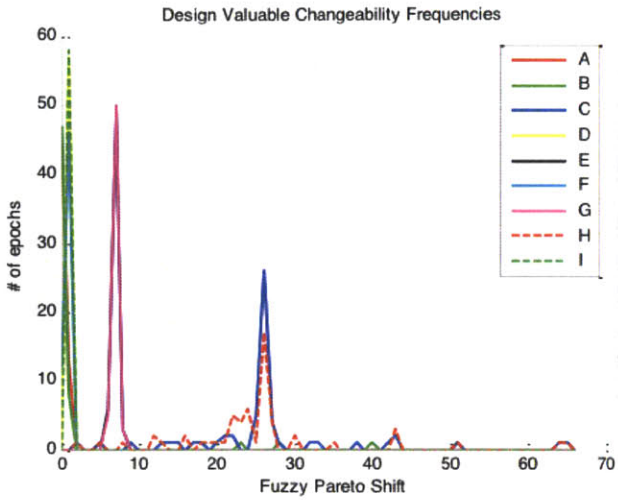


	Min	1st Q	Med	3rd Q	Max
A	-1	-1	-1	0	1
B	-1	-1	-1	0	40
C	2	25	26	26	65
D	0	0	0	0	1
E	4	7	7	7	7
F	-1	0	0	0	1
G	4	7	7	7	7
H	2	22	25.5	26	65
I	0	0	0	0	1

Figure 6-4: X-TOS Maximize Utility FPS Distribution

Figure 6-5 and Table 6-7 show the same information but for the Maximize Efficiency strategy. Interestingly, for this case study, the designs change largely similarly to the Maximize Utility strategy. The main exceptions are the removal of the few negative-efficiency changes (for Designs A, B, and F), which become no-changes with 0 FPS, because the Maximize Efficiency strategy does not allow for negative FPS changes. Also, Designs D and F, which were mostly unable to increase their utility under the previous strategy, resulting in many FPS scores of 0, instead score mostly +1 FPS under the Maximize Efficiency strategy, implying that they are slightly improving efficiency at the cost of slightly *decreased* utility. Designs C and H have nearly identical distributions to the previous strategy, suggesting that their selected changes simultaneously maximize utility *and* efficiency, which is a characteristic potentially of interest for the final design decision.

Table 6-7: X-TOS Maximize Efficiency FPS Order Statistics



	Min	1st Q	Med	3rd Q	Max
A	0	0	0	0	1
B	0	0	0	0	40
C	2	25	26	26	65
D	0	1	1	1	1
E	6	7	7	7	8
F	0	1	1	1	1
G	6	7	7	7	8
H	2	22	26	26	65
I	1	1	1	1	1

Figure 6-5: X-TOS Maximize Efficiency FPS Distribution

6.1.5 Perform Era Simulation and Analysis

Recall: in this final step, the main task is simulating potential long-run futures for the system and evaluating complete lifetime performance of the system and its changeability, including effects that may only arise when uncertainty is time-ordered.

A simple era constructor was created for X-TOS, defining eras as follows:

1. 20 randomly selected epochs (in this case, preference alterations)
2. Each epoch is 1 year in duration

This is an exceedingly simple era constructor, but it encompasses the basic features desired by this particular study. Since the epochs vary only by preference set, the goal of Era Analysis is to understand the perceived value impacts of uncertain varying preferences over time and, in particular, the use of the change mechanisms. This makes features such as varying epoch lengths or more sophisticated sampling of epochs unimportant, although they could be implemented as well.

One thousand eras were constructed and analyzed for each candidate design in the study. As the eras were being simulated, the total dollar cost, time cost, and number of changes were recorded, to be averaged at the end, along with tracking of FPN across the era and transition usage by change mechanism. That data is presented in Table 6-8, and it reveals a number of interesting outcomes.

Table 6-8: X-TOS Era Data on Design Transitions and FPN

Max Utility					Max Efficiency				
Initial Design	Avg # Trans.	Avg Total Trans Cost	Avg Total Trans Delay	Avg FPN	Initial Design	Avg # Trans.	Avg Total Trans Cost	Avg Total Trans Delay	Avg FPN
A	19.9	$386M	2.91 yrs	2.76	A	7.5	$271M	2.97 yrs	1.99
B	19.9	$382M	2.86 yrs	2.73	B	7.1	$260M	2.85 yrs	2.21
C	19.8	$396M	2.96 yrs	5.54	C	10.3	$319M	3.44 yrs	5.18
D	19.9	$422M	3.34 yrs	2.69	D	8.3	$310M	3.38 yrs	2.22
E	19.9	$432M	3.42 yrs	4.47	E	10.2	$335M	3.66 yrs	4.19
F	19.8	$420M	3.32 yrs	2.68	F	8.0	$300M	3.25 yrs	2.23
G	19.8	$425M	3.33 yrs	4.68	G	10.2	$329M	3.59 yrs	4.17
H	19.8	$419M	3.25 yrs	5.47	H	10.1	$306M	3.30yrs	5.06
I	19.9	$422M	3.34 yrs	2.64	I	8.4	$314M	3.41 yrs	2.28

First, it is clear that the Maximizing Utility requires a transition at nearly every epoch switch, as there are only 20 epochs in each era and all designs average 19.8 transitions per era or more. This makes sense considering that the epochs define different preferences and thus, assuming that the preferences change enough to differentiate themselves, each epoch will have a different best-utility design reachable by the system. It is also apparent that, despite significantly fewer transitions (less than half as many) and lower amounts of money spent on transitions (~25% savings), the Maximize Efficiency strategy has approximately the same amount of time delay from transitions (3 out of 20 years, ~15% of total era duration spent changing the design). Finally, by tracking FPN across the era, the average FPN experienced by the system can be taken as a measure of the system's lifetime cost efficiency, which varies between 2% and 5% inefficient for the different designs of interest: a relatively small variation. Thus, it appears that the designs are all quite similar in operation given the changeability strategies used here, with a slight advantage to the Maximize Efficiency strategy and passively robust, high NPT/fNPT designs (A, B) for their lower expected transition costs. Interestingly, while Design C has the worst average FPN for both strategies, Design I, which was identified for its high FOD, supplants Design A as the design with the best average efficiency under the Maximize Utility strategy, implying that Design I's utility improvements are more efficient that Design A's.

Overall though, the designs of interest all have relatively similar performance according to these metrics. Identifying *why* this is true has the potential to increase the complete understanding of the tradespace. By looking at the transitions selected by each strategy, likelihoods for using each change mechanism for a random epoch switch can be calculated. These are presented in Table 6-9. Remember that multi-arc transitions are included in VASC, and thus the "sum probability" for a design across all the change mechanisms can exceed 1, as multiple mechanisms can be used in a single epoch. The two most commonly used mechanisms are Perigee Burn and, surprisingly, Redesign. Perhaps this sort of behavior is not what was desired or expected in the system; the stakeholder may want to find a design that is functional and valuable over an era without needing to redesign. To address this, a Rule Removal study can be performed.

Table 6-9: X-TOS - Change Mechanism Usage Likelihood (Random Epoch Switch)

Max Utility

Design	Inclination Burn	Apogee Burn	Perigee Burn	Inclination Tug	Apogee Tug	Perigee Tug	Refuel	Redesign
A	0	0.02	0.93	0	0	0.05	0	0.17
B	0	0.02	0.93	0	0	0.02	0	0.17
C	0.17	0.72	0	0	0.1	0.79	0	0.17
D	0	0	1	0	0	0	0	1
E	0	0.02	0.86	0	0	0	0	1
F	0	0	1	0	0	0	0	0.84
G	0	0.02	0.86	0	0	0	0	1
H	0.79	0.19	0	0	0.02	0.17	0	0.83
I	0	0.02	0.9	0	0	0	0	1

Max Efficiency

Design	Inclination Burn	Apogee Burn	Perigee Burn	Inclination Tug	Apogee Tug	Perigee Tug	Refuel	Redesign
A	0	0	0.22	0	0	0	0	0.16
B	0	0	0.16	0	0	0.03	0	0.16
C	0.47	0.45	0	0	0.09	0.41	0	0.55
D	0	0	0.98	0	0	0	0	0.98
E	0	0.02	0.88	0	0	0	0	1
F	0	0	0.79	0	0	0	0	0.79
G	0	0.02	0.88	0	0	0	0	1
H	0.41	0.57	0	0	0.02	0.47	0	0.47
I	0	0.1	0.79	0	0	0	0	1

To perform a Rule Removal study, the strategies must be reevaluated without considering any change paths including the removed rule, which is the Redesign rule for this case. This allows the criticality of that rule to be evaluated for each design of interest, by calculating Removal Weakness: the difference in FPS for each epoch caused by the removal, which can be plotted in a distribution as in Figure 6-6. Some designs (C, F) have no change, but most have ~2% decrease in efficiency in most epochs, with worst cases approaching -12% for Maximize Utility and -6% for Maximize Efficiency.

Eras can also be re-run with the new strategic transitions to see era-level effects of the removal of redesign. These statistics are shown in Table 6-10. It is appears that, while number of transitions and average FPN are about the same, the dollar and time costs of transitions have changed dramatically. Transition time delay has decreased from years to days, as expected from removing the redesign cycle, dramatically increasing the amount of time for which the system is active; this could be very attractive to a stakeholder if this were a revenue-generating project or if utility-months was the lifetime value metric of choice rather than average FPN. However, transition costs have gone up about an order of magnitude as well (this is because the way transition costs for the redesign change mechanism were originally formulated in the 2006 study did not include the relaunch cost: more careful modeling would allow more accurate cost comparisons). And while the average FPN statistics are largely the same without the redesign change mechanism as they were before (a very small increase in average FPN is experienced by most designs, as one degree of their freedom to change has been removed), Design A is now most efficient for *both* strategies.

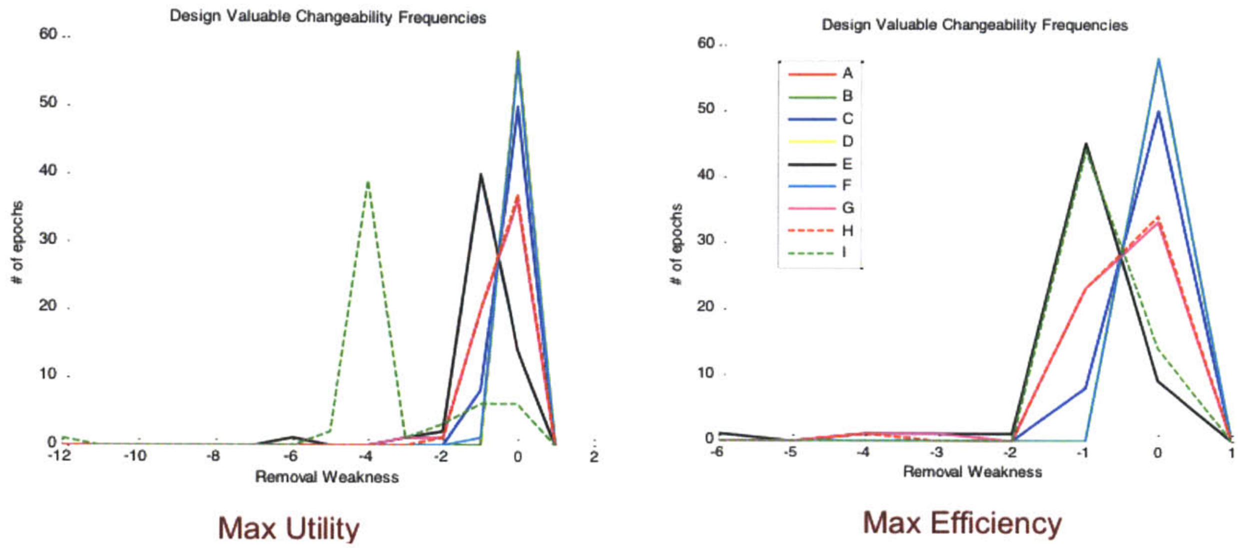


Figure 6-6: X-TOS Removal Weakness Distributions (Redesign Rule Removed)

Table 6-10: X-TOS Era Statistics (Redesign Rule Removed)

Max Utility					Max Efficiency				
Design	Avg # Trans.	Avg Total Trans Cost	Avg Total Trans Delay	Avg FPN	Design	Avg # Trans.	Avg Total Trans Cost	Avg Total Trans Delay	Avg FPN
A	19.9	$4.02B	9.5 days	2.71	A	7.4	$3.59B	8.4 days	1.79
B	19.9	$4.08B	9.5 days	2.68	B	7.1	$3.55B	8.2 days	1.96
C	19.8	$3.97B	11.0 days	5.69	C	10.4	$4.00B	12.4 days	5.56
D	19.9	$4.57B	10.8 days	2.86	D	8.2	$3.92B	9.6 days	2.06
E	19.9	$4.41B	12.2 days	4.51	E	10.8	$4.51B	13.9 days	4.57
F	19.8	$4.50B	10.3 days	2.77	F	8.	$4.18B	9.5 days	1.97
G	19.8	$4.51B	12.1 days	4.56	G	10.4	$4.32B	13.3 days	4.47
H	19.8	$4.39B	11.6 days	5.55	H	10.6	$4.25B	12.7 days	5.42
I	19.9	$4.26B	12.7 days	2.89	I	7.7	$3.49B	10.0 days	1.99

As a final synthesis, imagine that the system stakeholder has selected Design B (31) for its combination of high fNPT and high-scoring FPS in the few epochs in which it performs poorly, but is interested in potential tweaks to the design: how can VASC assist in this process? Looking back at Table 6-9, it is clear that B never utilizes a refuel and almost never utilizes the tug feature. If it is possible to isolate and remove these features, particularly the "tugable" design variable that enables all three tug mechanisms, this represents a potential means to reduce design costs of Design B. This benefit of reduced costs comes with the penalty of slightly reduced changeability-enhanced performance (by removing the few times you would choose to execute a tug) that can be quantified by another removal weakness study. Alternatively, if another potential change mechanism has been deemed feasible (for example, a variable angle sampling scoop), additional modeling could be used to estimate its cost, and it can be inserted into the study to calculate its lifetime performance benefits. Modeling efforts like this can be used to establish a "going rate" for changeability in the system: the cost/benefit tradeoff of adding or removing changeability from the selected design.

6.1.6 Discussion of VASC's Application to X-TOS

The insights derived from Multi-Epoch Analysis for X-TOS were mostly about the relationship between utility and efficiency for the degrees of freedom accessible to the designs via change mechanisms. Comparing the two strategies, it seems clear that the designs of interest are all capable of either slightly improving utility at the cost of a slight decrease in FPN, or vice-versa. The two suggested strategies do not result in significantly different FPS distributions, particularly for Designs C and E, which have the ability to change to maximize utility and efficiency at the same time, providing large value improvements.

The Era Analysis step provided some insight into the expectations for average lifetime efficiency and change costs for the designs of interest. In particular, a heavy reliance on the redesign change mechanism was identified and a Rule Removal was used to uncover the expected performance of the designs without this mechanism, which led to a small decrement in average efficiency but a large decrease in the delay time spent executing changes. Unfortunately, the lifetime statistics were very similar for all of the designs of interest, limiting the ability to appreciably distinguish their overall performance.

Overall, the results of the case study should be considered in regards to the original goals for using X-TOS: (1) to serve as an experimentation case to develop VASC and the changeability metrics, and (2) to test the metrics, particularly FPS, through empirical investigation for their capability to generate useful insights on valuable changeability. Success was achieved in both of these goals, as VASC is clearly applicable and functional on tradespace exploration cases like X-TOS, and the FPS distributions were shown to successfully distinguish between designs that generate little value from changeability and those that create large value improvements through the strategic usage of the change mechanisms.

However, the lack of separation between designs in lifetime value during the Era Analysis step limited the amount of insights and substantive distinctions between designs that were able to be made. This is a side-effect of the way in which the change mechanisms and epochs were generated for the X-TOS study. First, each design had access to the same set of change mechanisms, making valuable changeability distinguished only by the location in the tradespace of each design. If designs with *different change mechanisms* were compared, their changeability usage and value would vary much more significantly. Fortunately, modeling and comparing more differentiated designs is likely something that most real design projects will want to do, as different architectures designed for passive robustness versus changeability will likely feature very different change mechanisms (or none at all for the robust designs). Second, the fact that the only differentiation between X-TOS epochs is caused by small preference perturbations leads to a somewhat uniform epoch space. The perturbations cause design value to vary only slightly, leading to small tradespace reorderings and the plethora of +/- 1 FPS changes noted during Multi-Epoch Analysis. Also, designs that are "good" in one epoch are also "good" in essentially every epoch, leading to little separation in the lifetime performance of the designs of interest (no risk-reward type designs that are good in only some contexts). A more *differentiated epoch space*, modeling large shifts in preferences (i.e. for different "missions") or other impactful context variables, will likely lead to significantly more insights when comparing the designs of interest.

6.2 Space Tug

A space tug is a vehicle designed to rendezvous and dock with a space object; make an assessment of its current position, orientation, and operational status; and, then, either stabilize the object in its current orbit or move the object to a new location with subsequent release. A previous MATE study explored the tradespace for a general-purpose servicing vehicle of this type (McManus and Schuman, 2003). Three attributes formed the multi-attribute utility function: total ΔV capability, mass capability of the grappling system, and response time (modeled only in the binary: slow or fast). To provide these attributes, three design variables were considered in subsequent modeling activities: manipulator mass, propulsion type, and fuel load. A full-factorial design space, featuring 128 designs, was sampled and analyzed by inputting each possible combination of design variables from a set of enumerated values over a range into (1) a parametric cost estimation model and (2) a physics-based performance model. The primary purpose of the application of the Space Tug case in this research was to demonstrate the end-to-end process of VASC in a relatively simple case, but one with more significant design and epoch differentiation that X-TOS.

6.2.1 Set Up Epoch-Era Analysis

Recall: in this step, the necessary constructs for VASC's uncertainty model are created, particularly the epochs (using stakeholder preferences and context variables) and the change mechanisms / transition matrices.

In order to apply the Space Tug dataset for this analysis, the original three design variables were expanded to four design variables, which, when enumerated, resulted in 384 designs. The design variables were:

- Propulsion type (bipropellant, cryogenic, electric, or nuclear)
- Fuel mass
- Capability level of grappler
- Design for changeability (DFC) level

DFC level is implemented as a percentage mass penalty on the satellite and serves as an enabler of improved or additional change mechanisms. Think of DFC as the inclusion of extra design features or margin, at an additional cost, for example. DFC is implemented in three levels (0,1,2) and the associated change mechanisms are listed in Table 6-11. As described, all designs with either a bipropellant or cryogenic engine can switch between the two options and all designs are capable of changing fuel tank size, and both of these options reduce in cost with any investment in DFC. Also gained from investment in levels 1 or 2 of DFC is the ability to switch grappling capability; increasing capability improves utility, but decreasing capability reduces mass and thus cost. Finally, the DFC level 2 designs can also refuel in orbit, which extends lifetime while sparing the costs of redesigning and relaunching the satellite.

Table 6-11: Space Tug Change Mechanisms

No.	Change Mechanism	Effect	DFC level
1	Engine Swap	Biprop/Cryo swap	0
2	Fuel Tank Swap	Change fuel mass	0
3	Engine Swap (reduced cost)	Biprop/Cryo swap	1 or 2
4	Fuel Tank Swap (reduced cost)	Change fuel mass	1 or 2
5	Change Capability	Change Capability	1 or 2
6	Refuel in Orbit	Change fuel mass (no redesign)	2

These change mechanisms specify which *other* designs are accessible via a change for each design. The combination of multiple change mechanisms can lead to even further designs, and considering these multi-arc transitions leads to a full accessibility matrix, which indicates *all* available end states via any combination of change mechanisms for each design. The Space Tug full accessibility matrix is shown in Figure 6-7. The plot is read by locating a design number on the vertical axis and reading across to find all available other designs on the horizontal axis as indicated by a mark in the appropriate column. This plot gives a fast understanding of how connected the tradespace is, and will also qualitatively allow for an assessment of designs with many change options and thus the potential for high valuable changeability, particularly in the counting value.

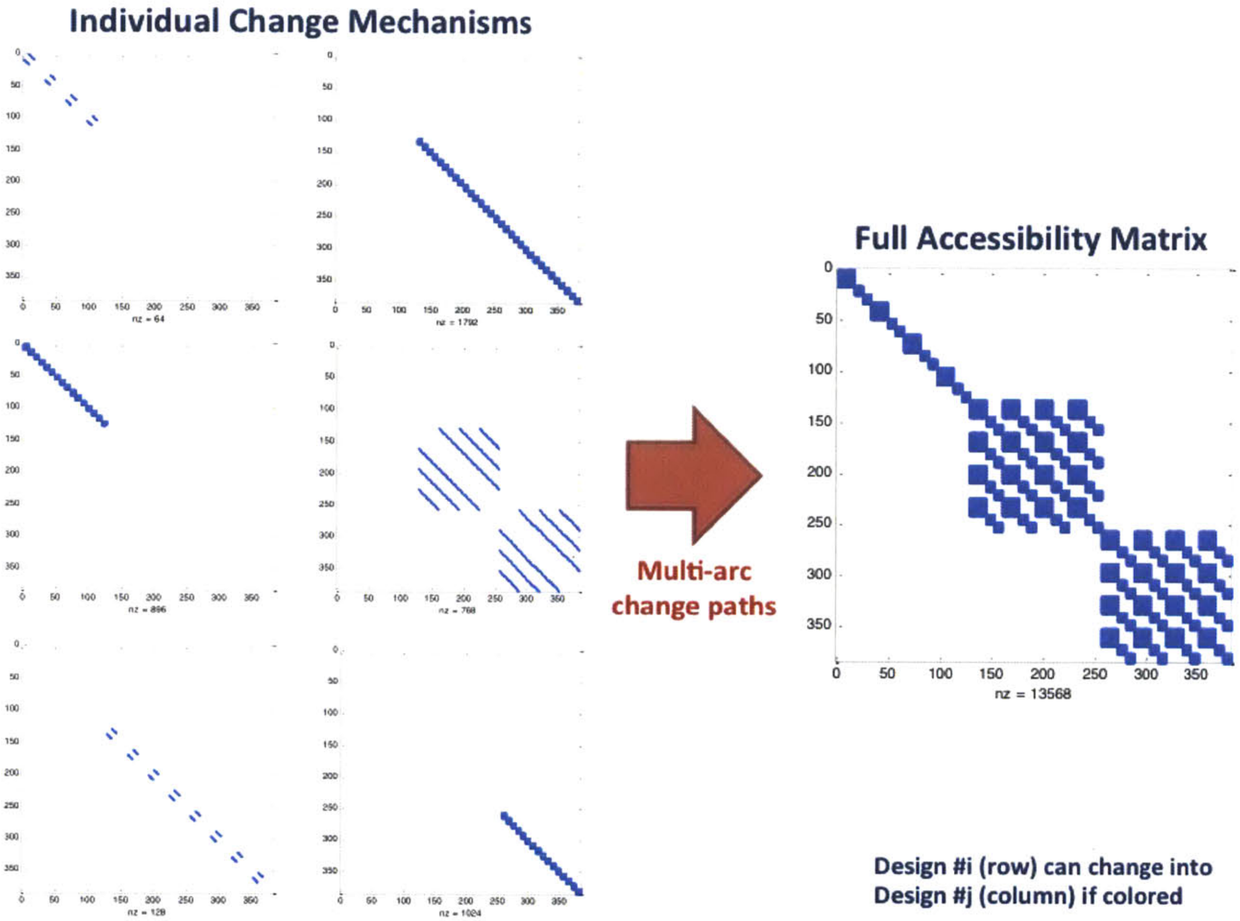


Figure 6-7: Space Tug Full Accessibility Matrix

To put Space Tug into Epoch-Era Analysis form, 16 epochs were created, generated from 2 contexts and 8 user preference sets. The two contexts corresponded to present or future technology level, which affects the transition costs of the change mechanisms and the fuel efficiencies and mass fractions for the different propulsion systems. The eight preference sets were created to correspond to different potential users of the Space Tug, with different "missions" to perform. The eight possible missions associated with "contracts" (see below) include: (1) baseline mission, (2) technology demonstration, (3) GEO satellite rescue, (4) satellite deployment assistance, (5) in-orbit refueling and maintenance, (6) garbage collection, (7) all-purpose military mission, and (8) satellite saboteur. These missions conceptually have significantly different requirements; thus the different preferences are associated with utility functions that vary dramatically in weights on the attributes and in the attribute-utility curves. The "storyline" of Space Tug is that each epoch in its lifecycle corresponds to the hiring of the tug by one of these users for a fixed-length contract.

6.2.2 Select Designs of Interest

Recall: in this step, screening metrics are utilized in order to reduce the number of designs considered in full detail to a manageable amount, ideally less than 10 for the comparisons to be most comprehensible.

After setting up the data for Epoch-Era Analysis, the next step is to identify designs of interest. For purposes of VASC, "interesting" designs are those that have a high likelihood of being valuable over a period of time, such as the intended lifecycle for a system. Two categories of potentially interesting designs include those that are "passively value robust" and those that are highly changeable. The former designs perform well across a number of epochs without needing to change. The latter designs have a large "degree" of change, but it is unknown if the accessible end states are of any value. Again, NPT/fNPT and FOD will be used as screening metrics in order to find and select these robust and changeable designs of interest, respectively.

The following figures show the design selection process for Space Tug. Figure 6-8 shows the NPT scores for the Space Tug design space, identifying designs that are passively Pareto efficient often. The designs with red stars are the ones selected for further analysis in VASC; the gray star outlines are designs that are similar in score to the other selected designs, but are omitted for being too similar in design (to promote diversity in the analysis) and to keep down the number of designs that are passed to the next steps of VASC. The remaining plots omit the gray stars for clarity. Figure 6-9 shows the 1% and 15% fNPT, which can be used to identify *nearly* passively Pareto efficient designs. Finally, Figure 6-10 shows the FOD scores for the design space with two different filters.

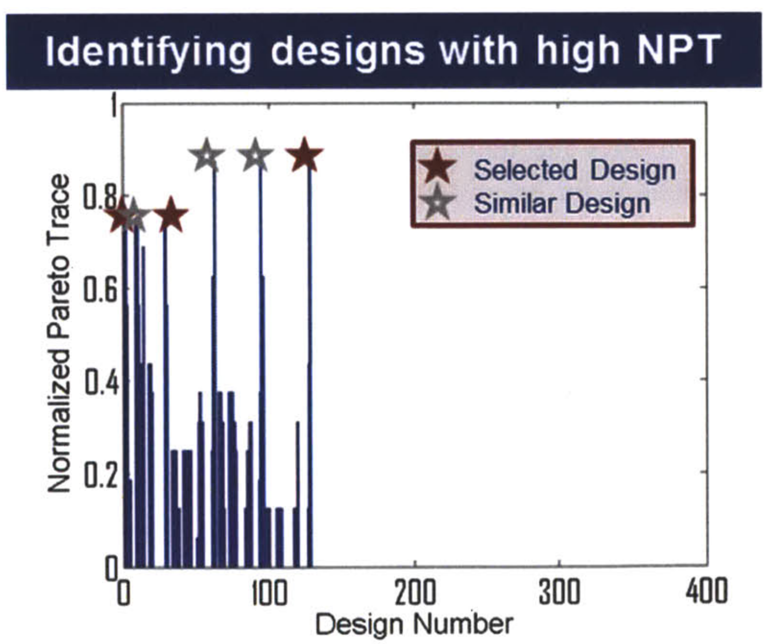


Figure 6-8: Space Tug Design Space NPT

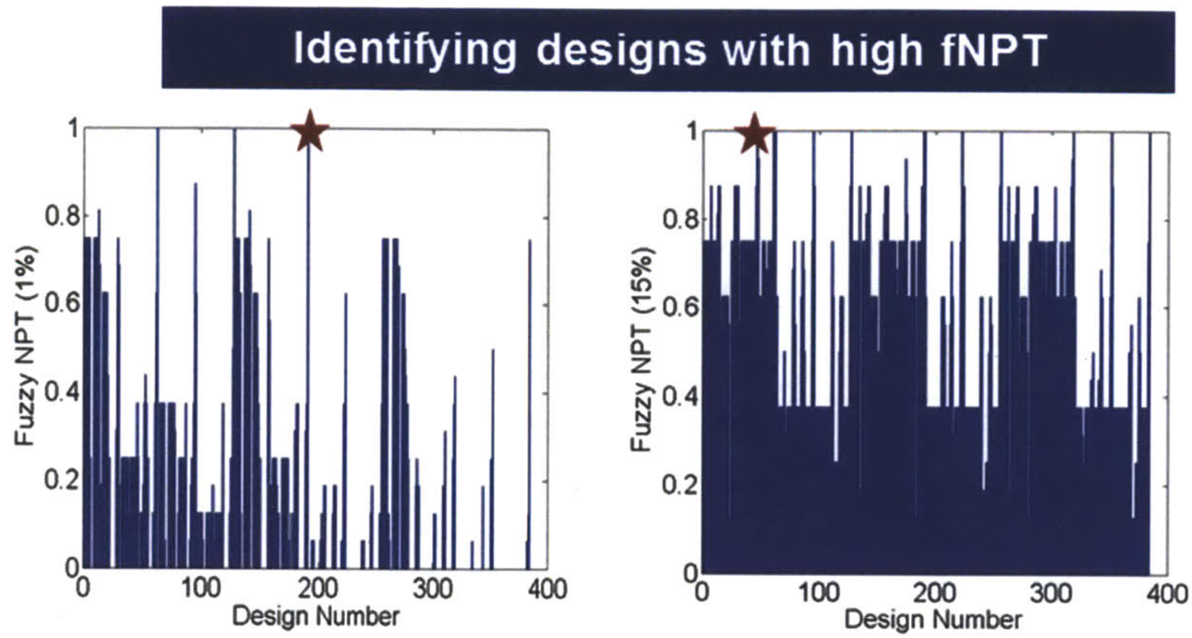


Figure 6-9: Space Tug Design Space fNPT

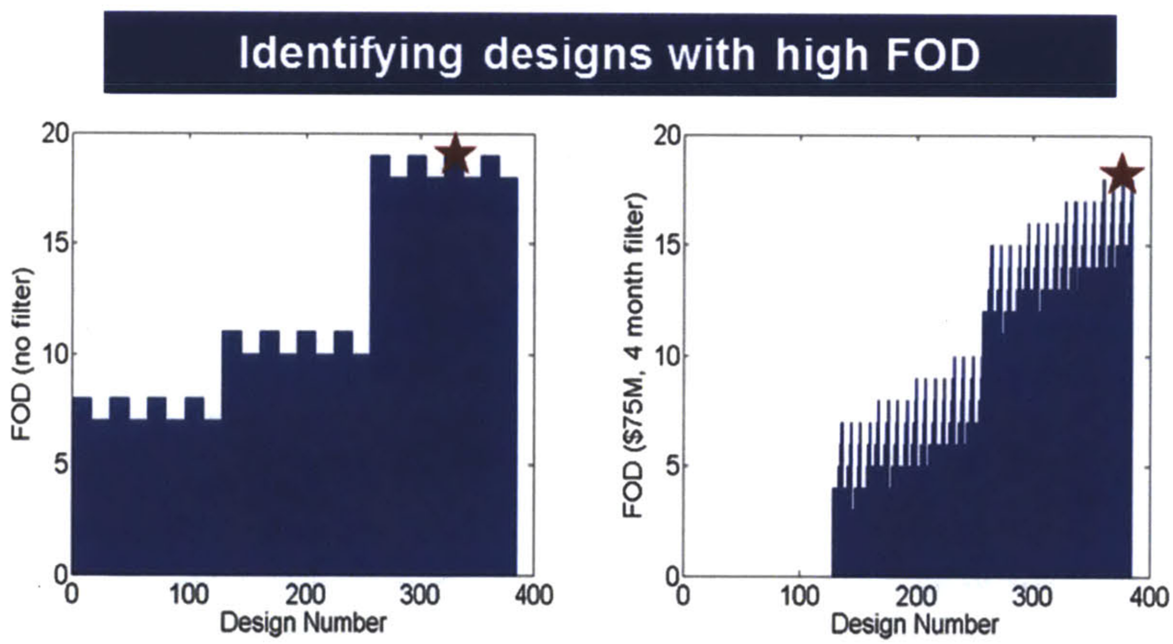


Figure 6-10: Space Tug Design Space FOD

After collecting these 8 designs, further analysis in VASC can proceed. The reference letters and design vectors for each design of interest, along with their associated costs and values in the performance attributes (for the present context), are shown in Table 6-12.

Table 6-12: Space Tug Designs of Interest

Design No.	Ref	Prop Type	DFC Level	Fuel Mass (kg)	Capability (kg)	Speed	Delta V (m/s)	Base Cost ($M)
1	A	Biprop	0	30	300	Fast	143	97
29	B	Nuke	0	1200	300	Fast	7381	306
47	C	Cryo	0	10000	1000	Fast	6147	628
128	D	Nuke	0	30000	5000	Fast	14949	3020
191	E	Nuke	1	10000	1000	Fast	16150	980
328	F	Biprop	2	50000	3000	Fast	4828	2804
376	G	Elec	2	30000	5000	Slow	27829	3952

6.2.3 Define Changeability Execution Strategies

Recall: in this step, the main task is creating the strategies used to determine which available change path will be selected by each design in each epoch.

As with X-TOS, this is an academic project and thus there is no true stakeholder to decide how changeability will be used. Thus, a few *possible* strategies are again considered for the use of changeability. For Space Tug, four strategies were tested:

1. **Maximize Utility** – Change to the highest reachable utility in every epoch, in order to make the system as good at its job as possible.
2. **Maximize Efficiency** – Change to the reachable design with the lowest FPN (closest to the Pareto front) in every epoch, to make the system as cost-utility efficient as possible.
3. **Survive** – Execute a change only if the system is "invalid" in the given epoch, or risks becoming invalid in the following epoch due to a low amount of fuel, by executing the least expensive change to a valid design. A design is considered invalid when it does not meet the minimum acceptable performance in one or more of the utility-generating attributes, and thus does not have a utility score and is not in the active tradespace for that epoch. An invalid design, which is technically not in the tradespace, is assigned an FPN of 101 for this case, which is worse than the worst possible FPN of a design actually in the tradespace (100).
4. **Maximize Profit** – Execute a change to maximize the profit earned during the current epoch. To use this strategy requires a revenue function, much like the practice for applying Real Options Analysis to non-revenue-generating systems:

$$R = \$200\text{M} + \$1000\text{M} * \text{Utility} * \text{MonthsServed}$$

Here, R is the revenue earned by the Space Tug from the user hiring it for any single epoch. There is a fixed amount that is supplemented by additional revenue that is proportional to both the utility of the system for the given user preferences and the number of months that the contract (epoch) is stated to last. Time delay for executing a change path is not counted in the number of months served for the epoch. Note that this is a somewhat peculiar strategy, because it relies on era-level information, namely the duration of the current epoch (duration is not specified in the multi-epoch domain). Thus, *the Maximize Profit strategy is not used in Multi-Epoch Analysis*. Also, in the absence of pre-computing the selected change paths for each design in each epoch with each possible *duration* of each epoch, which would take a large amount of memory to store, the era simulation under the Maximize Profit strategy will have to include a logical loop when each new contract (epoch and associated duration) arises, which slows down the simulation slightly. Regardless, this is a very interesting strategy to look at despite its higher computational requirements, as it utilizes a revenue function and attempts to capture a higher-order decision that a stakeholder might make.

6.2.4 Perform Multi-Epoch Analysis

Recall: in this step, the new valuable changeability metrics are applied to the designs of interest in order to extract information about their performance across the space of uncertainty, without the computational burden of simulation.

As the multi-epoch metrics most related to the screening metrics, eNPT and efNPT are used first in order to scan the designs for their *changeability-enabled robustness*, the ability to remain valuable over variable contexts considering all planned design changes, which considers both passive robustness and changeability simultaneously. Table 6-13 displays the results of these metrics for the designs of interest with all of the strategies, along with the regular NPT and fNPT. A few results are immediately apparent. For one, considering changeability does not always increase Pareto Trace; as mentioned previously, some strategies, such as Maximize Utility, will frequently sacrifice cost efficiency in the name of another goal (here, increasing utility). On the other hand, the maximize efficiency strategy does always score at least as well as the "do nothing" NPT, because it will never sacrifice proximity to the Pareto front during a change. It is also apparent that the level 1 and 2 DFC designs (E, F, G) do not improve from NPT to eNPT, because they have a fixed cost increase associated with changeability that distances them from the Pareto front regardless of where they transition to. However, when allowing for a 5% fuzzy margin, these designs see a distinct improvement between fNPT and efNPT, particularly design E which has a perfect score of one in every strategy. This is capturing information as desired: the highly changeable designs, while never strictly (0% fuzzy) efficient, can leverage their changeability to significantly improve value robustness across the epoch space. Looking at this metric, it appears that designs D and E are the most desirable options, with excellent scores across all strategies in efNPT. Figure 6-11 plots this data in bar chart form, which can provide a useful visual for seeing what designs perform well across which strategies and also what strategies lend themselves to more efficient administration of the system with which designs.

Table 6-13: Space Tug NPT/eNPT and 5% fNPT/efNPT for All Strategies

Designs	eNPT				efNPT (5% fuzziness)			
	Do Nothing (NPT)	*Max U*	*Max Eff*	*Survive*	*Do Nothing (fNPT)*	*Max U*	*Max Eff*	*Survive*
A	0.75	0	0.875	0	0.75	0	0.875	0
B	0.75	0	0.813	0.75	0.875	0	0.875	0.875
C	0	0	0.25	0	0.625	0.125	0.688	0.675
D	0.875	1	1	0.875	1	1	1	1
E	0	0	0	0	1	1	1	1
F	0	0	0	0	0	0.313	0.875	0
G	0	0	0	0	0	0	0.75	0

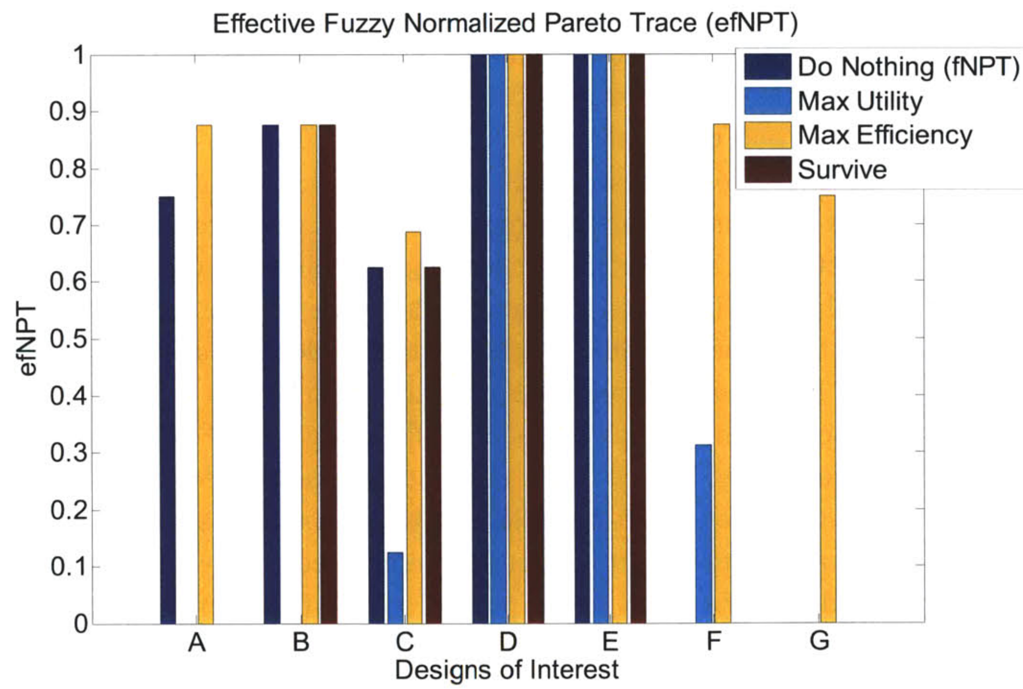


Figure 6-11: Space Tug efNPT Bar Chart

Using eNPT and efNPT has provided an understanding of the total performance, passive robustness and changeability influenced, of the designs across the epoch space. These statistics are followed up with an investigation of FPS, which isolates the value added by changeability. Remember, FPS is calculated for each design in each epoch, where the magnitude value of changeability is found in the score for each epoch, and the counting value is found in aggregate across the epochs.

Like eNPT, FPS is calculated for each strategy separately. The preferred way to view FPS data is with a distribution of the epoch scores, accompanied by a table of the order statistics of the distributions for each design. Order statistics are preferred to mean and standard deviation, because the distributions are frequently irregular and median is a vastly superior indicator of central tendency for them. The following figures and tables show the FPS distributions and order statistics for the three strategies. If a design is invalid and cannot transition to a valid design, its FPS is considered to be -101 for that epoch by convention, as this is lower than possible for any "within tradespace" negative efficiency change. Similarly, if an invalid design becomes valid, its "initial FPN" (which is actually undefined, as it is not in the tradespace) for the purposes of calculating FPS is treated as 101. The -101 FPS points are not included in the distribution plots to keep the scale reasonable.

Table 6-14: Space Tug Maximize Utility FPS Order Statistics

Design	Min	1st Q	Med	3rd Q	Max
A	-101	-19	-13	-8	93
B	-101	-25.5	-13.5	-6	-2
C	-10	-9	-6.5	-1	2
D	0	0	0	0	1
E	-3	0	0	0	0
F	-4	6	9	28	43
G	-101	-50.5	0	0	0

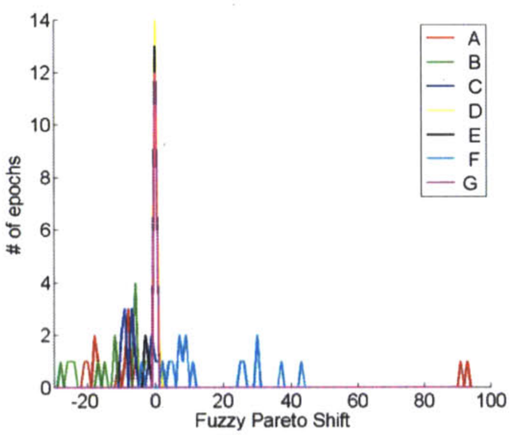


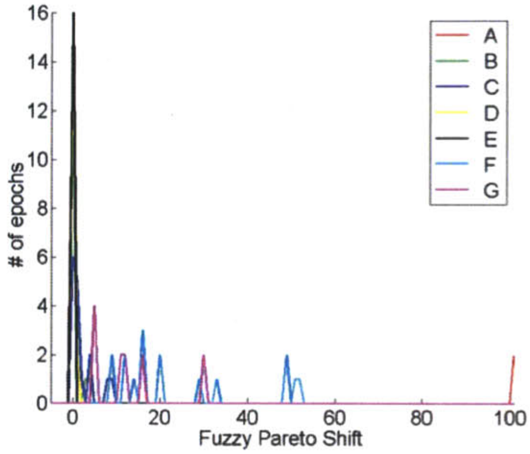
Figure 6-12: Space Tug Maximize Utility FPS Distribution

The Maximize Utility data presents us with a number of interesting insights:

1. Designs C, D, E, and F are never invalid when changeability is considered (no -101 FPS scores); this will be true for every strategy, and is possibly extremely desirable for very risk-averse stakeholders.
2. The large amount of weight in these distributions to the left of the zero point suggests that the Maximize Utility strategy does in fact result in a generally negative effect on efficiency, similar to X-TOS but this time with a slightly higher magnitude, implying that the tradeoff for higher utility comes at a more significant efficiency cost in this application.
3. Designs D, E, and G do not execute changes in a majority of epochs under this strategy, implying that they are unable to effectively improve their utility in any epoch.

4. Overall, it appears that Designs A and F have the most efficient changes in this strategy, with Design A featuring a small number of very highly valuable epochs and Design F featuring a more moderate improvement but over a larger number.

Table 6-15: Space Tug Maximize Efficiency FPS Order Statistics



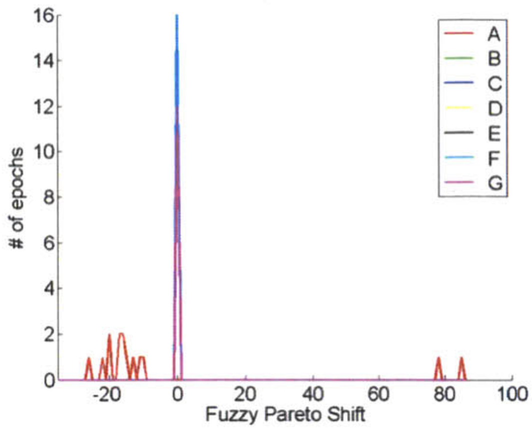
Design	Min	1st Q	Med	3rd Q	Max
A	-101	0	0	0	101
B	-101	0	0	0	4
C	0	0	1	3	9
D	0	0	0	0	1
E	0	0	0	0	0
F	9	13	18	41	52
G	-101	-48	8	14	30

Figure 6-13: Space Tug Maximize Efficiency FPS Distribution

The Maximize Efficiency strategy, as always, does not allow for negative FPS changes, except for those epochs for which there are no viable changes (-101 FPS). Note that the -48 1st quartile score for Design G is simply a mathematical artifact, as it scores -101 in exactly one-quarter of the epochs and it is then averaged up with a +5 score. As for useful insights:

1. There is significantly more weight at 0 in these distributions than there was for the Maximize Utility distributions. This implies that all of those slightly-negative efficiency changes have been replaced by not changing at all. This is partially due to preselecting the designs of interest with naturally efficient designs; a random design in the tradespace would likely still change in most epochs under this strategy.
2. Design F appears to be about the same, but now the other DFC level 2 design, G, is also displaying moderately high FPS scores, implying that it is better at improving its efficiency than it is at improving its utility.

Table 6-16: Space Tug Survive FPS Order Statistics



Design	Min	1st Q	Med	3rd Q	Max
A	-101	-21	-16.5	-12	85
B	-101	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	0
F	0	0	0	0	0
G	-101	-50.5	0	0	0

Figure 6-14: Space Tug Survive FPS Distribution

Unsurprisingly, the Survive strategy results in very few changes, as the strategy explicitly avoids changing unless in danger of failing. Of the designs of interest, only Design A is forced to consider this, as it is a minimum-fuel design and thus will always require some change in order to remain valid.

As a last step in Multi-Epoch Analysis, the ARI scores for each of the change mechanisms across the design space can be considered. Remember, ARI is used to compare change mechanisms as performance-deliverers by calculating the number of designs able to be passed in utility for a given design with a given change mechanism. The ARI in Epoch 1 (normalized by the number of designs in the tradespace) is shown in Figure 6-15. In this case, the ARI plots look nearly identical for each epoch, so just this one will suffice for insights, but in other cases the ARI scores can be compiled across epochs to get a sense for the average usefulness of a mechanism across many contexts.

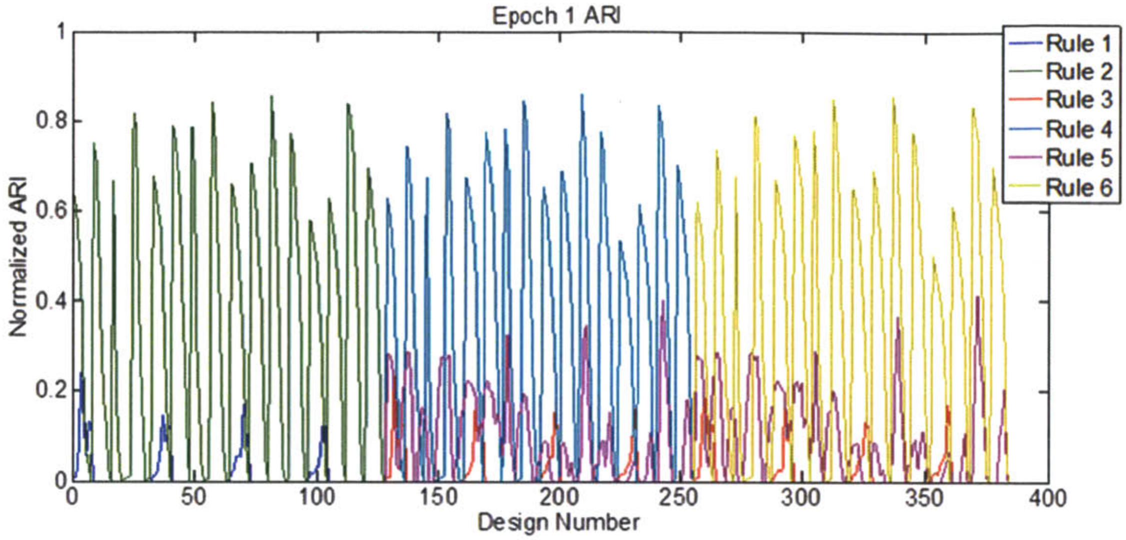


Figure 6-15: Space Tug Epoch 1 ARI Plot

The main insight able to be drawn from this figure is that it appears that mechanisms ("rules") 2, 4, and 6 are the dominant value providers, over twice as high in ARI as any other mechanisms. These are the fuel-increasing mechanisms, and are driving value by increasing the amount of ΔV available to the Space Tug. They divide the tradespace in thirds, because a different one is associated with each DFC level. In terms of the usefulness of this insight, it definitely suggests that the fuel-related change mechanisms are the most important, and thus the other mechanisms may be less critical and potential candidates for elimination or exclusion from the system in order to save time or money in system development. A Rule Removal study could be performed to analyze the effects of such a decision, but that is omitted here.

6.2.5 Perform Era Simulation and Analysis

Recall: in this final step, the main task is simulating potential long-run futures for the system and evaluating complete lifetime performance of the system and its changeability, including effects that may only arise when uncertainty is time-ordered.

For Space Tug, eras were constructed according to the following rules:

1. Epochs are chosen with a random user (mission type)
2. Epochs have a duration selected via a discrete uniform random distribution from 1 to 12 months
3. The technology context variable starts at 'present' and transitions to 'future' at a random point after 5 years
4. The total era length is 10 years

This is slightly more complicated than the X-TOS era construction, but this level of complexity is required in order to see effects of the more complicated epoch space and also in order to use the Maximize Profit strategy. For this study, 5000 eras were simulated for each design, which was determined to be enough samples to stabilize the profit data to two significant figures. The average revenues, costs and profits of the designs of interest are shown in Table 6-17, with the best and worst performing designs highlighted in green and red, respectively, for each of the four strategies.

Table 6-17: Space Tug Era Profit Data (all numbers are x10⁴ $M)

	MAX UTILITY			MAX EFFICIENCY		
Design	Avg Rev	Avg Cost	Avg Profit	Avg Rev	Avg Cost	Avg Profit
A	3.3	1.7	1.6	2.4	0.1	2.3
B	4.0	2.6	1.4	4.4	0.4	4.0
C	4.3	2.3	2	4.4	0.6	3.8
D	6.9	4.6	2.3	7.9	3.6	4.3
E	6.6	5.7	0.9	6.7	3.7	3.0
F	5.7	2.7	3	3.0	0.8	2.2
G	6.5	0.4	6.1	2.2	0.9	1.3
	SURVIVE			MAX PROFIT		
Design	Avg Rev	Avg Cost	Avg Profit	Avg Rev	Avg Cost	Avg Profit
A	3.6	0.6	3.0	3.0	0.2	2.8
B	4.9	0.6	4.3	4.3	0.2	4.1
C	5.3	0.7	4.6	4.7	0.3	4.4
D	8.6	1.6	7.0	7.7	0.7	7.0
E	6.9	1.0	5.9	6.5	0.6	5.9
F	7.1	0.3	6.8	7.5	0.3	7.2
G	6.7	0.4	6.3	7.4	0.4	7.0

This data is very interesting on a number of levels:

1. Three different designs (D, F, G) have the highest average profits under the four strategies. This confirms that *choice of strategy has a dramatic effect on value* for each design. Different designs perform better overall under different changeability strategies, again suggesting that there is no universally "best" strategy choice.
2. Despite the differences in best profit designs, Design D has the highest average revenues under all four strategies. Looking at the revenue function, it can then be deduced that Design D has the best combination of high utility and low change-execution delay.
3. The DFC level 0 designs (A, B, C, D) dominate the Maximize Efficiency strategy, ranking 5, 2, 3, and 1 respectively in average profit. This confirms that they do not require many changes to become or remain efficient, as they have very low costs.
4. Surprisingly, the Maximize Profit strategy, which maximizes *short-term* profit only, does <u>not</u> result in the highest average profits of all the strategies for every design. In fact, only the DFC level 2 designs (F, G) see the highest profits from that strategy, while the others benefit most from the Survive strategy. It appears then that actively seeking maximum profits at all times will lead the less changeable designs into some poor long-term decisions, which should definitely be acknowledged by the system stakeholder.

5. Across all of the strategies, Designs D and F have the best average performance using the profit data. If the system stakeholder anticipated using multiple strategies, these designs are probably the best choice.

All of those insights are very powerful, but they rely on a revenue function. For the purposes of this research, it is important to remember that a revenue function will not always be available, as it represents an additional level of assumption about the system. Thus, it is of interest to continue to use the other metrics in order to generate as much insight as possible. Identifying the most used change mechanisms is potentially of interest, as the most frequently used ones are candidates for improvement (via lowered execution costs or redundancy) and the least used ones are candidates for exclusion (and the associated cost savings).

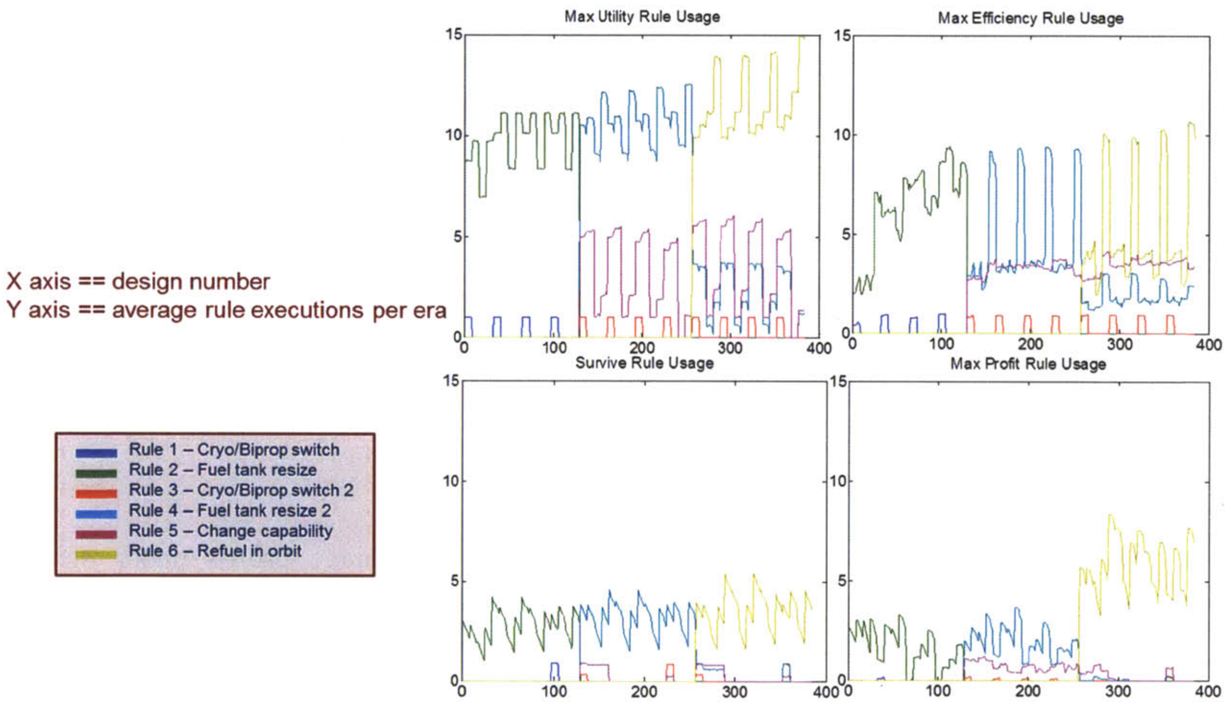


Figure 6-16: Space Tug Change Mechanism ("Rule") Average Uses per Era

In line with the insights from the ARI plot in the previous step of VASC, the fuel-related change mechanisms (2, 4, 6) are the most frequently used, as they provide the most value. There is also confirmation of one of the FPS insights, that the Maximize Utility and Maximize Efficiency strategies use significantly more changes than the Survive strategy, but it is now also clear that the Maximize Profit strategy is more in line with the Survive strategy, except that the DFC level 2 designs utilize the in-orbit refuel significantly more often because it is fast and inexpensive. Finally, mechanisms 1 and 3, the bipropellant/cryogenic propulsion switch options, are used less than once per era on average for every design under all strategies, making them strong candidates for exclusion.

The active FPN of the system can be tracked as it evolves over time, and statistics are reported on each design from this data. The results are shown in Table 6-18. Note that all of these are averaged across all 5000 samples, so it is, for example, the *average* best FPN achieved and the *average* average FPN for the whole lifetime. The Average (no fail) statistic represents the average FPN not considering any invalid 101 scores (as 101 is not a true FPN but just a numerical stand-in).

Table 6-18: Space Tug FPN Tracking Era Data

	MAX UTILITY				MAX EFFICIENCY			
Design	Best	Worst	Avg	Avg (no fail)	Best	Worst	Avg	Avg (no fail)
A	0.0	96.0	17.4	2.2	0.0	100.4	24.7	0.0
B	0.0	94.1	15.8	3.0	0.0	96.4	17.7	2.1
C	0.1	84.2	13.1	4.8	0.0	100.5	27.9	3.7
D	0.0	91.0	16.8	7.9	0.0	95.1	19.7	8.6
E	1.0	85.4	15.6	8.8	1.0	80.6	13.3	7.2
F	2.1	82.1	18.1	12.7	1.0	100.4	24.1	2.3
G	3.1	100.6	33.6	10.6	1.0	100.9	33.3	4.5
	SURVIVE				MAX PROFIT			
Design	Best	Worst	Avg	Avg (no fail)	Best	Worst	Avg	Avg (no fail)
A	0.0	99.3	20.1	1.4	0.0	100.5	25.6	0.3
B	0.0	97.5	19.3	2.9	0.1	97.9	20.1	3.2
C	0.0	93.8	16.5	4.3	0.0	99.9	25.5	4.1
D	0.1	96.1	26.8	16.2	0.7	100.4	38.5	19.9
E	1.0	87.3	14.4	5.5	1.4	97.0	22.9	8.5
F	3.2	100.8	38.2	16.9	3.2	100.7	38.3	17.3
G	3.7	100.9	44.0	21.2	2.9	100.7	38.2	15.5

Again, there are a number of useful insights that can be drawn from this data:

1. The DFC level 0 designs (A, B, C, D) have better *best* FPNs, but the other designs have better *worst* FPNs. This suggests that, for this case, *changeability is avoiding worst-case scenarios*, rather than switching between local optima in each epoch.
2. Design A has the best average (no fail) FPN for all four strategies, but is invalid too often to ever have the best overall average. This is because, as discussed earlier, it has very little fuel and will often run out and require a change, and if the current mission (epoch) does not last long enough to allow a change, it becomes invalid.
3. Design G is regularly ranked last or near-last in all categories. Its high failure rate and high cost are not compensated for by a large enough utility.
4. Design E appears to be the best compromise between strategies for this metric, as it ranks very highly amongst the designs of interest in all strategies for both worst and average FPN. Again, this is potentially valuable if the stakeholder anticipates changing strategies over time.

Finally, it would be beneficial to consider the cost/benefit tradeoffs, or "going rates", for adding or removing changeability from some of these designs of interest. This sort of "under the hood" information can be extremely valuable to non-technical decision-makers, who may have difficulty differentiating between designs or conceptually understanding changeability development and usage. For example, suppose that the system stakeholder is leaning towards Design D for all of the positive reasons uncovered in this analysis so far. Design D is a DFC level 0 design though, and the stakeholder is interested in the potential benefits of adding additional changeability. For Space Tug, Design D can be compared to the designs that are the same except for higher DFC level investments. This comparison is shown in Table 6-19.

Table 6-19: Space Tug Average Profit Era Data (Design D and Similar)

Design	DFC Level	Revenue (10^4 \$M)	Cost (10^4 \$M)	Profit (10^4 \$M)
D	0	7.7	0.7	7
256	1	7.4	0.8	6.6
384	2	10.7	0.3	10.4

It appears that the DFC level 1 equivalent of Design D is a less effective design; the percentage mass penalty is especially taxing on this high-functional design vector, and the marginal benefits of DFC level 1 do not make up for it. However, the DFC level 2 variant, Design 384, earns significantly more average profits over its lifetime, which may make it of interest to the stakeholder. Interestingly, a part of the increased profit is a decreased cost despite more investment in DFC and thus a higher initial cost; in this case, the accumulated costs of design changes over the era are reduced enough by the more advanced change mechanisms to make up for the initial cost increase. Of course, this increase in initial cost may represent a barrier to entry depending on available funds for the project. This initial cost versus lifetime benefit tradeoff of varying changeability can be calculated for any design of interest, as shown in Table 6-20 for three designs that emerged as valuable over the course of this case study. The tradeoff can also be calculated using any lifetime value metric (for example, Average FPN). Alternatively, if the assumption is deemed acceptable, a discount rate could be applied to the revenue stream and DCF analysis could be performed to provide a more prescriptive judgement about whether or not the tradeoff between initial cost and lifetime profit is "worth it" when considering the time value of money.

Table 6-20: Space Tug Changeability "Going Rates"

-DFC tradeoff	Design	+DFC tradeoff
N/A	D	+$544M initial cost, +$34B profit over 10 years
-$80M initial cost, -$4B profit over 10 years	E	+$80M initial cost, +$21B profit over 10 years
-$384M initial cost, -$20B profit over 10 years	F	N/A

Not all cases will be modeled with a design variable explicitly tied to the presence of particular change mechanisms, like DFC level does. For problems such as these, additional modeling efforts would need to occur in order to generate these going rates, particularly for considering adding new change mechanisms, but with the benefit that they need only apply to a perturbation in the chosen design and not modeled for the entire tradespace.

6.2.6 Discussion of VASC's Application to Space Tug

Multi-Epoch Analysis alone could identify design D as a valuably passive robust design, with the highest Pareto Trace and a Fuzzy Pareto Shift (FPS) distribution with nearly all of its weight on zero, indicating few changes. Era Analysis confirms this result, showing Design D to have the most consistent performance across changeability strategies and to have the most accumulated utility across an average era among the designs of interest. Design E, which has the same parameters as D but for a lower capability level and a higher DFC level, was identified in Multi-Epoch Analysis as a fuzzily passive robust design, but mostly dominated by D due to its lower utility and higher cost under each preference set. However, Era Analysis reveals design E to have one advantage over D, as its improved changeability allows it to satisfy slightly more contracts per era (albeit at a lower utility), which may be valuable if uninterrupted viability is of particular interest to system stakeholders. Meanwhile, Design F was identified in Multi-Epoch Analysis as valuably changeable using its FPS distribution showing a large number of efficiency-improving change options in the epoch space under most strategies. However, the magnitude of that value is only shown more quantitatively once Era Analysis is performed, with Design F demonstrating the highest average lifecycle profits under the Maximize Profit strategy of any design-strategy combination as it leverages its low-cost change mechanisms to great effect as contexts change over time. The "going rate" for changeability addition or subtraction from each of these three designs was also able to be calculated from the era simulation data.

This case study demonstrates a number of fundamental aspects of VASC that make it such a powerful design approach. Of particular interest is the large amount of descriptive data generated, allowing system designers to see Space Tug's changeability usage measured in as many ways as possible. Passively robust and highly changeable designs were able to be compared on multiple dimensions capable of capturing their value on similar scales, and make judgements about their behavior when exposed to uncertainty. Space Tug also demonstrated, in contrast to X-TOS, the range of insights and behaviors that can be captured in even a small case study when the designs and epochs are sufficiently differentiated.

6.3 Satellite Radar

After completing analysis on the X-TOS and Space Tug data sets, VASC was attempted to be applied to a Satellite Radar case study. Satellite Radar is a satellite constellation designed to provide 24-hour all-weather imaging and tracking of strategic ground targets. This case study is significantly larger than the previous two, featuring 23,328 designs (from 12 design variables), 648 epochs (from 6 epoch variables), and 8 change mechanisms (Ross et al., 2009). In addition to this, the Satellite Radar model is designed to track the entire system lifecycle through the Design, Build, Test, and Operations phases, with different expected phase schedule times for each design and different change mechanisms available at different costs in each of the phases. Together, these features combine to make for a robust case study, but one that is significantly more challenging to implement in both the multi-epoch and era domains than previous VASC cases.

One of the main barriers for the use of VASC on Satellite Radar is computational power. VASC is a data-intensive analysis approach, and a considerable amount of computation is performed, taking as much as a day on a single workstation class computer even for the smaller case studies already presented. Although the *time* requirement of computation is nontrivial, it was discovered that the *memory* requirements for analysis and simulation were the main obstacle for the application of VASC to this case. In general, the scaling of VASC up to larger case studies is of considerable importance for its further maturation and development as a design approach, and thus the discussion on the topics of computational scaling techniques is held later in this thesis: refer to Section 7.1.3 for more information on this subject.

The other barrier to applying VASC on Satellite Radar is the newly included wrinkles of schedule tracking and lifecycle phases. Conceptually, these are not irreconcilable with VASC. Because lifecycle phase determines what change mechanisms are available and how much they cost, the strategies must be applied *to each phase separately* in Step 3. In some sense, this suggests that a multiple-phase study in VASC will simply take the form of VASC applied independently to each phase in steps 3 and 4, with the results intelligently combined by the design team to draw conclusions. Era Analysis remains the same, as modeling a system lifecycle by necessity requires progressing through the phases together (not independently), but demands a more sophisticated era constructor and simulation code. Since each design has an expected duration for each phase, when a transition is executed the time cost needs to be applied as a delay in schedule and then the new expected duration would be the duration of the end state minus the time already spent in that phase, perhaps modified by some "similarity factor" between the initial and end states of the change. Phase changing would occur independently of epoch switches, as soon as the schedule time is reached. In addition to providing a more realistic lifecycle model, the inclusion of phases allows for additional interesting strategies and Era Analysis metrics. Because time to fielding the system is a particularly important design criterion for many systems, "minimize expected schedule time" is a potential strategy of interest. Alternatively, total schedule delay (transition time delay costs plus potential extended schedule time of new design) can be used as a threshold, as in "maximize utility without increasing expected schedule time by more than 1 year." Average total time to fielding including all transitions can be an output of Era Analysis as well.

Thus, VASC has the potential to grow and make large improvements in the design of even more complicated systems than the ones considered in the full case studies of this thesis. It is the author's hope that VASC can be implemented on the Satellite Radar case study in the near future.

7 Discussion

This section will cover a range of important discussion points related to VASC and its use. First, the applicability of VASC to different problems is considered, with respect to system type, changeability type, and effort required. Then, a direct comparison to Real Options Analysis is presented using the Space Tug case study, with particular attention paid to the differences in the insights derived from the analysis. Finally, a number of opportunities for future work in extending VASC are presented.

7.1 Applicability of VASC

7.1.1 Problem Types

VASC was originally created for use on engineering system design problems for which changeability offered potential value in defending against the presence of exogenous uncertainty. This is a very large class of problems, and some are not as well suited to VASC as others. First, let us consider the sources of uncertainty. To work with VASC, the uncertainty that the system is exposed to must fit within the EEA framework: that is, it must be able to be viewed at fixed, discrete levels in order to be used as an epoch variable and allow for the tradespace-paradigm of uncertainty to be used. This is obviously well suited to categorical variables (i.e. war/peace uncertainty) but is also not a problem for many continuous variables because, as with design tradespaces, it is often simple to enumerate the continuous variable with a small set of fixed levels. However, some problems feature uncertainties for which the continuous nature of the uncertainty is of some importance. As a simple example, consider a true financial problem, for which the underlying uncertainty is a stock price. The stock price is expected to change on a near-continuous basis, and the price itself is a continuous variable. Small, frequent changes are an important characteristic of stock prices, their valuation, and strategies for managing stock assets. To capture this in EEA, one would need to use an epoch variable (stock price) with an extremely large number of potential levels and the duration of epochs would be extremely short, limiting the conceptual benefits of Multi-Epoch Analysis and increasing the complexity and time requirements for both Multi-Epoch and Era Analysis. Even though uncertainties like this *can* be modeled in EEA and analyzed in VASC, other techniques (in this example, options theory) are likely superior for the problem type.

Second, the type of changeability inherent in the system should be considered. A distinction has been made between systems with differing amounts of change mechanisms, and end states per mechanism, as in Figure 7-1. Most prior changeability research has been focused directly on the lower-left quadrant, where designs have a relatively low number of change mechanisms, and those mechanisms enable transitions to a relatively low number of alternative states, and VASC is fully capable of working with these types of systems.

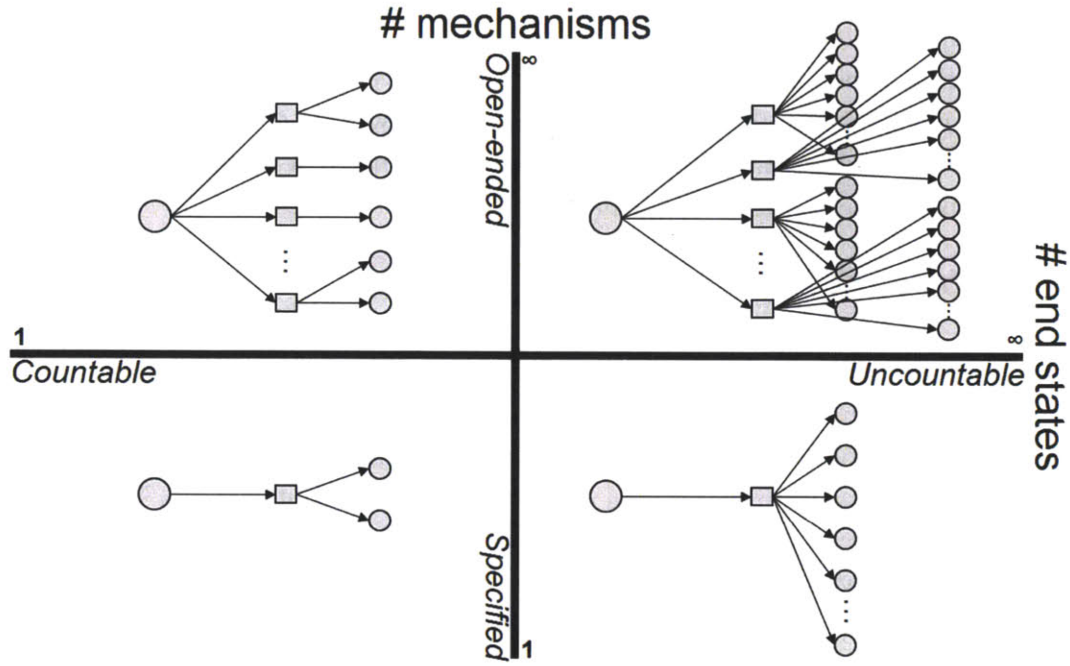


Figure 7-1: Countable and Uncountable Changes (from Ross et al. 2008)

Progressing to other quadrants of the figure, however, can present some problems. Increasing the number of mechanisms available to the system is mostly a computational problem; the metrics that directly compare change mechanisms (ARI, era-level usage frequencies, etc.) scale linearly with the number of mechanisms in play, but creating the multi-arc transition full accessibility matrix that allows VASC to consider the benefits of mechanism coupling has a worse scaling. Thus as the number of mechanisms increases, the number of arcs able to be considered for multi-arc paths in a reasonable amount of time is lowered. More detail on this is included in Section 7.1.3. Scaling up the number of available end states for each mechanism is not a problem for the VASC metrics, as the strategy step will always down-select to one path, but again will result in longer computation times for the full accessibility matrix, as there will be more possible path options to consider when determining the non-dominated paths to each design point. In the limit as the number of end states becomes infinite, such as for a change mechanism that modifies a continuous design variable that is split into an increasingly high number of levels, the problem diverges from the concept of tradespace exploration. Currently, VASC has not been tested on non-tradespace problems, but details on the possibilities for extending VASC to that problem type are included in Section 7.3.1.

Having covered the uncertainty and the change mechanisms, the third and final part of the VASC description is worth mentioning, the "engineering system design problem." Any problem that can be framed using EEA and a changeability strategy are candidates for VASC, not just engineering systems. However, without validation it is difficult to say how much benefit the insights achieved from VASC would give to a design problem in an alternative field. Perhaps of key interest here is the problem of engineering design for Systems of Systems (SoS), which have many similarities to basic systems but also fundamental differences. A discussion of the steps necessary to extend VASC for SoS design is included in Section 7.3.4.

Finally, it is important to note that VASC is a powerful tool for revealing information about changeability early in the design process, but not all models will generate equally interesting results. In order for VASC to deliver compelling insights about the difference in changeability usage and value between designs, the designs and the different epochs they are used in must themselves be sufficiently differentiated. This was demonstrated quite effectively by the difference in results between the X-TOS and Space Tug case studies. For X-TOS, which was created such that every design had access to the same change mechanisms and the uncertainties were entirely small-shift preference uncertainties, the returns of VASC were limited to small statements about the value of changeability in the multi-epoch space, with little differentiation between designs at the era level. On the other hand, Space Tug featured varying amounts of change mechanisms and more significant variation in the epochs, and VASC was able to find considerable differences between the usage of change mechanisms and the lifetime value of the designs of interest that were not obvious. Therefore, it is important to consider the degree of design/epoch differentiation in the system model when choosing whether or not to apply VASC, as the return for the effort is appreciably lower for problems with a smaller design space or uncertainty space.

7.1.2 Human Requirements

VASC is a definitively human-in-the-loop process, requiring inputs from system stakeholders and technical domain experts along with the guidance of a competent system designer possessing familiarity with the method and changeability in general. Although the metrics themselves are formulaic, for VASC to successfully provide insight into available valuable changeability, careful attention must be paid throughout the process, particularly in the early setup phases of the approach. Additionally, the more in-depth insights must be uncovered from the metrics because they are frequently not apparent at first glance; thus, the system designer employing VASC must be familiar with the metrics and how to interpret them as well as with the technical aspects of the system. Table 7-1 shows the required and recommended types of human input for VASC.

Table 7-1: Human-in-the-Loop Actions in VASC

Involved Person	VASC Step	Human Input
System stakeholder / decision-maker	Setup EEA	Potential uncertainties
		Preferences / utility functions
	Select Designs of Interest	Favorite designs (from previous analysis or otherwise grandfathered in)
	Define Strategies	Intended use of changeability
	Era Simulation and Analysis	Preferred metrics for establishing "going rate" tradeoffs
Technical domain expert	Setup EEA	Design variables
		System design → performance model
		Change mechanisms
		Potential uncertainties
	Select Designs of Interest	Intuitive / expert opinion of potentially valuable designs
	Era Simulation and Analysis	Models / assumptions for evolution of uncertainty
System designer / VASC driver	Setup EEA	Potential need for additional modeling related to change mechanisms, if not present in existing system model
	Select Designs of Interest	Select designs using screening metrics, preferably with substantial differences in design variable levels
	Define Strategies	Suggestion of strategies if stakeholder is uncertain
		Suggestion of alternative strategies for comparison
	Multi-Epoch Analysis	Interpretation of data/graphs
	Era Simulation and Analysis	Interpretation of data/graphs

For the best results to be obtained, all human parties involved in VASC should be familiar with the approach, with at least one system designer confident enough to be the driver of the method. As of this time, there has been no research on the topic of teaching VASC to systems engineers. However, the main concepts of VASC should be readily accessible to any systems engineer with an understanding of changeability and other changeability valuation techniques. The most important concepts to grasp are the use of EEA and the process of applying a changeability execution strategy to a set of available change options, as this will be enough to set up and run VASC. The more in-depth insights able to be extracted in the later steps come from increased understanding of the metrics utilized in VASC, particularly what they are measuring and what can be learned from their application. Observationally, as VASC has been presented to other systems engineers in both informal and formal settings, viewing an example case study has typically inspired the most comprehension of the *logic* and *purpose* behind each of the steps. Therefore, it is recommended that case studies such as the Space Tug be used in any early attempts to familiarize people with VASC, until a more detailed set of lectures or a curriculum for learning VASC is developed.

7.1.3 Computational Requirements

While the previous section discussed the many elements requiring human involvement in the approach, many tasks in VASC are performed by a computer. In particular, the algorithmic creation of the full accessibility matrix, the calculation of the metrics, and the simulation of eras are all computer automated. Table 7-2 shows some of these tasks and a description of their approximate scaling in computational time.

Table 7-2: Approximate Scaling of VASC Activities

Activity	Worst Scaling Variable	Approximate Order
Create full accessibility matrix with non-dominated multi-arc transitions	Primary: # arcs Secondary: # change mechanisms	$(\#mechanisms)^{\#\ arcs}$
FPN	# designs x # epochs	Linear, but depends on the shapes of the tradespaces
NPT (and fuzzy/effective versions)	# designs x # epochs	Linear
Strategic change path selection	# designs x # epochs	Linear, but depends on complexity of decision logic
FPS	# designs x # epochs	Arithmetic only, but requires FPN and strategy
Rule removal study	Primary: # arcs Secondary: # change mechanisms	Requires a repeat of full accessibility matrix creation
Era simulation	# eras per design	Linear, but more complicated strategies requiring era information can be more affected by # epochs per era

The scaling of VASC to large cases is overall quite reasonable, with the vast majority of tasks featuring only linear scaling with both the number of designs and number of epochs. The main exception to this rule is the creation of the full accessibility matrix out of multi-arc transitions, which scales by the number of arcs considered raised to the power of the number of change mechanisms in the system. Thus, for systems with a large number of change mechanisms, the maximum number of arcs that can be considered at one time is effectively reduced. Conceptually, the "full" aspect of the full accessibility matrix represents that every possible path from initial to final design, via any number of arcs, is shown and technically this may not be true if the maximum number of arcs is restricted. However, for case studies approximately the size of the ones included in this thesis, 2 to 3 arcs will capture the vast majority of mechanism coupling benefits. There are no drawbacks to considering only a subset of non-dominated multi-arc changes other than missing potential "higher arc" paths of interest.

While the computational time scaling properties of VASC are reasonable, the memory requirements are much more restrictive. VASC, as a descriptive, exploratory process, generates a large amount of data for the system designers using it to analyze. For large systems, this data can quickly approach the limits of the memory capacity of workstation-class computers. Consider the full accessibility matrix alone: its complete size will be a $N_{designs}$ by $N_{designs}$ array, for which each entry will have a number of values equal to the number of non-dominated change paths from the initial design to the final design, times the number of dimensions of cost being recorded for each change. As mentioned earlier, this amounts to 20 GB of data for the Satellite Radar case study, considering *only* single arc transitions.

However, VASC is not infeasible for case studies as large as, or larger than, Satellite Radar. What will be key in implementing VASC on such cases is intelligent use of data management and parallelization. MATLAB® stores all of the variables in the workspace in active RAM and when the amount of data in the workspace nears the maximum amount, all operations begin to slow down exponentially. Ideally, only the data needed at any given time would be in the workspace. However, since loading and unloading data from the workspace takes a finite amount of time directly proportional to the amount of data, there will be some optimal amount of data to be stored and called at a time. For example, during era simulation it is unnecessary to have access to the strategic changes for any design but the current design; a code structure that loads only the relevant changes in active memory and swaps them out for the appropriate set whenever the system transitions to another design would minimize memory requirements but require a load/unload of data after every transition. The current method (all data stored as active) maximizes memory requirements and *would* minimize time requirements if not for the slowdown associated with maxing out a computer's RAM. Thus, an optimal point for the time to run VASC lies between these two extremes, loading and unloading small batches of data as they become necessary. The optimal point will be a function of both case size and computer memory. However, the ability to load and unload data on the fly is not a feature of the existing VASC code base and thus the potential speed increases of such a method cannot yet be benchmarked.

A potentially valuable addition to VASC is parallelization: tasking multiple computers/processors with different jobs simultaneously to accomplish more work at once. VASC is well suited to parallelization in all of its steps, as the calculations required are mostly independent across designs and epochs. As an example, consider the selection of strategic change paths in Step 3: the logic used to determine if and how a given design will transition in a given epoch is completely independent of all other determinations of the same type, thus allowing the process to be split up amongst any number of available computers with only the small additional cost of eventually combining all of the results together. This is also true for Multi-Epoch Analysis (each design can be evaluated separately) and Era Analysis (each simulated era is independent of all others). Again, the existing VASC code base is not equipped to handle parallelization, but implementing it is a relatively low-complexity matter, as it simply requires specifying which designs/epochs should be evaluated in each function rather than requesting all of them at once. While the active memory reduction suggested in the previous paragraph would allow less powerful computers to perform VASC, the parallelization technique will be more critical in speeding up the total computational time. Because of the high degree of independence between tasks, two computers will do the computations in about half the time and so on, with diminishing returns only appearing once the number of computers exceeds the combinations of designs/epochs or designs/eras in consideration: easily in the millions for even moderately sized studies.

7.2 Comparison to Real Options Analysis

Real Options Analysis (ROA) is, to some extent, the preferred approach of the status quo for valuing changeability in engineering systems. VASC is attempting to provide an alternative to ROA, so it is definitely of interest to compare the two approaches directly. The following subsections describe a potential means of framing the Space Tug case study from Section 6.2 using ROA, and then compare the two approaches.

7.2.1 Space Tug with ROA

To keep the two versions of Space Tug as similar as possible, this study applies ROA to the same design space and with the same change mechanisms. Because the DFC level variable is embedded in the design space, the change mechanisms cannot be used as distinct options. In this case, the "options" being valued are the different DFC levels themselves, which are associated with varying levels of changeability. The nature of the Space Tug design space, with the "options" being included or not included in each design point, prevents the designation of an option price that must be paid to add changeability. The closest construct for comparison is the difference in cost between designs that are the same except for their DFC level, which can be roughly interpreted as what was "paid" for the increased changeability. However, the change in initial cost is not the only effect of including the change mechanisms. Recall that the DFC level was implemented as a *mass* penalty, which also reduces the ΔV available to the system. This performance decrement is not captured under the traditional ROA definitions of an option price. Also, since the "options" are included by default in particular designs, instead of valuing a design with and without an option (and then ascribing the difference as the value of the option, accepting the option as "worth it" if its value exceeds its cost), the analysis will have to simply compare designs.

The two main modeling tasks are the modeling of uncertainty and the modeling of system changes. For uncertainty, the user preferences should vary over time. Because there are three attributes, this is a three-dimensional uncertainty, which makes it extremely difficult to model as Geometric Brownian Motion. Instead, that behavior can be approximated with a pseudo-binomial lattice. A parameter can be specified that each attribute weight in the multi-attribute utility function will be multiplied or divided by (with equal probability) at given intervals (and then re-normalize to 1). This will allow for a range of user preferences and reordering of the relative value of different performance parameters, where a higher multiplier parameter results in more variability. Also note that the weights are simulated as changing at random intervals between 1 and 12 months, to match the VASC application of Space Tug.

For the system change model, a procedural, three-tier priority is defined for executing available change mechanisms whenever the attribute weights change. This priority is:

1. If more fuel is needed, refuel completely
2. If swapping propulsion types is beneficial to short run profit, execute it
3. If increasing grappler capability one level is beneficial to short run profit, execute it

This is only meant to model a reasonable decision-making process when it comes to executing change mechanisms, prioritized roughly in order of their importance/urgency.

With this setup, sample lifetimes can now be simulated, tracking the revenues (using the same revenue model as with VASC) and costs of the system over time and applying a fixed discount rate in order to calculate the system's Net Presents Value (NPV). The main output and decision metric of most ROA studies is NPV, typically viewed as a cumulative distribution or "target curve" to summarize the effects of uncertainty. An example of some such target curves is shown in Figure 7-2, for a design with its three levels of DFC and also a DFC level 0 version that never executes any change mechanisms.

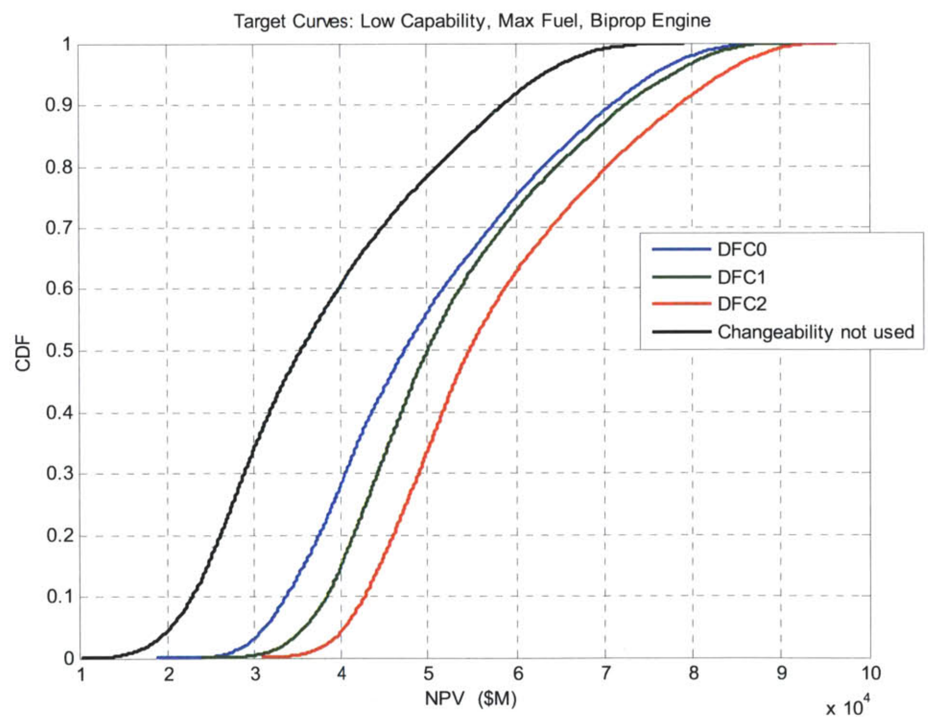


Figure 7-2: Space Tug Varying DFC Level Target Curves

Interpreting a target curve plot is mostly related to the idea of "dominance", where a design is considered to dominate another if its target curve is completely to the right and below, implying that its percentile chance of exceeding any fixed lifetime NPV is greater than the dominated design's. In this case, each successive level of added DFC dominates the previous. If the DFC levels had associated option costs, the expectations (ENPV) of these distributions could be compared, and the DFC declared to be worth the investment if the expected increase in NPV was greater than the option price. Since the option prices are embedded in the initial cost of each design in this case, the chart is read alone: since the DFC level 2 design is dominant, it is the best selection.

If the entire design space is simulated, the results could be queried in order to find different "best" designs under different criteria. For example, frequently the design with the highest ENPV is of interest, but an extremely risk averse stakeholder may prefer the design with the best worst-case scenario, or minimum, NPV. Three of these potentially interesting designs are shown in Table 7-3, and their target curves are shown in Figure 7-3.

Table 7-3: Space Tug ROA Designs of Interest

Rationale	Design #	Grappler Capability	Prop.	Fuel	DFC Level	P_0	ENPV	P_{100}
Least Risk (highest P_0 NPV)	319	Medium	Nuclear	10,000 kg	2	4.01	6.88	9.75
Most Opportunity (highest P_{100} NPV)	286	Low	Nuclear	3,000 kg	2	3.19	6.49	9.79
Best Expectation (highest ENPV)	352	High	Nuclear	30,000 kg	2	2.97	7.32	9.70

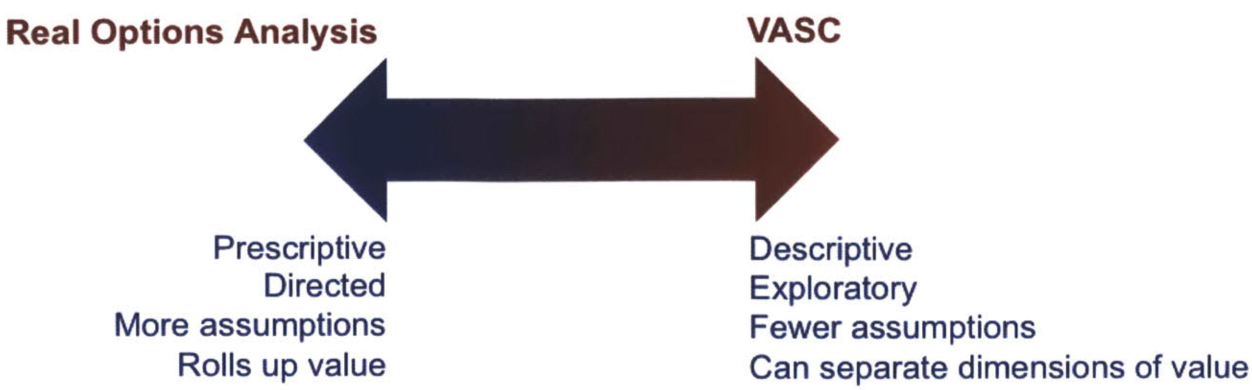


Figure 7-5: Notional "Analysis Spectrum" from ROA to VASC

ROA is designed to provide focused, prescriptive judgements such as: "This change option is 'worth it', because the benefits exceed the costs" or "This design is the 'best' design for this metric". The benefit of this sort of analysis is that it is simple and easy to convey knowledge to non-technical stakeholders for them to make decisions on. On the other hand, VASC is designed to explore as much of the design space as possible and provide large amounts of descriptive data about the usage of changeability and its value, using as many dimensions as possible. The reduced assumption set inherent in VASC also reduces the set of variables for which sensitivity analysis is critical.

7.2.3 Computational Benefits

VASC is a computationally heavy approach, and thus will often require more man-hours and computational time than ROA to perform properly. However, VASC does have the benefit of its predetermined change executions via the strategy step, which negates the need for a reevaluation of the change logic at each decision point. Thus, for cases that require a very large number of simulations or those for whom the changeability execution decision logic is complex and takes an appreciable time to run, VASC will see a speed advantage over a large number of simulations. The predetermined changes also allow for Multi-Epoch Analysis, for which there is no corollary in ROA. Multi-Epoch Analysis does not require simulation and the multi-epoch metrics are extremely fast to compute after the completion of the strategy determination.

7.3 Future Work

VASC is still young, and there are many avenues for improving and extending the approach. The following subsections will cover a number of possible extensions or alterations to VASC that may prove to be valuable research threads in the near future.

7.3.1 VASC without Tradespaces

Currently, VASC is framed entirely within the field of tradespace exploration. The synergy between the agent-mechanism-effect change framework, EEA, and tradespace exploration is strong enough that originally formulating VASC as a tool for advanced tradespace

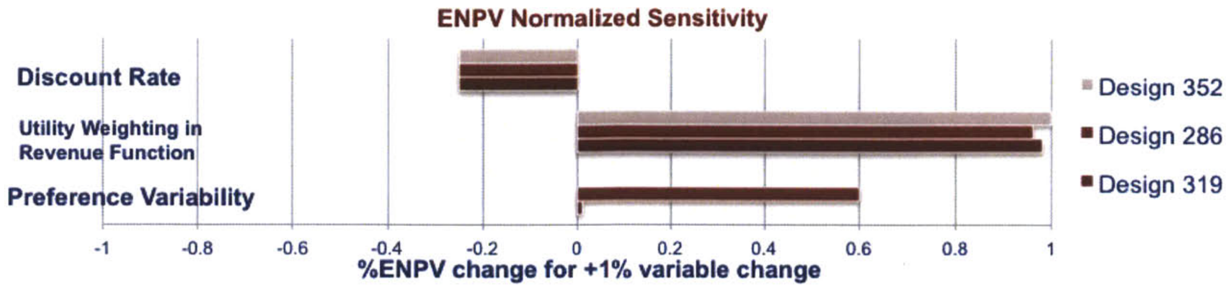


Figure 7-4: Space Tug ROA Sensitivity

This plot shows a nearly one-to-one relationship between ENPV and the assumed weighting of the revenue function's utility aspect, prompting consideration of the confidence level of that assumption. Also of note is that the preference variability causes very little sensitivity except for Design 286, the "Most Opportunity" design, which benefits significantly from increased variability.

7.2.2 Insight Benefits

It is important to recognize that the term "Real Options Analysis" covers a wide range of design analysis techniques, and that this was only one possible interpretation of Space Tug into an ROA framework. However, it required a fair number of assumptions that aren't necessarily appropriate for Space Tug; for example, it is not clear that the modeling of the individual attribute weights in the utility function should be a stationary, recombinant process, and this may make the results of the study questionably accurate. The EEA model of uncertainty captures the concept of distinct users with different preferences requiring Space Tug services at random times more accurately than this stationary process, as it does not require any form of continuity from one user to the next (unlike the process used for the ROA study). There is also a significantly reduced range of insights able to be obtained from ROA, as more emphasis is placed on monetization and aggregation of lifetime value, without any other significant value metrics to consider. Still, ROA and VASC are not completely different. Indeed, it would not be unreasonable to apply an EEA uncertainty framework with the procedural change mechanism execution logic, or alternatively to apply a more detailed changeability strategy to the simple uncertainty model. VASC and ROA operate on a sort of "analysis spectrum" that is notionally pictured in Figure 7-5.

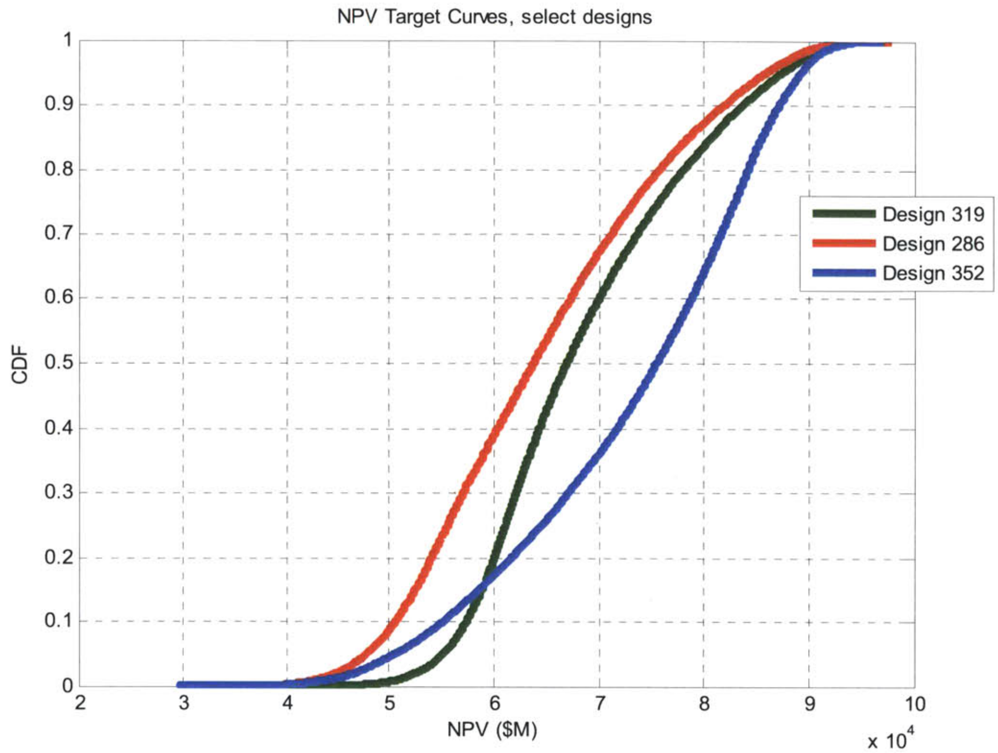


Figure 7-3: Space Tug ROA Designs of Interest Target Curves

These designs at least do not strictly dominate each other (although the Least Risk design is very close to doing so against the Most Opportunity design). The system stakeholder would likely make his decision from amongst designs like this, simulated and determined to be the "best" according to some criteria.

Also note that sensitivity analysis is important for ROA studies, as the results typically depend on multiple parameters introduced for the monetization analysis. Performing the simulations again with different parameter values allows for the sensitivity of some output (usually ENPV) to be calculated, as in Figure 7-4, for the same three designs of interest with respect to the chosen discount rate, the weighting of utility in the revenue function, and the variability parameter for the attribute weights.

exploration was entirely natural. However, tradespace exploration is not a perfect design technique and may not be appropriate or desired for every problem. Drawbacks associated with tradespace exploration include a potentially large time commitment for evaluating many design points, and the possibility of missing fine details due to the granularity of sampling continuous design variables at fixed levels. A clear means to improve VASC, therefore, would be the development of a modification to VASC that does not require full tradespaces to perform the analysis, allowing the main concepts of VASC to be applied to smaller, single-point design studies.

Fortunately, VASC should be readily amenable to this problem type. The most noticeable departure from the original formulation of VASC would be a shift in the "anatomy" of a change. Rather than a change mechanism enabling a design to traverse a path to any number of other specified designs, change mechanisms would open up design variables to be varied anywhere within their feasible bounds. To some extent, this is akin to the concept of "open-ended" transitions in Ross's work with Multi-Attribute Tradespace Exploration, in that designs are able to change to previously unspecified and unevaluated points (Ross, 2006). With this understood, the remaining barrier is adjusting the idea of what a changeability strategy does in order to take into account this new design transition type. Essentially, the decision logic that previously selected from amongst predetermined change paths would have to be changed into a multivariable optimization algorithm, with a degree of freedom in design variables determined by the available change mechanisms and the objective function determined by the strategy's goal. Step 3 of VASC at this point essentially becomes an optimization problem, with the strategy as the objective function and the change mechanisms defining the feasible design space. Setting up and running this optimization would make the strategy resume its normal role: determining what change a design would make when faced with any possible epochs. Multivariable optimization can also be used to find design space Pareto fronts, enabling the use of many of the VASC multi-epoch metrics. Finding these Pareto fronts can be accomplished using methods such as the Normal Boundary Intersection, Adaptive-Weighted Sum, or similar techniques (Das and Dennis,1998; Jilla and Miller, 2001; Kim and de Weck, 2006).

So, from a feasibility standpoint, the non-tradespace version of VASC is essentially just one with a multivariable optimization embedded in the strategy step. This would allow for the Multi-Epoch Analysis and comparison of any number of chosen design points. Computational intensity rises significantly, however, when progressing to Era Analysis, at which point the multivariable optimization will need to be run at every epoch switch in every sample era, as the optimal changes will not have been pre-calculated for the "changed" designs as they were for the other designs contained within the tradespace in the original formulation. This eliminates one of VASC's advantages over previous changeability analysis techniques, which was the ability to exploit EEA to pre-calculate desired design changes and thus not require continual reevaluation of the change logic during simulation. The use of multivariable optimization also requires significantly more tuning effort by the system designer in order to prevent possible inconsistency in change selection, as multivariable optimizers are often very sensitive, and small inaccuracies may build up over the course of a simulated era. Regardless of these complications, the possibility of using VASC without needing to generate and evaluate a large tradespace of designs is both feasible and attractive, and further research efforts could readily create a multivariable optimization variant of VASC and test its effectiveness at generating insight.

7.3.2 Design "Families"

One interesting emergent behavior of designs under the guidance of a consistent changeability strategy is what this research has tentatively named "families" of designs. As epochs change over time, designs change as well and will frequently settle into some steady-state cycle between a set of designs: the "optimal" design points in each epoch that are reachable from the original design. These "families" of optimal designs that share duties across the epoch space are potentially better descriptors of lifetime performance and value than the initial design point or any transient behavior in gradually changing into the family. The families are also the end result of applying stakeholder changeability preferences over time on a static design, implying that in at least some way they are more desirable than other designs. They are also computationally easy to find without simulation, as a simple algorithm can simply take the matrix of designs' targeted end state designs in each epoch and loop the logic until each design is associated with every other design it will ever possibly change into. Then, the steady-state families are the smallest subsets of the design space such that any design that reaches one of the member designs must reach all of the others. Note that some designs are able to end up in more than one family (or even never reach their associated family), depending on the order in which the epochs arise over time; however, once it enters a family, it will not leave. It is also easy to use this information to identify "source" and "sink" designs: those designs that will never be the end state of a change, and those that will never be the initial state of a change, respectively.

The potential use of design families in VASC is readily apparent. Era Analysis would no longer compare lifecycles of initial design points, but rather the performance of the various design families. Any given strategy may have single or multiple design families, so the comparison of lifetime value would be between, for example, "Maximize Utility Family A," "Maximize Utility Family B," and "Survive Family A." The understanding of design families, and using their information in order to tailor long-term system behavior and value, also opens up the possibility for the creation and implementation of short-term strategies to maximize the transient performance of the system (even if that is simply by converging to the family as quickly as possible). However, this raises the question of whether or not there is any purpose in building an initial design that is not a part of a steady-state family, as it is unclear if transient behavior serves any purpose for most systems. As an example of when transient behavior may indeed prove beneficial, consider a system planning a staged deployment, where the final stages are too expensive to be built up front. Another concern is that, if the system stakeholder anticipates modifying his changeability strategy over time, finding lifetime value behavior of families becomes unclear as designs in the same family under one strategy may diverge into different families when the strategy changes. Still, the consideration of design families may prove to be a significant improvement to the process of designing for lifetime value delivery, and it is recommended that future research look into the possibility of incorporating them into VASC or other analysis methods.

7.3.3 Long-term Strategies

One common response systems engineers have had upon viewing VASC is the shortsightedness of the changeability execution strategy. In any epoch, it considers the starting design point and determines the best possible accessible end design point according to the strategy, and then executes that change. The obvious question is, "If we are designing for lifetime value, why do our changeability strategies not consider the long-run?" The simplest answer for that question is that restricting ourselves to short-sighted changes, reactionary to the arrival of a new epoch, allows us to reap the computational benefits of Multi-Epoch Analysis, by predetermining all selected change decisions. The idea of planning for a long term goal with a strategy like "change to the design with the highest expected lifetime value" would require the iteration of steps 3 and 5 of VASC in order to even create the right strategy: selecting changes, simulating for lifecycle value, amending the original changes to match the strategy, and repeating until convergence. This is similarly true for stakeholders who want to utilize changeability in anticipation of epoch shifts, as they are simply "playing the odds" and attempting to make guesses about what to do with the system using their educated but uncertain knowledge of the future. Short-sighted strategies are also completely deterministic, and do not utilize assumptions about the future in order to *predict* future value. Reducing the number of assumptions used for valuing changeability was identified as a key improvement over existing methods, and thus short-sighted strategies fit directly into this goal.

None of this discussion has been to suggest that the desire to utilize a long-term strategy of this type is unreasonable: it simply presents many additional challenges. Right now, VASC is designed to compare multiple short-sighted strategies with the goal of finding the strategy and design combination that maximizes long-term value. Future research could look into long-term strategies and the time requirements for determining them, or the convergence properties of the strategy-era iterative loop, which is likely to vary by strategy and system. If long-term strategies prove to be feasible additions to VASC, then they would provide large potential benefits in finding high-value solutions.

7.3.4 Moving Beyond System Design

VASC has the potential to serve as a model of changeability valuation that extends beyond the field of technical system design. Many other fields could benefit from increased understanding and inclusion of changeability in their design. Of particular interest is the area of Systems of Systems (SoS) design, which has many similarities and key differences with traditional system design (Maier, 1998; Mekdeci et al., 2011). As a very brief definition, the designation SoS is typically used to refer to large, complex systems that have component subsystems that are themselves systems with some degree of independence or autonomy of purpose, which work together to achieve some higher scope goal. For example, a group of radio towers, satellites, and mobile communication stations may form an SoS designed to provide all-day communication for soldiers moving over a large area. The potential benefits of changeability for SoS are significant, as the interfacing of multiple subsystems presents a design challenge but an opportunity for mix-and-match or staged deployment, to which VASC should be able to help ascribe value. Future research could target the application of VASC to SoS, which may require some adjustment to the definition of the design space or the agent-mechanism-effect change framework and corresponding changes to the VASC process.

Alternatively, there is some potential to improve SoS changeability valuation by combining VASC with portfolio theory concepts, which have been previously used to analyze engineering systems (Walton, 2002). In this case, an SoS could be modeled as a portfolio of "asset" systems, which combine to deliver some total value. Changeability would be manifested as the ability to add or remove assets from the complete SoS. The main barrier to implementing this idea is that standard portfolio theory considers all assets *independent* in the value they produce, which is untrue of SoS by the very definition of working together to achieve a higher goal (Ricci et al., 2012). However, there is potential for research between VASC and portfolio theory, and the intuitive nature of the comparison between assets and SoS subsystems is very attractive.

7.3.5 Survivability through Disturbance Protection

VASC has the potential to be applied to what in many ways is the reverse of changeability: survivability. Survivability can be defined as the ability of a system to resist exogenous disturbances, or to prevent changes to the system from occurring *outside* of the control of the system stakeholders. VASC could be modified to model this problem with essentially the reverse of the current change mechanism definition. By defining exogenous change mechanisms that force the traversal of a change path with some sort of probability, perhaps associated with system degradation or outside attacks, VASC can be used to find the value of the system under the threat of these uncontrollable changes. The creation of some sort of "resistance mechanism" that blocks or reduces the likelihood or severity of those exogenous changes would then provide the value associated with the difference between the original and more survivable versions (Beesemyer, 2012). The value of resistance mechanisms can be calculated using the same concept as VASC's rule removal studies except that in this case, the system with the removed change mechanism is presumably the better variant. By preserving the data- and exploration-heavy use of VASC, this could possibly offer an improvement over existing survivability valuation techniques, which depend largely on aggregation and integration of value over the system lifetime (Richards et al., 2009).

7.3.6 Visualization Techniques

VASC, as an approach that generates a massive amount of descriptive data and relies on the system designer to make judgements, runs the risk of creating "information overload" for the analysts using it. It can be daunting to run the analysis through a computer, receiving gigabytes of data, with only immature tools designed for exploring the results. For this reason, effective data visualization techniques are of key importance for VASC and other similar methods. A simple but information-dense figure can do more to communicate information than any amount of descriptive writing or abstract data mining. The creation of this figure, however, is significantly more difficult than straightforward statistical analysis. This thesis relied on distribution graphs and color-coded tables in order to present some metrics in a user-friendly form, but many of the lifetime statistics output by era simulation were simply shown in unadorned tables, indicating plenty of room for improvement. Advanced research into visualization techniques could potentially make VASC easier to learn and perform for less-experienced engineers, allowing for the discovery of important patterns through interaction, such as is being done with interactive tradespace exploration (Ross et al., 2010).

7.3.7 Approaching other Changeability Problems

Finally, it is worth acknowledging again that VASC is only designed to assist in the *valuation* of changeability. Referring again to the changeability "lifecycle" shown in Figure 7-6, future research would greatly benefit the field of changeability in engineering by addressing both the *development/creation* of potential change mechanisms in a system and the *execution* of changeability.

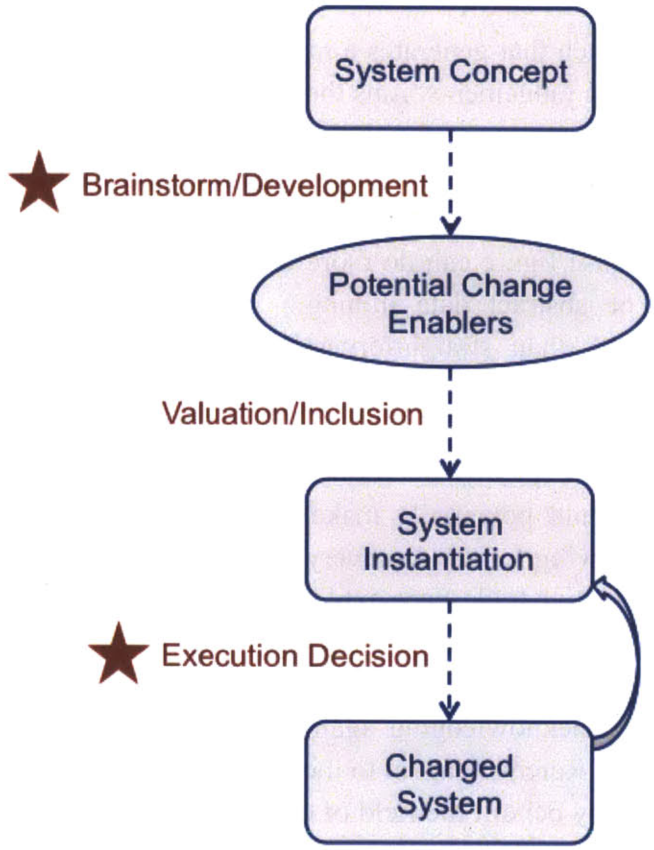


Figure 7-6: Conceptual Changeability "Lifecycle" – Other Areas in Need of Research

The concept of the changeability strategy in VASC essentially attempts to "guess" what the executions will be, but a more prescriptive technique for finding or suggesting optimal changeability strategies and for training stakeholders and decision-makers how to properly utilize changeability would provide massive improvements to the state of changeability by enabling more successful use of any included change mechanisms. Similarly, a more structured process for identifying potential locations for the insertion of changeability in a system, and the form that those potential change mechanisms should take, would be a great improvement over the current method suggested in VASC, which is informed entirely by expert opinion and previous knowledge. Research into DSM/ESM "hot spots," as discussed earlier in the literature review, has made headway in this task, but there remains room for improvement (Kalligeros, 2006; Bartolomei, 2007). Overall, a more complete treatment of the entire changeability lifecycle would be extremely beneficial to the effectiveness of including changeability in the early system design decision-making process, especially if the different aspects could be synthesized into a single method.

8 Conclusion

This research began with the goal of improving the metrics and methods for valuing changeability that were available to system designers. A literature review was used to identify three guiding research questions:

1. Can a method or set of metrics be created that values changeability with fewer inherent assumptions than existing methods?
2. Is it possible to analyze the multi-dimensional value of changeability using a single method, and without losing any fidelity?
3. Can metrics that accomplish the above goals also display qualities beneficial for generality, including dataset independence and context universality?

Now, at the conclusion of the research, all three of these questions can be answered in the affirmative. Epoch-Era Analysis was able to frame the uncertainty necessary for a valuation of changeability with relatively few assumptions compared to Real Options Analysis. Then, the combination of Epoch-Era Analysis with the changeability execution strategy allowed for the consideration of both the magnitude and counting value of changeability, by not requiring integrative lifecycle value calculations that conflate these two aspects. Finally, Fuzzy Pareto Shift was shown to be an independent and universal metric that can be used to value an executed change, while other new metrics were able to probe for other types of value derived from changeability.

8.1 Key Contributions

The eight key contributions of this research, as viewed by the author, are as follows:

1. **The creation of a means to approximate the eventual realized usage of system changeability, in the *changeability execution strategy*.** This allows design decisions to be made with the eventual *use* of changeability in mind, rather than attempting the valuation process by considering all *potential* change options. An example is shown in Figure 8-1.

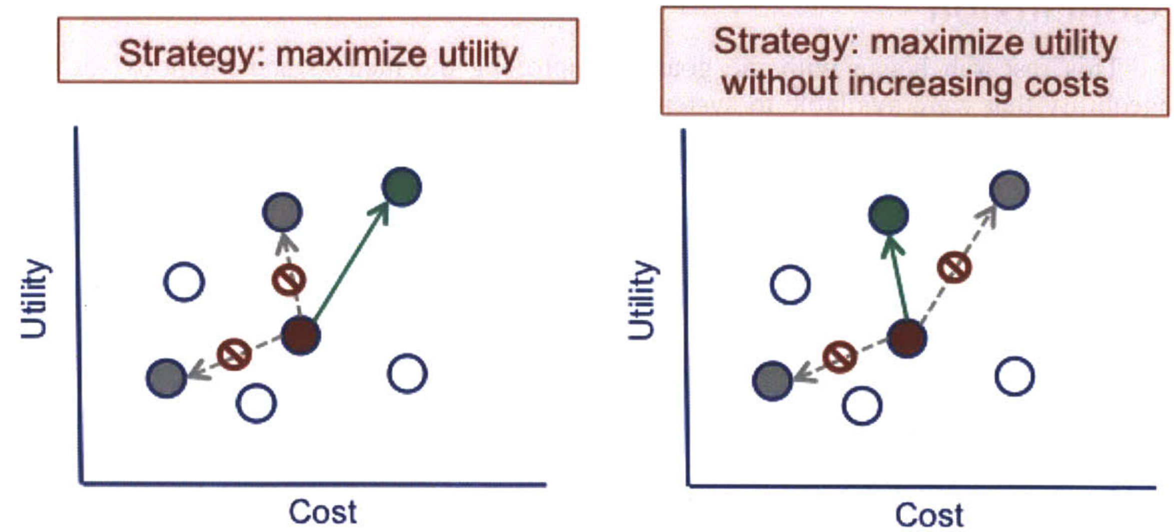


Figure 8-1: Application of Two Changeability Strategies to a Design

2. **The ability to properly consider both the *magnitude* and *counting* value of changeability in the same analysis, using the Epoch-Era Analysis framework combined with the changeability execution strategy.** This provides a more complete picture of changeability value for the benefit of system designers trying to determine the value of adding changeability in order to justify the costs of its inclusion. Figure 8-2 is a simple tradespace illustrative of the difference between magnitude and counting value.

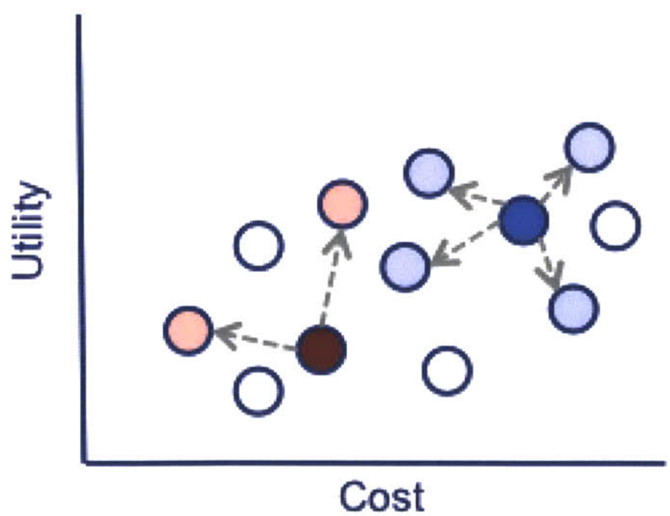


Figure 8-2: A Notional Tradespace Illustrating High Magnitude Value (Red) vs. High Counting Value (Blue)

3. **A suite of new changeability *metrics* with which to uncover difficult-to-extract insights about system changeability.** These metrics, particularly Fuzzy Pareto Number and Fuzzy Pareto Shift, also possess beneficial qualities such as dataset independence and context universality, which allows them to be used to compare the changeability of different designs under different circumstances more effectively. Others, such as Effective Normalized Pareto Trace, quantify system properties previously justified mostly heuristically, with statements such as "more changeability will lead to improved robustness against uncertainty." The metrics are summarized in Table 8-1.

Table 8-1: Summary of Metrics Developed in this Research

Metric Acronym	Stands For	Targeted Insight	Definition
eNPT, efNPT	Effective (Fuzzy) Normalized Pareto Trace	Robustness via changeability	Fraction of epochs in which design's changed end state is on the (fuzzy) utility-cost Pareto front
FPN	Fuzzy Pareto Number	Design efficiency	% margin needed to include design in the fuzzy Pareto front
FPS	Fuzzy Pareto Shift	Efficiency value of a change	Difference in FPN before and after executed change
ARI	Available Rank Improvement	Potential utility value of a mechanism	# of designs able to be passed in utility using a change mechanism

4. **A method for valuing changeability, with an emphasis on data collection, design exploration, and limiting embedded assumptions, entitled the *Valuation Approach for Strategic Changeability*.** VASC offers a distinctly different paradigm for changeability valuation than existing techniques. Real Options Analysis, as the most common of these techniques, is of particular interest for comparison. ROA was shown to provide a more prescriptive set of judgements, indicating directly what designs and options are the superior choices, at the cost of less data output and an increased set of assumptions, limiting its applicability. VASC attempts to solve the same problem as ROA but operates on the other end of the "spectrum" of analysis techniques, deliberately removing assumptions in order to increase general applicability and the amount of information that can be collected about designs of interest. The five steps of VASC are (1) Set up Epoch-Era Analysis, (2) Select designs of interest, (3) Define changeability execution strategies, (4) Perform Multi-Epoch Analysis, and (5) Perform Era Simulation and Analysis. A high-level data flow of VASC is pictured in Figure 8-3.

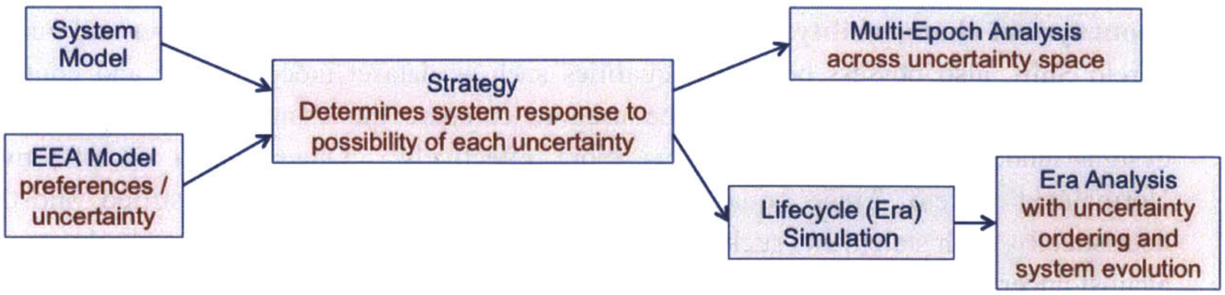


Figure 8-3: VASC Data Flow

5. ***Avoids monetization* of value, which was identified as a common obstacle for the application of previous changeability valuation methods to certain systems.** Monetization is the key assumption that limits the applicability of ROA to many engineering systems, which do not deliver value in the form of revenue. As demonstrated, VASC can utilize a monetized revenue function when appropriate, but is not reliant on it for insight into the value of changeability.

6. **A *repeatable*, and optionally *iterative*, method that increases the amount of information available to system engineers early in the design process.** VASC is meant to be applied early in the design process. By assisting system designers in understanding the potential value of changeability early on, VASC enables them to make more educated decisions when they have the most leverage. Additionally, part of the value of changeability is associated with increased leverage and freedom later on in the lifecycle of the system, so the justification of the inclusion of changeability *also* improves this frequent system design challenge. VASC is structured to be repeatable, such that different designers will come to the same information, and hopefully the same insight, when applying VASC to the same problem. Finally, VASC is also able to be performed iteratively, allowing for gradual refinement of the designs of interest before eventually selecting a final design choice.

7. **A demonstration of the feasibility of considering *multi-arc change paths*, and the capture of the associated change mechanism coupling benefits.** While the idea of change mechanisms working together to produce added value is not new, computational limitations previously hindered the calculation of this value. The creation of the full accessibility matrix, as shown in Figure 8-4, and its use in VASC demonstrates the ability for computers to now properly account for this value.

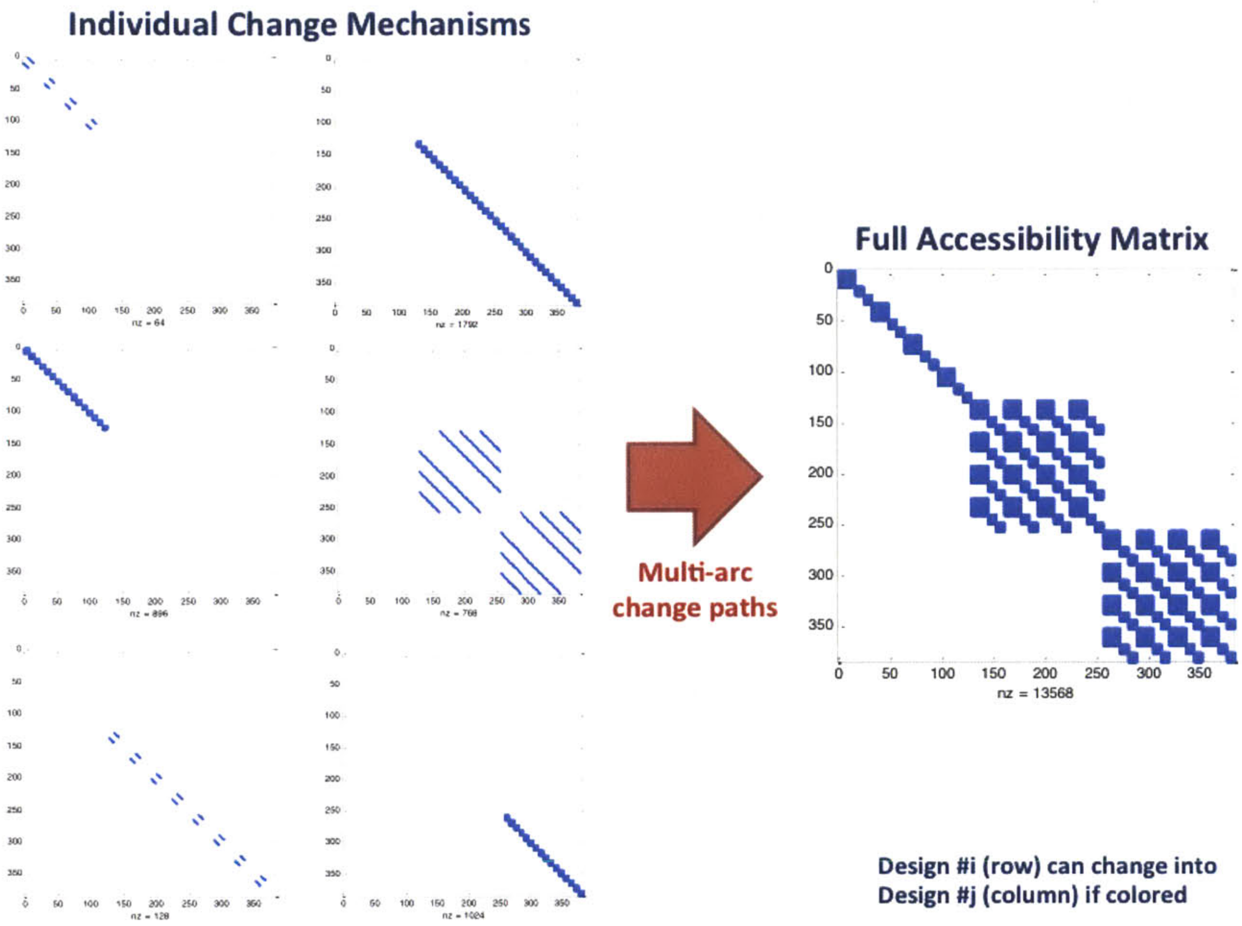


Figure 8-4: Creation of the Full Accessibility Matrix

8. **The application of VASC to two *case studies* in technical system design.** The demonstration of VASC on two different systems (X-TOS, Space Tug) shows the variety of insights able to be obtained through its use. A third case study (Satellite Radar) was unable to be completed, but was able to identify critical improvements that must be made in order for VASC to be applied to very large-scale tradespaces.

8.2 Final Thoughts

VASC is an approach, developed over two years of research, designed to improve upon the existing methods of changeability valuation by incorporating more types of value and fewer restrictive assumptions. The research described in this thesis appears to demonstrate the success of VASC in accomplishing these goals. However, significant work remains before changeable systems, valued early in the design process, can be viewed with the same level of confidence as passively robust systems. Hopefully future research can expand on VASC and aspects of changeability beyond valuation in order to further raise the profile of changeability as a means of creating systems with significant long-term value to stakeholders.

9 Acronym List

ARI – Available Rank Increase
DCF – Discounted Cash Flow
DM – Datar-Mathews method
DoDAF – Department of Defense Architecture Framework
DMM – Domain Mapping Matrix
DSM – Design Structure Matrix
EEA – Epoch-Era Analysis
eNPT / efNPT – Effective (fuzzy) Normalized Pareto Trace
ESM – Engineering Systems Matrix
FOD – Filtered Outdegree
FPN – Fuzzy Pareto Number
FPS – Fuzzy Pareto Shift
IRF – Integrated Real options Framework
MATE – Multi-Attribute Tradespace Exploration
NPT / fNPT – (fuzzy) Normalized Pareto Trace
NPV – Net Present Value
OD – Outdegree
ROA – Real Options Analysis
RSC – Responsive Systems Comparison method
SoS – System of Systems
TDN – Time-expanded Decision Network
VASC – Valuation Approach for Strategic Changeability
VDEA – Value Driven Enterprise Architecting
VFT – Value-focused Thinking
VWFO – Value-Weighted Filtered Outdegree

10 References

16.89, Space System Engineering, "X-TOS: Final Design Report." Cambridge, MA, Massachusetts Institute of Technology: 105, 2002.

K. Baldwin, "DoD Systems Engineering : Status Report," US Department of Defense, Report at the NDIA SE Division Meeting, June 15 2011.

J. Bartolomei, D. E. Hastings, R. de Neufville, and D. H. Rhodes, "Using a Coupled-Design Structure Matrix Framework to Screen for Real Options "In" an Engineering System," *INCOSE International Symposium 2006*, Orlando, FL, July 2006.

J. Bartolomei, "Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure," PhD Thesis, Engineering Systems Division, Massachusetts Institute of Technology, 2007.

J. C. Beesemyer, "Empirically Characterizing Evolvability and Changeability in Engineering Systems," SM Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2012.

F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol 81, p. 637–654, 1973.

M. Danilovic and T. R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices," *International Journal of Project Management*, vol. 25, no. 3, pp. 300-314, Apr. 2007.

I. Das and J. Dennis, "Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems", *SIAM Journal on Optimization,* Vol. 8, No.3, pp. 631-657, 1998.

R. de Neufville, "Uncertainty Management for Engineering Systems Planning and Design," *Proceedings of the 1st Engineering Systems Symposium ESS 04*. MIT Engineering Systems Division, p. 19, 2004.

R. de Neufville, S. Scholtes, and T. Wang, "Real Options by Spreadsheet: Parking Garage Case Example," *Journal of Infrastructure Systems*, vol. 12, no. 2, p. 107, 2006.

A. Deshmukh, M. Wortman, D. Jacques, T. Housel, K. Sullivan, and P. Collopy, "RT-18: Value of Flexibility," Phase 1 Progress Report, Systems Engineering Research Center, 2010.

M. E. Fitzgerald, A. M. Ross, and D. H. Rhodes, "A Method Using Epoch-Era Analysis to Identify Valuable Changeability in System Design," 9th Conference on Systems Engineering Research, Los Angeles, CA, April 2011.

M. E. Fitzgerald and A. M. Ross, "Mitigating Contextual Uncertainties with Valuable Changeability Analysis in the Multi-Epoch Domain," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012a.

M. E. Fitzgerald and A. M. Ross, "Sustaining Lifecycle Value: Valuable Changeability Analysis with Era Simulation," 6th Annual IEEE Systems Conference, Vancouver, Canada, March 2012b.

M. E. Fitzgerald, A. M. Ross, and D. H. Rhodes, "Assessing Uncertain Benefits: a Valuation Approach for Strategic Changeability (VASC)," INCOSE International Symposium 2012, Rome, Italy, July 2012. (forthcoming)

E. Fricke and A. P. Schulz, "Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle," *Systems Engineering*, vol. 8, no. 4, pp. 342-359, 2005.

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

C. D. Jilla and D. W. Miller, "Assessing the Performance of a Heuristic Simulated Annealing Algorithm for the Design of Distributed Satellite Systems," *Acta Astronautica*, vol. 48, 2001, pp. 529-543.

K. Kalligeros, O. D. Weck, R. D. Neufville, and A. Luckins, "Platform identification using Design Structure Matrices," *Sixteenth Annual International Symposium of the International Council On Systems Engineering INCOSE*, Orlando, FL, 2006.

I.Y. Kim and O. de Weck, "Adaptive weighted sum method for multi-objective optimization: a new method for Pareto front generation", *Structural and Multidisciplinary Optimization*, vol. 31 no. 2, 105-116, February 2006.

M. Maier, "Architecting Principles for Systems-of-Systems," *Systems Engineering*, vol. 1, no. 4, pp. 267-284, 1998.

S. Mathews and V. Datar, "A Practical Method for Valuing Real Options: The Boeing Approach," *Journal of Applied Corporate Finance*, vol. 19, no. 2, pp. 95-104, 2007.

H. McManus and T. Schuman, "Understanding the Orbital Transfer Vehicle Trade Space," AIAA Space 2003, Long Beach, CA, September 2003.

B. Mekdeci, A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Examining Survivability of Systems of Systems," in *INCOSE International Symposium 2011*, Denver, CO, June 2011.

T. Mikaelian, "An Integrated Real Options Framework for Model-based Identification and Valuation of Options under Uncertainty," PhD Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 2009.

T. Mikaelian, D. H. Rhodes, D. J. Nightingale, and D. E. Hastings, "Model-based Estimation of Flexibility and Optionability in an Integrated Real Options Framework," in *2009 3rd Annual IEEE Systems Conference*, Vancouver, Canada, 2009.

C. Mills, "Value-Driven Enterprise Architecture Evaluation for the Joint Force Protection Advanced Security System," MS Thesis, Air Force Institute of Technology, 2009.

J. Mun and T. Housel, "A Primer on Applying Monte Carlo Simulation, Real Options Analysis, Knowledge Value Added, Forecasting, and Portfolio Optimization Acquisitions," White paper for the Naval Postgraduate School, Monterey, CA, 2010.

S. C. Myers, "Finance Theory and Financial Strategy," *Interfaces*, vol. 14, no. 1, pp. 126-137, Jan. 1984.

A. Olewnik, T. Brauen, S. Ferguson, and K. Lewis, "A Framework for Flexible Systems and Its Implementation in Multiattribute Decision Making," *Journal of Mechanical Design*, vol. 126, no. 3, p. 412, 2004.

A. Olewnik and K. Lewis, "A decision support framework for flexible system design," *Journal of Engineering Design*, vol. 17, no. 1, pp. 75-97, 2006.

J. G. Pierce, "Designing Flexible Engineering Systems Utilizing Embedded Architecture Options," PhD Thesis, Interdisciplinary Studies: Systems Engineering, Vanderbilt University, 2010.

N. Ricci, A. M. Ross, and D. H. Rhodes, "Developing a Dynamic Portfolio-Based Approach for Systems of Systems Composition," WP-2012-2-1, 2012.

M. G. Richards, A. M. Ross, N. B. Shah, and D. E. Hastings, "Metrics for Evaluating Survivability in Dynamic Multi-Attribute Tradespace Exploration," *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 1049-1064, Sep. 2009.

A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, 2004.

A. M. Ross and D. E. Hastings, "The Tradespace Exploration Paradigm," in *INCOSE International Symposium 2005*, Rochester, NY, 2005.

A. M. Ross, "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration," PhD Thesis, Engineering Systems Division, Massachusetts Institute of Technology, 2006.

A. M. Ross and D. E. Hastings, "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," in *AIAA Space 2006*, San Jose, CA, 2006.

A. M. Ross and D. H. Rhodes, "Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis," *INCOSE International Symposium 2008*, Utrecht, the Netherlands, 2008.

A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining Changeability : Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value," *Systems Engineering*, vol. 11, no. 3, pp. 246-262, 2008.

A. M. Ross, H. L. McManus, D. H. Rhodes, and D. E. Hastings, "Responsive Systems Comparison Method : Dynamic Insights into Designing a Satellite Radar System," *AIAA Space 2009*, Pasadena, CA, 2009.

A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Using Pareto Trace to Determine System Passive Value Robustness," *3rd Annual IEEE Systems Conference*, pp. 285-290, Vancouver, Canada, March 2009.

A. M. Ross, H. L. McManus, D. H. Rhodes, and D. E. Hastings, "A Role for Interactive Tradespace Exploration in Multi-Stakeholder Negotiations," AIAA Space 2010, Anaheim, CA, September 2010.

J. Saleh, G. Mark, and N. Jordan, "Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems," *Journal of Engineering Design*, vol. 20, no. 3, pp. 307-323, Jun. 2009.

N. B. Shah, J. Wilds, L. Viscito, A. M. Ross, and D. E. Hastings, "Quantifying Flexibility for Architecting Changeable Systems," in *Conference on Systems Engineering Research 2008*, Los Angeles, CA, 2008.

M. R. Silver and O. L. de Weck, "Time-expanded decision networks: A framework for designing evolvable complex systems," *Systems Engineering*, vol. 10, no. 2, pp. 167-188, 2007.

R. M. Smaling, "System Architecture Analysis and Selection Under Uncertain," PhD Thesis, Engineering Systems Division, Massachusetts Institute of Technology, 2005.

D.V. Steward, "The Design Structure System: A Method for Managing the Design of Complex Systems." *IEEE Transactions on Engineering Management* 28: 71-74, 1981.

K. J. Sullivan, W. G. Griswold, Y. Cai, and B. Hallen, "The structure and value of modularity in software design," in *European Software Engineering Conference on the Foundations of Software Engineering*, vol. 26, no. 5, p. 99., 2001.

R. E. Swaney and I. E. Grossmann, "An Index for Operational Flexibility in Chemical Process Design," *Chemical Engineering*, vol. 31, no. 4, pp. 631-641, 1985.

L. Viscito, D. Chattopadhyay, and A. M. Ross, "Combining Pareto Trace with Filtered Outdegree as a Metric for Identifying Valuably Flexible Systems," in *Conference on Systems Engineering Research 2009*, Loughborough University, UK, 2009.

L. Viscito and A. M. Ross, "Quantifying Flexibility in Tradespace Exploration : Value Weighted Filtered Outdegree," in *AIAA Space 2009*, Pasadena, CA, 2009.

M. Walton, "Managing Uncertainty in Space Systems Conceptual Design Using Portfolio Theory," PhD Thesis, Engineering Systems Division, Massachusetts Institute Technology, 2002.

J. Wilds, J. Bartolomei, R. de Neufville, and D. E. Hastings, "Real Options 'In' a Micro Air Vehicle," in *Conference on Systems Engineering Research 2007*, Hoboken, NJ, 2007.

11 Appendices

The following sections contain a sampling of the MATLAB® functions that were used in the case studies of this thesis. They are meant to serve as examples for any potential adopters of VASC who want to implement their own project, in MATLAB® or another coding language, *not* as a prescriptive guide on how best to code the approach. They are, however, freely offered to all. All of the following functions are originally written by Matthew Fitzgerald unless otherwise noted.

Some commonly used abbreviations in the following functions:

- d = current or active design number
- d2 = potential transition end state for design d under consideration
- available = all potential end state designs
- td = total number of designs
- r = number of transition rules (change mechanisms)
- e = current epoch under consideration

11.1 Full Accessibility Matrix

Creating the full accessibility matrix is the work of two functions. First is ruleCollapse2, which creates all potential permutations of transition rule executions (up to the specified number of arcs), and then loops through them one at a time. The helper function designPathTool is used to find all of the potential combinations of change paths possible for a given initial design and rule permutation, and then evaluates whether or not each path is Pareto efficient compared to other paths between the initial and final design states. This is the most expensive operation in VASC, as designPathTool must work recursively and check <u>every</u> possible path, except for paths that travel to the same design more than once (as these are always dominated). Note that the input transition matrices are (1 by r) cell arrays of (td by td) matrices that display the dollar and time costs of transitions for individual mechanisms. An entry of zero in the transition matrices indicates no change path (so an extremely small number like 1e-6 is typically used to indicate a "free" change). The ruleCollapse2 function is also set to autosave every time a loop finishes and it has been 20 minutes since the last save; this can be disabled to save time, but is usually valuable because this function can run for days on larger data sets.

RuleCollapse2

```
% function to collapse given transition matrices into a single matrix

function [allRulePaths allDesignPaths allCostPaths] =
ruleCollapse2(Tcost,Ttime,minArc,maxArc,fuzzy,hotStart)
% INPUTS
% Tcost {r}(td,td) - transition dollar costs
%                       0 = no transition
% Ttime {r}(td,td) - transition times for each rule
% minArc (scalar) - minimum # arc transitions considered
% maxArc (scalar) - maximum # arc transitions considered
```

```
% fuzzy (scalar) - fuzzy pareto % allowed
% hotStart - if it is 0 --> start at beginning
%            otherwise, it should be a row vector [rulePath] to be the
%            first rulePath to start on
% OUTPUTS
% allPaths {td,td} - cell array of non-dominated paths from row design# to
%                    column design #.  Comes in three separate arrays, one
%                    for the transition rules used, one for the transition
%                    costs, and one for the string of intermediate designs

% initialize variables
td = size(Tcost{1},1);
r = length(Tcost);
allRulePaths = cell(td,td);
allDesignPaths = cell(td,td);
allCostPaths = cell(td,td);
for i = 1:td
    for j = 1:td
        % these two need to be cell arrays of cells, since the entries will
        % be of different size
        allRulePaths{i,j} = {};
        allDesignPaths{i,j} = {};
    end
end

% construct permutations
permMat = cell(1,maxArc);
% one arc
permMat{1} = (1:r)';
% loop through remainder
for arcs = 2:maxArc
    prevSize = size(permMat{arcs-1},1);
    for rule = 1:r
        temp = [rule*ones(prevSize,1) permMat{arcs-1}];
        permMat{arcs} = [permMat{arcs} ; temp];
    end
end

% start a timer for saving purposes
tic;
backTime = 0;

% check hotStart
if all(hotStart)

    % find starting option number if hotStart is not zero
    optStart = 1;
    for i = 1:length(hotStart)
        optStart = optStart + r^(length(hotStart)-i) * (hotStart(i) - 1);
    end

    % abbreviated loop for the hotStart-length arcs
    for option = optStart:size(permMat{length(hotStart)},1)
        % set rule path
        rulePath = permMat{length(hotStart)}(option,:);
```

```
        % loop through all potential start designs
        for startDesign = 1:td
            [rulePath startDesign]

            % update allPaths for this rule path and this start design
            [allRulePaths allDesignPaths allCostPaths] =
designPathTool(startDesign,rulePath,rulePath,Tcost,Ttime,allRulePaths,allDesi
gnPaths,allCostPaths,fuzzy);
        end

        % check timer and save progress if it has been 20 minutes since
        % last rule path completion
        time = toc;
        if time > 60*20
            backTime = 0;

save('RuleCollapseAutosave.mat','allRulePaths','allDesignPaths','allCostPaths
');
        else
            backTime = backTime + time;
        end
        tic;

    end
    % main loop for remainder of duration
    % loop through permMat
    for arcs = (length(hotStart)+1):maxArc
        % loop through options in there
        for option = 1:size(permMat{arcs},1)
            % set rule path
            rulePath = permMat{arcs}(option,:);

            % loop through all potential start designs
            for startDesign = 1:td
                [rulePath startDesign]

                % update allPaths for this rule path and this start design
                [allRulePaths allDesignPaths allCostPaths] =
designPathTool(startDesign,rulePath,rulePath,Tcost,Ttime,allRulePaths,allDesi
gnPaths,allCostPaths,fuzzy);
            end

            % check timer and save progress if it has been 20 minutes since
            % last rule path completion
            time = toc;
            if time > 60*20
                backTime = 0;

save('RuleCollapseAutosave.mat','allRulePaths','allDesignPaths','allCostPaths
');
            else
                backTime = backTime + time;
            end
            tic;
```

```
        end
    end




else % if hotStart was 0, just start at the beginning
    % loop through permMat
    for arcs = minArc:maxArc
        % loop through options in there
        for option = 1:size(permMat{arcs},1)
            % set rule path
            rulePath = permMat{arcs}(option,:);

            % loop through all potential start designs
            for startDesign = 1:td
                [rulePath startDesign]

                % update allPaths for this rule path and this start design
                [allRulePaths allDesignPaths allCostPaths] =
designPathTool(startDesign,rulePath,rulePath,Tcost,Ttime,allRulePaths,allDesi
gnPaths,allCostPaths,fuzzy);
            end

            % check timer and save progress if it has been 20 minutes since
            % last rule path completion
            time = toc;
            if time > 60*20
                backTime = 0;

save('RuleCollapseAutosave.mat','allRulePaths','allDesignPaths','allCostPaths
');
            else
                backTime = backTime + time;
            end
            tic;
        end
    end
end

end
```

DesignPathTool

```
% Recursive path tool

function [allRulePaths allDesignPaths allCostPaths] =
designPathTool(designPathIn,remainingRulePath,fullRulePath,Tcost,Ttime,allRul
ePaths,allDesignPaths,allCostPaths,fuzzy)

% IF THERE ARE REMAINING RULES TO EXECUTE
if ~isempty(remainingRulePath)
    % identify current rule, remove from rule path
```

```
    currRule = remainingRulePath(1);
    rulePathOut = remainingRulePath;
    rulePathOut(1) = [];

    % identify current design and possible end states from current design
    currDesign = designPathIn(end);
    endStates = find(Tcost{currRule}(currDesign,:) > 0);

    % loop through possible end states as new design paths but ONLY IF D2
    % IS NOT ALREADY IN THE DESIGN PATH (this is always non-efficient for
most applications)
    for d2 = endStates
        if isempty(intersect(designPathIn,d2))
            [allRulePaths allDesignPaths allCostPaths] =
designPathTool([designPathIn
d2],rulePathOut,fullRulePath,Tcost,Ttime,allRulePaths,allDesignPaths,allCostP
aths,fuzzy);
        end
    end

else % IF THE FULL RULE PATH IS EXECUTED
    % identify start and finish states
    start = designPathIn(1);
    finish = designPathIn(end);

    % calculate total costs
    dollarcost = Tcost{fullRulePath(1)}(start,designPathIn(2));
    timecost = Ttime{fullRulePath(1)}(start,designPathIn(2));
    % go through steps incrementing cost for this design path
    for step = 2:length(fullRulePath)
        currRule = fullRulePath(step);
        d1 = designPathIn(step);
        d2 = designPathIn(step+1);
        dollarcost = dollarcost + Tcost{currRule}(d1,d2);
        timecost = timecost + Ttime{currRule}(d1,d2);
    end

    % append data to the appropriate location of allPaths cell arrays
    allCostPaths{start,finish} = [allCostPaths{start,finish} ; [dollarcost
timecost]];
    allDesignPaths{start,finish} = [allDesignPaths{start,finish} ;
designPathIn];
    allRulePaths{start,finish} = [allRulePaths{start,finish} ; fullRulePath];


    % recalculate pareto set of the costs if it is not the only one there
    % then trim out all nonpareto paths
    if size(allCostPaths{start,finish},1) > 1
        setIndex = gen_pareto_set(allCostPaths{start,finish},[0
0],1,false,fuzzy);
        allCostPaths{start,finish} = allCostPaths{start,finish}(setIndex,:);
        allDesignPaths{start,finish} =
allDesignPaths{start,finish}(setIndex,:);
        allRulePaths{start,finish} = allRulePaths{start,finish}(setIndex,:);
    end
```

```
    % this update of allPaths is what is recorded as output

end

end
```

11.2 Pareto Set

Finding the Pareto set of a group of multi-dimensional data is of critical importance to VASC. The following function was originally created by Nirav Shah, and slightly modified with his help. Note that it carries the capability to determine fuzzy Pareto sets as well.

```
function pareto_set = gen_pareto_set(Z_all, varargin)
% GEN_PARETO_SET Returns row indices of the (fuzzy) pareto optimal subset
%
% pareto_set = gen_pareto_set(Z_all)
%           or
% pareto_set = gen_pareto_set(Z_all, BiB)
%           or
% pareto_set = gen_pareto_set(Z_all, BiB, sortObj,returnUnique)
%           or
% pareto_set = gen_pareto_set(Z_all, BiB, sortObj,returnUnique,FuzzyK)
%
%   Arguments passed as and empty matrix, [], will retain default values
%
%   Z_all:      Objective values (one row per observation)
%
%   BiB:        Should objective i be bigger is better (BiB(i) = 1) or
%   smaller is better (BiB(i) = 0).  Default si smaller is better for all
%   objectives.
%
%   sortObj:    The returned pareto set will ordered by this objective.
%   Default is not sort the pareto set.
%
%   returnUnique:  Set to true if you want a unique subset of pareto set
%   members.  Default is to return all pareto set members.
%
%   FuzzyK:     Front fuzziness factor.  Amount by which an observation can
%   be away from the front and still considered optimal.  Expressed as a
%   fraction of the objective function(s) range.  Default is to allow no
%   fuzziness.
%
% Version: 2009.10.01
% Nirav B. Shah 01/2009

% Debug plotting flag
plotDebug = false;

% Process and check the input arguments
% check the number of arguments
error(nargchk(1,5,nargin,'struct'));

numObj = size(Z_all,2);
numDes = size(Z_all,1);
```

```
% Defaults
sortObj = 0; % Pareto set is unsorted
fuzzMargin = zeros(1,numObj); % No fuzz margin
returnUnique = false; % return all set members

if nargin >= 2
    if ~isempty(varargin{1})
        BiB = logical(varargin{1});
        if length(BiB) ~= size(Z_all,2)
            error('GEN_PARETO_SET:BadBiBsize',...
                'The number of element in argument BiB must equal
size(Z_all,2)');
        end;
        % make all objective smaller is better
        Z_all(:,BiB) = -Z_all(:,BiB);
    end;
end;

minZ = min(Z_all);
maxZ = max(Z_all);

if nargin >= 3
    if ~isempty(varargin{2})
        sortObj = varargin{2};
        if (sortObj < 1) || (sortObj > numObj)
            error('GEN_PARETO_SET:SortObjOutOfRange',...
                'sortObj must be between 1 and the number of objectives');
        end;
    end;
end;
if nargin >= 4
    if ~isempty(varargin{3})
        returnUnique = logical(varargin{3});
    end;
end;
if nargin >= 5
    if ~isempty(varargin{4})
        fuzzK = varargin{4};
        fuzzMargin = fuzzK * (maxZ - minZ);
        if (fuzzK < 0) || (fuzzK > 1)
            error('GEN_PARETO_SET:fuzzKOutOfRange',...
                'fuzzK must be between 0 and 1');
        end;
    end;
end;


%%%%%%%%%%%%% Find the pareto front %%%%%%%%%%%%
% compute each points manhattan distance from utopia
[utopDist,utopIdx] = sort(sum(abs((Z_all-
repmat(minZ,numDes,1))./repmat(maxZ,numDes,1)),2));

% march through points in order of distance
paretoFlag = -ones(1,numDes);
```

```
for candidate = utopIdx'
    if paretoFlag(candidate) == -1
        canObj = Z_all(candidate,:);

        % build up domination vector for the candidate
        if any(canObj - fuzzMargin < minZ)
            dom = false;
        else
            dom = true(numDes,1);
            for theObj = 1:numObj
                dom = dom & Z_all(:,theObj) < canObj(theObj) -
fuzzMargin(theObj);
            end;
        end;

        % if there were no dominating designs ...
        if (isempty(find(dom,1)))
            % find all the designs that the candidate dominates and
            % remove them from consideration

            if all(canObj + fuzzMargin < maxZ)
                dom = true(numDes,1);
                for theObj = 1:numObj
                    % note the switch in the direction of inequality vs above
                    dom = dom & Z_all(:,theObj) >= canObj(theObj) +
fuzzMargin(theObj);
                end;
%                 dom = all(Z_all >= repmat((canObj +
fuzzMargin),numDes,1),2);
                % remove the dominated designs from consideration
                paretoFlag(dom) = 0;
            end;
            % mark the candidate as in the pareto set
            paretoFlag(candidate) = 1;
        else
            % remove the candidate from consideration
            paretoFlag(candidate) = 0;
        end;
    end;
    % Plotting routine for debugging
    if plotDebug && numObj > 1 && (candidate == utopIdx(end) ||
mod(candidate,numDes/100)<1)
        plot(Z_all(:,1),Z_all(:,2),'g.',...
            Z_all(paretoFlag == 1,1),Z_all(paretoFlag == 1,2),'r.',...
            Z_all(paretoFlag == 0,1),Z_all(paretoFlag == 0,2),'b.');
        title(['GEN_PARETO_SET Debug plot -- ' ...
            'Percent Complete: ' num2str(floor((1-
sum(paretoFlag<0)/numDes)*100))],...
            'Interpreter','none');
        xlabel('Z_1');
        ylabel('Z_2');
        drawnow;
    end;
end;

% idenitfy pareto flagged indices
```

```
pareto_set = find(paretoFlag == 1);

% Post process set depending upon flags passed
if returnUnique
    % ensure uniqueness
    [B,I] = unique(Z_all(pareto_set,:),'rows');
    if length(I) ~= length(pareto_set)
        warning('GEN_PARETO_SET:RepeatedParetoDesigns',...
            'There are repeated points on the Pareto Front, a unique sub-set
will be returned');
    end;
    pareto_set = pareto_set(I);
end;
if sortObj
    [sortObjVals,sortParetoIdx] = sort(Z_all(pareto_set,sortObj));
    pareto_set = pareto_set(sortParetoIdx);
end;
```

11.3 Fuzzy Pareto Number

This function assigns FPN to each design in each epoch by repeatedly calling gen_pareto_set for decreasing levels of fuzziness. An alternative method would be to find the 0% fuzzy Pareto front in each epoch and then determine the buffer necessary to include each point directly, but it is unclear if that method would be faster.

```
function fuzzyParetoNumbers = calcFuzzyParetoNumbers(Utility,Cost)
% This function calculates the minimum XX% Fuzzy Pareto Set for XX = 0:100
% necessary for a design to be included, for each design in each epoch.

% Inputs:
% Utility should be the utility matrix (i,j) for i=design, j=epoch, using
% either single or multi attribute utility
% Cost should be the cost for each design in each epoch, oganized the same
% way.  OPERATING costs should be used if intending to use this function
% with changeability analysis tools.

% Requirements:
% calls gen_pareto_set.m

numdesigns = size(Utility,1);
numepochs = size(Utility,2);

fuzzyParetoNumbers = -ones(numdesigns,numepochs);

% set all NaN's to 0 utility, max cost for the purposes of calculating fuzzy
% percentages
idx = isnan(Utility);
Utility(idx) = 0;
Cost(idx) = max(max(Cost));

for epoch = 1:numepochs
    disp(['Calculating Fuzzy Pareto Levels for epoch: ' num2str(epoch)])
```

```
    for percent = 100:-1:0
        currSet = gen_pareto_set([Utility(:,epoch),Cost(:,epoch)],[1
0],1,false,percent/100);
        fuzzyParetoNumbers(currSet,epoch) = percent;
    end
end

% retroactively set all NaNs to fuzzy pareto 101
fuzzyParetoNumbers(idx) = 101;


end
```

11.4 Available Rank Improvement

This simple function calculates ARI (normalized by the number of designs in the tradespace) for every design/rule pair in a given epoch (determined by the MAU input).

```
% ARI calculator

function ARI = calcARI(Tcost,MAUorder)

numD = size(Tcost{1},1);
numR = length(Tcost);

for d = 1:numD
    rank(d) = find(MAUorder==d)./numD;
end

for d = 1:numD
    d
    currRank = rank(d);
    for r = 1:numR
        available = find(Tcost{r}(d,:)>0);
        if ~isempty(available)
            availableRank = rank(available);
            best = min(availableRank);
            ARI(d,r) = max(currRank-best, 0);
        else
            ARI(d,r) = 0;
        end
    end
end
```

11.5 Strategy

The following is an example strategy function, which outputs the end state design number, transition rules used, and total cost of the change for the selected path of each design in each epoch. In particular, this function uses the Maximize Utility strategy, with the ability to apply a maximum transition cost threshold as well. Note that this function operates on the full accessibility matrix: reading in rulePaths and costPaths as inputs, which are the outputs generated by ruleCollapse2.

```
% maximize utility (w/ threshold) strategy file

function [endStates rulesExecuted transCost] =
STRATcollapsed_maxU(MAU,rulePaths,costPaths,cost_thresh,time_thresh)

numDesigns = size(MAU,1);
numEpochs = size(MAU,2);

endStates = zeros(numDesigns,numEpochs);
rulesExecuted = cell(numDesigns,numEpochs);
transCost = cell(numDesigns,numEpochs);

% set NaN MAUs to -1 (failure)
MAU(isnan(MAU)) = -1;

for d = 1:numDesigns
    disp(['Calculating Maximize Utility strategy results for design: '
num2str(d)])
    for e = 1:numEpochs

        % find available designs (nonempty cols of costPaths)
        available = [];
        for d2 = 1:numDesigns
            if ~isempty(costPaths{d,d2})
                available = [available d2];
            end
        end

        % find available design with highest MAU
        availableMAU = MAU(available,e);
        [sortedMAU indices] = sort(availableMAU); % sorts largest = last
        sortedAvailable = available(indices);

        % designate as unsolved and loop until solved
        unsolved = 1;
        while unsolved
            if isempty(sortedAvailable)
                break
            end
            currTarget = sortedAvailable(end);
            currTargetMAU = sortedMAU(end);
            % break out if the MAU being considered is worse than existing
MAU
            if currTargetMAU <= MAU(d,e) && MAU(d,e) == -1
                break
            end
            if currTargetMAU < MAU(d,e) && MAU(d,e) ~= -1
                break
            end
            % consider the different paths available to the best end state,
sort and select the cheapest one
            pathOptions = costPaths{d,currTarget};
            [~, idx] = sort(pathOptions(:,1));
```

```
            pathOptions = pathOptions(idx,:);
            for i = 1:size(pathOptions,1)
                if pathOptions(i,1) < cost_thresh && pathOptions(i,2) <
time_thresh
                    endStates(d,e) = currTarget;
                    rulesExecuted{d,e} = rulePaths{d,currTarget}{idx(i),:};
                    transCost{d,e} = pathOptions(i,:);
                    unsolved = 0;
                    break
                end
            end
            % if none of the paths for the best utility design were
acceptable to the thresholds, delete that design from consideration and loop
through again on the next best design
            sortedAvailable(end) = [];

        end
        % reports no change for this (d,e) if previous loop does not find a
solution
        % this is fine if the design is feasible, but we should note
        % failure (endState = NaN) if it is not
        if endStates(d,e) == 0 && MAU(d,e) == -1
            endStates(d,e) = NaN;
        end
    end
end
end
```

11.6 Fuzzy Pareto Shift

FPS (like may multi-epoch metrics) is inexpensive to compute after the determination of the strategic paths. This function reads in only the targeted end state designs (for each initial design in each epoch) and the FPNs (for each design in each epoch) and performs the appropriate differencing in order to find the FPS for each design in each epoch.

```
% fuzzy pareto shift calculator
function [FPS] = calcFPS(endStates,fuzzyParetoNumbers)

numDesigns = size(endStates,1);
numEpochs = size(endStates,2);

FPS = zeros(numDesigns,numEpochs);

for d = 1:numDesigns
    disp(['Calculating Fuzzy Pareto Shift for design: ' num2str(d)])
    for e = 1:numEpochs
        d2 = endStates(d,e);

        if isnan(d2)  % if this was a failure, set to failure value (-101)
            FPS(d,e) = -101;
        elseif d2 == 0 % no transition
            FPS(d,e) = 0;
        else
            FPS(d,e) = fuzzyParetoNumbers(d,e) - fuzzyParetoNumbers(d2,e);
        end
```

```
    end
end
```

11.7 Era Simulation

The following is an example era simulation function (in this case, it is the "random assortment epochs" used in the X-TOS case study), that simulates a single era for a single initial design. The inputs are the initial design and era length, the selected end states and transition costs from the strategy function, and the FPNs for tracking purposes. It outputs statistics such as the total transition usage and average FPN: tailoring the era function to report any data deemed to be of interest is an important step, as there are many dimensions of data present in a system changing over time. A "success" metric is also present here despite not being mentioned in the X-TOS case study, because this function was designed to work with data sets that did not pre-eliminate invalid designs; if invalid designs were considered, some designs could "fail" by becoming invalid and not being able to change to a valid design, truncating the era. The likelihood of successful completion of a full lifetime is potentially of great interest for systems at risk of sudden failure. In VASC, this function is repeated thousands of times in order to reach statistical significance over the randomness inherent in the epoch shifts.

```
% Era Simulator - Random epoch of duration 1
function [success, totalTCost, totalTTime, totalTrans, avgFPN] =
eraSim_random(initialDesign, transDesigns, transCosts, FPN, eraLength)
% INPUTS
    % initialDesign = design # of starting point
    % transDesigns, transCosts = (design,epoch) outputs of STRATEGY_____.m
files
    % eraLength = integer length (in arbitrary units) of the era
    % FPN = (design,epoch) matrix of FPN values
% OUTPUTS
    % success = 1 if "eraLength" is met, 0 otherwise
    % totalTCost = total transitioning $ cost
    % totalTTime = total transitioning time cost
    % totalTrans = total # transitions
    % avgFPN = average FPN for the design as it changes over the era

numEpochs = size(transDesigns,2);
currDesign = initialDesign;
totalTCost = 0;
totalTTime = 0;
totalTrans = 0;
time = 0;
success = 0;
FPNtrack = [];

while time<eraLength
    currEpoch = ceil(numEpochs*rand(1));
    FPNtrack = [FPNtrack FPN(currDesign,currEpoch)];
    time = time + 1;

    destination = transDesigns(currDesign,currEpoch);
    if destination == -1
        break
    elseif destination ~= 0
```

```
        totalTCost = totalTCost + transCosts{currDesign,currEpoch}(1);
        totalTTime = totalTTime + transCosts{currDesign,currEpoch}(2);
        totalTrans = totalTrans + 1;
        currDesign = destination;
    end
end

if time >= eraLength
    success = 1;
end

avgFPN = mean(FPNtrack);

end
```